Proceedings of the Workshop on the Thirteenth International Competition on Legal Information Extraction and Entailment (COLIEE 2026)

in association with
the 21st International Conference on
Artificial Intelligence and Law

COLIEE 2026 Organizers

Randy Goebel, University of Alberta, Canada
Yoshinobu Kano, Shizuoka Univesity, Japan
Calum Kwan, University of Alberta, Canada
Mi-Young Kim, University of Alberta, Canada
Ken Satoh, Center for Juris-informatics, Japan
Hiroaki Yamada, Institue of Science Tokyo, Japan
Masaharu Yoshioka, Hokkaido University, Japan

July 6, 2026

Program Committee

Wachara Fungwacharakorn	Center for Juris-Informatics, ROIS-DS, Japan
Randy Goebel	University of Alberta
H M Quamran Hasan	University of Alberta
Yoshinobu Kano	Shizuoka University
Mi-Young Kim	Department of Computing Science, U. of Alberta, Canada
Yuntao Kong	NII
Calum Kwan	University of Alberta
Tung Le	Faculty of Information Technology, University of Science, VNU-HCM, Vietnam
Ha-Thanh Nguyen	National Institute of Informatics
Md Abed Rahman	University of Alberta
Ken Satoh	Center for Juris-Informatics, ROIS, Japan
Hsuan-Lei Shao	Taipei Medical University
Nguyen Tien Huy	University of science HCM
Thi-Hai-Yen Vuong	yenvth@vnu.edu.vn
Hiroaki Yamada	Institute of Science Tokyo
Masaharu Yoshioka	Hokkaido University

Additional Reviewers

Fungwacharakorn, Wachara
Kim, Yeji
Nitta, Katsumi
Zin, May Myo

Table of Contents

Cross-Architecture LLM Ensembles, Feature-Based Reranking and Retrieval-Augmented Prompting for Legal Information Processing

Amal Saad Alshehri
Durham University
Department of Computer Science
Durham, United Kingdom
Jazan University
Department of Computer Science
Jazan, Saudi Arabia
ashahri@jazanu.edu.sa

Nelly Bencomo
Durham University
Department of Computer Science
Durham, United Kingdom
nelly.bencomo@durham.ac.uk

Amir Atapour-Abarghouei
Durham University
Department of Computer Science
Durham, United Kingdom
amir.atapour-abarghouei@durham.ac.uk

ABSTRACT

Legal information processing encompasses a heterogeneous set of retrieval, entailment and judgment prediction problems, requiring combinations of text matching, reasoning and robust generalisation with limited supervision. This paper presents a unified study of these challenges through a suite of open-weight systems spanning legal case retrieval, case entailment, statute retrieval and entailment and legal judgment prediction. In this paper, we report Team DU's participation in all five tasks of COLIEE 2026, with all systems relying exclusively on open-weight models. For Tasks 3 and 4, all models were released before 15 July 2025, as required by the competition rules. For Task 4 (statute entailment), a cross-architecture ensemble of nine models from three families achieves 96.3% accuracy, placing first among 33 submissions from 11 teams. For the Pilot Task (tort prediction and rationale extraction), a multi-view system that combines five claim-level models and refines the case verdict using features derived from the claim predictions achieves 73.1% TP accuracy and 68.2% RE F1 as an unofficial submission, scoring above all official entries on TP and matching the highest on RE. For Task 2 (legal case entailment), changing only the prompt instruction from single-selection to multi-selection raises F1 from 0.343 to 0.555 in post-competition evaluation on released gold labels, exceeding the best official submission (F1 = 0.490). For Task 3 (statute retrieval and entailment), replacing the entailment model with Qwen3-235B and a structured legal reasoning prompt raises accuracy from 79.3% to 91.5% in post-competition analysis. For Task 1 (legal case retrieval), a learning-to-rank system that combines lexical and semantic retrieval with structural, citation authority, and temporal features (34 in total) achieves F1 = 0.314 (rank 11 of 54 submissions from 22 teams). Taken together, these results show that legal information processing benefits from different forms of inductive bias across tasks, with cross-architecture ensembling, feature-based reranking and retrieval-augmented prompting each proving most effective in different settings.

COLIEE 2026, June 12, 2026, Singapore

KEYWORDS

COLIEE, legal information retrieval, legal entailment, LLM ensemble, learning-to-rank, legal judgment prediction

1 INTRODUCTION

Legal information processing involves a diverse family of problems, including document retrieval, textual entailment and judgment prediction, each of which places different demands on representation learning, structured reasoning and decision-making under domain-specific constraints [1]. These tasks become particularly challenging in legal settings, where documents are long, terminology is highly specialised, relevant evidence may be distributed unevenly across a text and small linguistic distinctions can materially alter legal meaning.

COLIEE[1] provides a well-established benchmark for studying these challenges in a controlled setting and has run an annual competition on legal information extraction and entailment running since 2014 [6, 8]. The 2026 edition [8] comprises four established tasks and one pilot task. Tasks 1 and 2 address Canadian case law: Task 1 requires retrieving cases originally cited by a query case, and Task 2 requires identifying which paragraphs from a candidate case entail a given decision fragment. Tasks 3 and 4 address Japanese statute law: Task 3 requires both retrieving relevant Civil Code articles and predicting entailment, while Task 4 tests entailment alone with the relevant articles provided by the organisers. The Pilot Task addresses legal judgment prediction for Japanese tort cases, requiring both a binary verdict prediction (tort prediction, TP) and per-claim acceptance labels (rationale extraction, RE). We participate in all five tasks as Team DU. Table 1 summarises our submissions. Our strongest results are on Task 4, where a cross-architecture ensemble achieves first place. On the Pilot Task, our unofficial submission achieves 73.1% TP accuracy and 68.2% RE F1, scoring above all official entries on TP and matching the highest on RE. Post-competition analysis with the released gold labels identifies improvements for Tasks 2 and 3 that require no changes to the system architecture.

The remainder of this paper is organised by task to reflect the heterogeneity of the benchmark. Sections 2 through 6 describe

[1] https://coliee.org/

Table 1: Summary of Team DU submissions across five tasks.

Task	Approach	Best Score	Rank
1	LightGBM LTR (34 features)	F1 = .314	11/54
2	MonoT5 + few-shot LLM	F1 = .343	22/35
3	BM25 bigrams + Qwen-72B	Acc = .793	14/22
4	Cross-arch meta-ensemble (DU1)	Acc = .963	**1st**
Pilot	5-view BERT + verdict bridge	TP = .731	—*

*Unofficial entry; TP = .731 exceeds all official entries and RE F1 = .682 matches the highest official result (see Section 6).

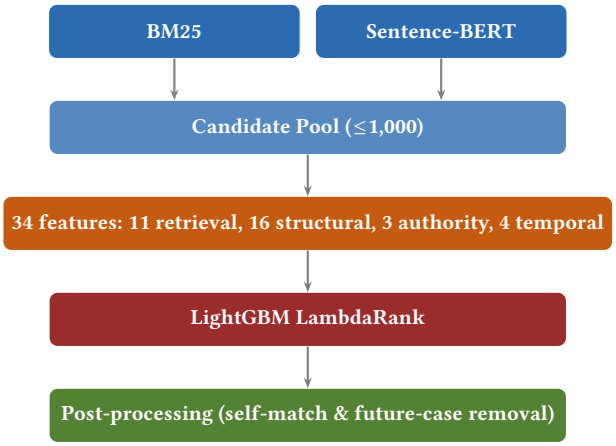


Figure 1: Task 1: Stage 1 retrieves candidates via BM25 and dense retrieval. Stage 2 extracts 34 features across four families and feeds them to a LightGBM reranker. Stage 3 enforces domain constraints.

the method, results, and analysis for each task. Section 7 discusses cross-task patterns and Section 8 concludes.

2 TASK 1: LEGAL CASE RETRIEVAL

Legal case retrieval is challenging because cited authorities are often related by legal function rather than surface lexical overlap alone, while the underlying documents are long and structurally heterogeneous. Given a query case from the Canadian Federal Court, where all references to cited cases have been removed from the text, the objective of our system is to identify which cases in the corpus were originally cited by the query. The organisers provide a training set of 2,001 labelled query cases over a corpus of 7,708 candidates. The test set contains 400 queries over a separate corpus of 1,848 candidates. Performance is measured by micro-averaged F1 at k=5, where each query receives exactly five predicted citations.

2.1 Method

The system has three stages (Figure 1).

Stage 1: Hybrid candidate retrieval. BM25 [12] and a Sentence-BERT dense retriever [11] each return the top 1,000 candidates per query. The union of both ranked lists forms a candidate pool containing 99.1% of the true cited cases on the development set. BM25 excels at matching specific legal terminology and case names, while dense retrieval captures semantic similarity when surface terms differ.

Stage 2: Feature-based reranking. For each query-candidate pair, 34 features from four families are extracted. The *retrieval family* (11 features) includes normalised BM25 scores, dense cosine similarity, RRF [3] scores combining the BM25 and dense rankings, Jaccard similarity, overlap coefficient, and IDF-weighted overlap. The *structural family* (16 features) segments documents into Background, Issues, Analysis, and Decision sections using rule-based heading detection and positional heuristics, and computes paragraph-level Jaccard similarities (maximum, mean of the top three, and a coverage score) within and across sections. The *authority family* (3 features) captures normalised citation indegree from the training corpus, the difference in indegree between query and candidate, and log-transformed indegree, exploiting the observation that frequently cited cases are more likely to be cited again. The *temporal family* (4 features) encodes the signed and absolute year difference between filing dates, a binary indicator for the 1 to 15 year citation window, and a binary indicator for candidates decided after the query, reflecting that 82% of real citations fall within this window. These 34 features are passed to a LightGBM [9] model trained with a LambdaRank objective on the 2,001 queries from the COLIEE 2026 training set.

Stage 3: Post-processing. Self-matches and candidates dated after the query are filtered out, contributing +2.7 pp to F1 on the development set. Three runs are submitted: DU1 (1,000 trees, 63 leaves, learning rate 0.05), DU2 (score-level fusion of DU1 and DU3), and DU3 (2,000 trees, 127 leaves, learning rate 0.03).

2.2 Results and Analysis

Table 2 reports the official results. DU3 achieves F1 = 0.3141, ranking 11th of 54 submissions from 22 teams. The competition winner, NOWJ, achieves F1 = 0.4220, a gap of 10.8 pp. DU2 (score-level fusion of the two tree configurations) achieves F1 = 0.3136, nearly identical to DU3, indicating that the two tree configurations make similar errors and fusion provides no additional diversity. Post-competition scaling to 4,000 trees (DU9) raises F1 to 0.3456, reducing the gap to the winner by 3.15 pp.

The median document length of 4,573 tokens exceeds the 512-token limit of most encoder models. We tested both a fine-tuned cross-encoder and a zero-shot large language model as alternatives to the feature-based reranker, but neither improved over LightGBM, because truncation discards the very sections (Analysis and Decision) that our structural features identify as most informative.

Feature ablation. Table 3 reports a cumulative feature ablation on the development set. Starting from BM25 scores alone (F1 = 16.50%), adding retrieval features raises F1 to 27.56% (+11.06 pp), structural features to 29.37% (+1.81 pp), authority features to 30.49% (+1.12 pp), and temporal features to 31.18% (+0.69 pp). Expanding to all available training data raises F1 to 35.70% (+4.52 pp). In the final model, the top five features by total gain are RRF at k=10 (14.1%), RRF at k=5 (6.5%), mean top-3 paragraph Jaccard (3.8%), citation indegree (2.5%), and maximum paragraph-level IDF overlap (2.0%). The dominance of the two RRF features confirms that combining lexical and semantic retrieval signals is the most informative input.

Feature interaction effects. The two binary temporal indicators (whether the candidate falls within the 1–15 year citation window and whether the candidate postdates the query) have near-zero

Table 2: Task 1 official results (selected) and post-competition.

Rank	Run	Prec.	Rec.	F1
1	NOWJ	.424	.421	.422
3	JNLP	.434	.393	.413
11	**DU3**	**.295**	**.337**	**.314**
	DU2	.294	.336	.314
	DU1	.285	.325	.304
Post-competition				
	DU9	.324	.370	**.346**

Table 3: Task 1 feature ablation on the development set. Features are added cumulatively.

Feature Set	F1 (%)	Δ (pp)
BM25 only	16.50	—
+ Retrieval features (11)	27.56	+11.06
+ Structural features (16)	29.37	+1.81
+ Authority features (3)	30.49	+1.12
+ Temporal features (4)	31.18	+0.69
+ All available training data	35.70	+4.52

individual importance. However, removing both together degrades F1 by 2.08 pp on the development set, a larger effect than the authority or temporal groups individually. Binary features serve as split guides in gradient-boosted trees, allowing subsequent splits to learn separate thresholds for each partition.

3 TASK 2: LEGAL CASE ENTAILMENT

Case entailment is particularly sensitive to granularity mismatch, since the system must align a short decision fragment with one or more entailing paragraphs embedded in a much larger candidate case. Given a decision fragment and candidate paragraphs from another case, the system must identify which paragraph(s) entail the decision. The training set contains cases with gold paragraph labels. The test set contains 100 cases with 294 gold paragraphs (average 2.94 per case), evaluated by F1.

3.1 Method

The system has three stages (Figure 2).

Stage 1: Lexical retrieval. BM25 [12] retrieves the top 100 candidate paragraphs per decision fragment, achieving 99.0% recall at rank 100 on the test set.

Stage 2: Neural reranking. MonoT5-COLIEE [10], a T5-based cross-encoder fine-tuned on the COLIEE Task 2 training data with hard negative mining, serves as the primary reranker. Qwen3-Reranker-0.6B provides a complementary score, and the two are combined with weights 0.8 and 0.2 respectively. The top 20 paragraphs are retained for the next stage.

Stage 3: LLM entailment selection with retrieval-augmented few-shot prompting. A large language model receives the decision fragment, the reranked top-20 paragraphs, and three solved training examples retrieved by BM25 from an index built over all training queries. By grounding the LLM in concrete precedent rather than relying on zero-shot reasoning, retrieval-augmented few-shot

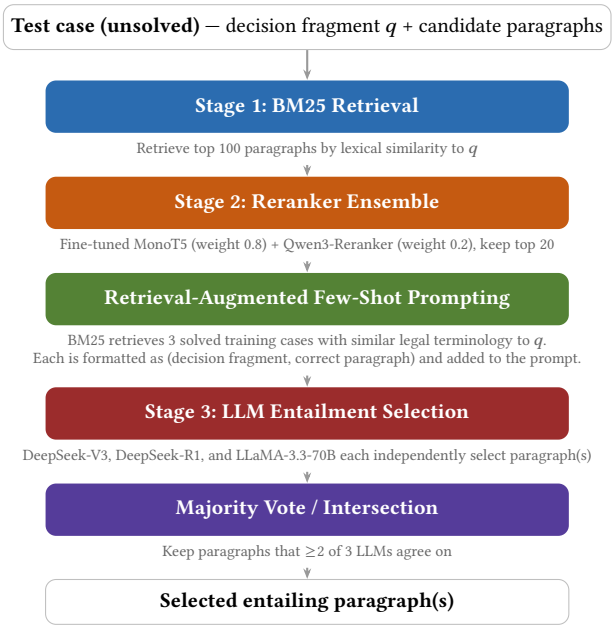


Figure 2: Task 2: three-stage pipeline with retrieval-augmented few-shot prompting. Three solved training cases, retrieved by BM25 similarity, are provided to each LLM as demonstrations. No test labels are used.

prompting adds 5.7 F1 points over zero-shot prompting on the development set. Three models (DeepSeek-V3, DeepSeek-R1 [4], LLaMA-3.3-70B) independently select paragraphs. Three runs use different aggregation strategies: DU1 (intersection of DeepSeek-V3 and R1), DU2 (DeepSeek-V3 alone, selected for its best precision-recall tradeoff on the development set), DU3 (three-model tiebreaker without few-shot demonstrations, testing whether the models alone can compensate for the absence of retrieved examples).

In-distribution hard negative training. A methodological contribution in Stage 2 is the MonoT5-v2 variant, which addresses a distribution mismatch in the standard hard negative mining approach. MonoT5-v1 trains by pairing each correct paragraph with five incorrect paragraphs from the same case as negative examples. At inference time, however, the model scores paragraphs that have already been reranked by the ensemble, which have a different score distribution from BM25 negatives. MonoT5-v2 resolves this by using a two-pass approach: a first pass produces the reranker's own top-20 predictions on the training set, and the non-gold paragraphs from these predictions serve as hard negatives for a second pass of fine-tuning. This ensures that the model trains on the exact distribution of candidates it will encounter at inference time. MonoT5-v2 improves the reranker's ability to retain gold paragraphs in the top 20, reducing the number of gold paragraphs lost before they reach the LLM stage.

3.2 Results and Analysis

Table 4 reports the official results. DU2 achieves the second-highest precision among all 35 submissions (0.753) but low recall (0.218), resulting in F1 = 0.338. The competition winner, IAI, achieves F1 =

Table 4: Task 2 results: official submissions and post-competition multi-selection (same pipeline, different prompt instruction).

Configuration	Prec.	Rec.	F1
Official submissions			
IAI (winner)	.450	.537	.490
DU3	.691	.228	.343
DU2	**.753**	.218	.338
DU1	.663	.214	.324
Post-competition (multi-selection prompt)			
DU majority vote	.619	.503	**.555**

0.490. The single-selection prompt was calibrated on the development set, where 80% of cases have exactly one correct paragraph (average 1.22). The test set averages 2.94 correct paragraphs per case, with 95% having more than one, capping the theoretical single-selection F1 ceiling at 0.508.

Development versus test distribution shift. DU2 achieves F1 = 0.764 on the development set but only F1 = 0.338 on the test set, a 42.6 pp gap entirely attributable to the distribution shift in the number of gold paragraphs per case. This experience motivates a general recommendation: when the number of correct answers per query is uncertain, the safest strategy is to allow multi-selection with a confidence threshold rather than enforcing a fixed count.

Post-competition multi-selection. Table 4 reports the effect of changing only the prompt instruction to permit multi-selection, with every other component identical. Under majority voting across three models, this achieves F1 = 0.555 in post-competition evaluation on released gold labels, exceeding the best official submission by 6.5 points. The entire gap is attributable to a single prompt instruction.

Recall loss decomposition. Of the 230 missed gold paragraphs in the submitted single-selection system, 122 (53%) are missed because the single-selection constraint prevents returning more than one paragraph per case even when the system has ranked multiple correct paragraphs highly. A further 61 (27%) are missed because the system selects an incorrect paragraph, and 47 (20%) are missed because the LLM returns no selection at all. The single-selection constraint accounts for the majority of the recall loss.

Under multi-selection, DeepSeek-V3 with few-shot prompting achieves F1 = 0.549, DeepSeek-R1 achieves F1 = 0.501, and DeepSeek-V3 without few-shot achieves F1 = 0.533, while the majority vote of all three (F1 = 0.555) exceeds every individual model, confirming complementary selection errors across the three models.

4 TASK 3: STATUTE RETRIEVAL AND ENTAILMENT

Statute retrieval and entailment combines two sources of difficulty: identifying the correct provisions from a compact but highly cross-referential code base, and then reasoning over those provisions with precise control of legal language. Task 3 requires retrieving relevant articles from the 725-article Japanese Civil Code and predicting entailment (Y/N) for 82 test queries from the bar examination. The training set comprises 965 queries from the Japanese bar examination, each annotated with the relevant article(s) and a Y/N entailment label.

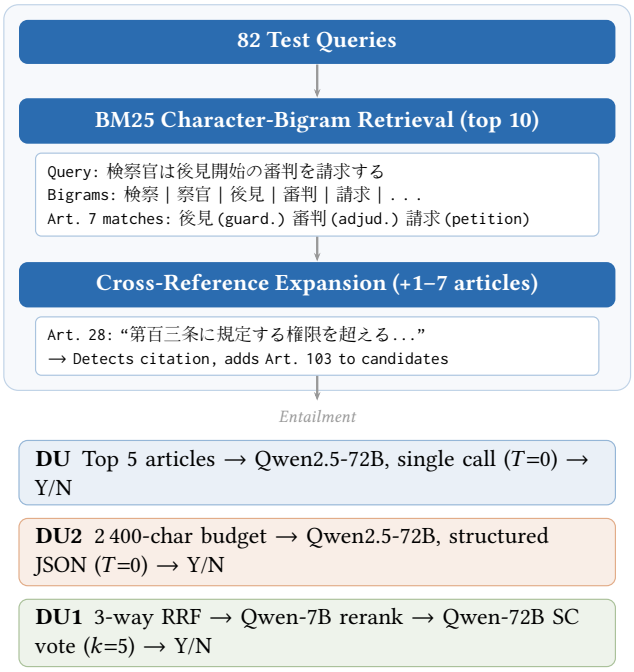


Figure 3: Task 3: BM25 matches queries and articles using character bigrams. Cross-reference expansion detects citations between articles. Three entailment configurations use Qwen2.5-72B with different prompting strategies.

4.1 Method

The system has two stages (Figure 3).

Retrieval: BM25 with character bigrams. Because Japanese text does not use spaces between words, standard BM25 cannot split text into tokens directly. Instead of relying on a morphological analyser, which can mis-segment specialised legal terms, we split every text into overlapping pairs of consecutive characters (character bigrams). For example, 後見開始 (commencement of guardianship) produces the bigrams 後見, 見開, 開始. BM25 [12] then matches queries and articles by shared bigrams and retrieves the top 10 candidates.

Cross-reference expansion. Civil Code articles frequently cite other articles by number. If BM25 retrieves an article that references another, we detect the citation using regular expressions and add the referenced article to the candidate pool, typically expanding from 10 to 11–17 articles per query. Cross-reference expansion captures legally relevant articles that lack surface-text overlap with the query.

Entailment. Three runs use Qwen2.5-72B-Instruct [15]: DU reads the top 5 articles in a single prompt, DU2 limits context to 2,400 characters with structured JSON output (achieving the best accuracy at 79.3%), and DU1 fuses BM25, TF-IDF, and BGE-M3 dense embeddings [2] via RRF [3] with self-consistency voting [14]. Despite its additional complexity, DU1 achieves the lowest accuracy

Table 5: Task 3 results: official submissions and post-competition analysis on the same retrieved articles.

Configuration	Correct	Acc. (%)	Rank
Official submissions (rank/22)			
NOWJ_run1	78/82	95.1	1
DU2	65/82	**79.3**	**14**
DU	63/82	**76.8**	**18**
DU1	62/82	**75.6**	**19**
Post-competition (same retrieved articles)			
Qwen3-235B (IRAC)	75/82	**91.5**	–
9-expert ensemble (DU3)	71/82	86.6	–

(75.6%), illustrating that pipeline complexity can degrade performance when the additional components introduce noise.

4.2 Results and Analysis

Table 5 reports the official results. DU2 achieves 65/82 correct (79.3%, rank 14 of 22 submissions). The competition winner, NOWJ, achieves 78/82 (95.1%). Retrieval recall ranges from 0.94 to 0.98 across the three runs, confirming that the entailment model rather than retrieval is the primary bottleneck. Error analysis reveals a systematic positive bias: of the 14 errors DU2 makes, 12 are false positives (wrong Y) and only 2 are false negatives (wrong N).

Ensemble prediction balance. The 9-expert ensemble applied post-competition predicts 42Y/40N on the 82 test queries, closely tracking the gold distribution of 41Y/41N and reducing the false-positive count from 12 to 6 compared to DU2's 51Y/31N prediction.

Post-competition ensemble. Table 5 reports results from applying our Task 4 ensemble (Section 5), the top-scoring system on the Task 4 leaderboard, to the same retrieved articles. Qwen3-235B with IRAC prompting achieves 91.5% (75/82 correct), while the nine-expert majority vote achieves 86.6% (71/82). The improvement from 79.3% to 91.5% represents a gain of +12.2 pp from replacing the model alone. Notably, the 9-expert ensemble (86.6%) underperforms the single Qwen3-235B model (91.5%) on Task 3, unlike Task 4, where the ensemble dominates. This reversal occurs because the Task 3 candidate pool includes distractor articles that confuse weaker ensemble members, and their incorrect votes dilute the signal from the stronger experts.

5 TASK 4: STATUTE ENTAILMENT

By removing the retrieval stage, Task 4 isolates the pure entailment problem and provides a controlled setting for analysing how different model families reason over statutory text. This task provides the relevant Civil Code articles and requires only the entailment decision (Y/N) for 82 test queries from the bar examination. Unlike Task 3, where the system must first retrieve the relevant articles, Task 4 isolates the entailment component by providing gold articles as input. The gold label distribution is approximately balanced, consistent across all exam years. This section presents our first-place system.

5.1 Expert Model Selection

We assemble nine expert configurations from three model families: DeepSeek [4], Llama [7], and Qwen [15], spanning two architecture types (Mixture-of-Experts and Dense Transformer). The selection criteria are: architectural diversity across families, complementary prompting strategies, and individually competitive zero-shot performance. The three model families collectively span parameter counts from 70B (Llama-3.3-70B Dense) to 671B (DeepSeek-R1 MoE), providing diversity not only in training data and architecture but also in model scale. Table 6 lists all nine configurations. Individual accuracies on the 258-question validation benchmark (H30+R01+R02) range from 80.2% to 84.9%, while the unweighted 9-expert majority vote (DU3) reaches 91.9% and the DU1 meta-ensemble built on top of it reaches 93.0%. Figure 4 illustrates the architecture.

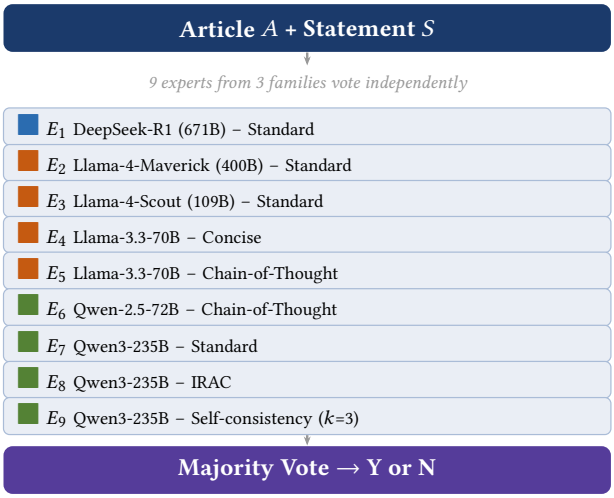


Figure 4: Task 4 base 9-expert ensemble (submitted as DU3; reused inside DU1's meta-ensemble). Three families — DeepSeek, Llama, Qwen — vote independently on Y/N.

Table 6: The 9 experts submitted as DU3 (also reused inside DU1). Validation accuracy (%); per-split in Table 8.

	Model	Arch.	Prompt	Val.
E_1	DeepSeek-R1 (671B)	MoE	Standard	84.1
E_2	Llama-4-Maverick (400B)	MoE	Standard	83.7
E_3	Llama-4-Scout (109B)	MoE	Standard	83.3
E_4	Llama-3.3-70B	Dense	Concise	80.2
E_5	Llama-3.3-70B	Dense	CoT Strict	80.6
E_6	Qwen-2.5-72B	Dense	CoT Strict	82.9
E_7	Qwen3-235B	MoE	Standard	84.1
E_8	Qwen3-235B	MoE	IRAC	84.5
E_9	Qwen3-235B	MoE	SC (k=3)	84.9
DU3 (9-expert majority vote)				**91.9**
DU1 (meta-ensemble, see Section 5)				**93.0**

5.2 Expert Instruction Design

Each expert receives a Japanese-language system prompt specifying the reasoning procedure. Table 7 lists the five prompt variants in both Japanese and English. The core *Standard* prompt instructs the model to: (1) identify the article's requirements precisely, (2) verify exception clauses, (3) check negation consistency, (4) examine quantifier expressions, and (5) determine entailment based

Table 7: Prompt variants for Task 4. All prompts include an explicit debiasing statement. Japanese originals shown with English translations.

Variant	Instruction
Standard	判断の手順：1.条文の要件・条件を正確に読み取る 2.例外規定・ただし書きを確認する 3.否定の整合性を確認する 4.量化表現・限定語を確認する 5.条文の論理的帰結として導かれる場合はYと判定する *(1) Read requirements precisely. (2) Check exceptions. (3) Verify negation consistency. (4) Check quantifiers. (5) Judge Y if the statement follows logically.*
Concise	条文のみを根拠に、陳述文が論理的に正しいかを判定してください。正しければY、正しくなければNを最後の行に出力してください。 *Based solely on the article, determine if the statement is logically correct. Output Y or N.*
CoT Strict	ステップ1：条文の要件を箇条書きで列挙 ステップ2：陳述文の主張を箇条書きで列挙 ステップ3：各要件との整合性を確認 ステップ4：ただし書き・例外を確認 ステップ5：限定語に注意 ステップ6：総合判定 *Step 1: List requirements. Step 2: List claims. Step 3: Check consistency. Step 4: Check exceptions. Step 5: Note quantifiers. Step 6: Judgment.*
IRAC	IRAC法を用いて分析：1.Issue（争点）2.Rule（法規範）3.Application（適用）4.Conclusion（結論） *(1) Issue. (2) Rule. (3) Application. (4) Conclusion.*
SC	Standard prompt, k=3 at T=0.7, majority vote.

exclusively on the provided text. The *Concise* prompt removes the multi-step instruction and asks for only the final label. The *CoT Strict* prompt enforces a rigid six-step chain-of-thought procedure. The *IRAC* prompt follows the standard legal reasoning framework (Issue, Rule, Application, Conclusion). The *Self-consistency* variant generates k=3 independent responses at temperature 0.7 and returns the majority answer [14]. All prompts include an explicit instruction that the label distribution is approximately 50% Y and 50% N, and that the model should not be biased towards either label. This debiasing statement reduces positive bias on the validation set. The choice of Japanese as the prompt language is deliberate: language-matched prompts avoid translation-induced errors in legal terminology.

5.3 Ensemble Construction and Submitted Runs

Three runs were submitted, layering progressively more aggregation logic on top of a shared pool of (model, prompt) experts.

DU3 is an unweighted 9-expert majority vote, with the expert subset (Table 9, "DU3" column) selected by exhaustive combinatorial search over the validation benchmark. DU3 achieves 91.9% on validation (Table 8) and 77/82 = 93.9% on the test set.

DU2 also uses a 9-vote majority, but two of the nine votes are deliberation-derived. When a base ensemble's vote margin is at most one, three judge models (Qwen3-235B, DeepSeek-R1 and Llama-4-Maverick) re-evaluate the question with detailed reasoning prompts; their unanimous consensus and majority consensus form two additional "deliberation experts" that participate in the final vote alongside seven direct experts. DU2 achieves 92.6% on validation and 79/82 = 96.3% on the test set.

DU1 (primary) is a hierarchical meta-ensemble: three sub-ensembles vote independently and the final DU1 prediction is the majority across their outputs. The sub-ensembles are Sub-A (8 experts plus a temporal-aware Qwen3-235B expert on R01 only; Table 9, "Sub-A" column), Sub-B (5 experts; "Sub-B" column), and Sub-C, which is architecturally identical to DU2. DU1 achieves 93.0% on validation and 79/82 = 96.3% on the test set — the highest accuracy among 33 submissions from 11 teams.

Table 8: Validation accuracy per split for each submitted run and the best single expert.

Run	H30	R01	R02	Overall
DU3 (plain 9-vote)	94.0	87.3	96.3	91.9
DU2 (9-vote w/ deliberation)	94.0	88.2	97.5	92.6
DU1 (meta-ensemble)	94.0	88.2	98.8	93.0
Best single (E_9)	91.0	78.2	88.9	84.9

Table 9: Expert membership: DU3 (plain 9-vote) and two of DU1's three sub-ensembles (Sub-A, Sub-B). DU1's third sub-ensemble is identical to DU2 (Section 5). Run accuracies in Table 8.

Model Family	Prompt	DU3	Sub-A	Sub-B
DeepSeek-R1	Standard	✓		
DeepSeek-R1	Meticulous			✓
Llama-4-Maverick	Standard	✓	✓	
Llama-4-Scout	Standard	✓	✓	
Llama-3.3-70B	Standard			✓
Llama-3.3-70B	Concise	✓		
Llama-3.3-70B	CoT	✓	✓	
Llama-3.3-70B	English		✓	
Qwen-2.5-72B	CoT	✓	✓	✓
Qwen-2.5-72B	IRAC			✓
Qwen3-235B	Standard	✓	✓	✓
Qwen3-235B	IRAC	✓	✓	
Qwen3-235B	SC (k=3)	✓	✓	
Total experts		9	8*	5

*Sub-A adds a temporal-aware Qwen3-235B expert on R01 only.

The best single expert (E_9, Qwen3-235B with self-consistency) achieves 84.9% overall, a gap of 8.1 pp below DU1, demonstrating the substantial contribution of cross-architecture voting and hierarchical aggregation. Plain 9-expert majority voting (DU3, 91.9%) outperformed all five learned combination methods tested (logistic regression, random forest, gradient boosting, SVM, MLP), which ranged from 85.3% to 90.3%. With only 258 validation questions, the training folds provide insufficient data for reliable parameter estimation.

5.4 Results

Table 10 reports the official test results. DU1 and DU2 both achieve 79/82 (96.3%), the highest accuracy among 33 submissions from 11 teams. The best competitor achieves 78/82 (95.1%). The three errors involve complex interactions between exception clauses and quantifier expressions. Two are false negatives where a majority of experts misinterpret a conditional exception beginning with ただし ("provided that") as a negation of the main rule. The third error is a false positive caused by a subtle quantifier distinction between すべての ("all") and いずれかの ("any") that none of the nine experts detects, resulting in a unanimous but incorrect Y judgment.

Table 10: Task 4 official test results (selected).

Rank	Run	Accuracy
1	**DU1**	**96.3%**
1	**DU2**	**96.3%**
3	JNLP-3	95.1%
3	IAIrun2	95.1%
6	**DU3**	**93.9%**

Table 11: Same-model versus cross-architecture ensemble accuracy on the 258-question validation benchmark.

Configuration	Accuracy
Best single expert	84.9%
Best same-model ensemble (Llama-3.3, 3 prompts)	87.2%
Best 5-expert cross-arch ensemble (search-selected)	91.9%
9-expert cross-arch plain vote (DU3)	91.9%
DU1 meta-ensemble (hierarchical)	93.0%

5.5 Cross-Architecture versus Same-Model Ensembles

Table 11 compares same-model and cross-architecture ensembles. The best same-model ensemble (Llama-3.3-70B with three prompts) reaches 87.2% on the validation set. A best-of-search 5-expert cross-architecture ensemble reaches 91.9%, the 9-expert plain vote (DU3) also reaches 91.9%, and the DU1 meta-ensemble built on top of the 9-vote layer reaches 93.0%, demonstrating a 4.7 pp gap attributable to cross-architecture diversity over same-model voting and a further 1.1 pp from hierarchical aggregation. Cross-family expert pairs disagree on 17.5% of questions versus 11.4% for same-family pairs, confirming that models from different families make more independent errors, which is the fundamental condition for effective majority voting.

5.6 Analysis

Expert removal. The most impactful single removal from DU1 is Llama-4-Maverick (−3.9 pp), followed by Qwen-2.5-72B (−3.5 pp). Removing all three Qwen3-235B variants reduces accuracy by 3.9 pp (from 91.9% to 88.0%), confirming that each model contributes to the ensemble's performance.

Error taxonomy. We analyse 4,825 reasoning traces from five models across all 965 training questions. Of these, 813 traces contain an incorrect final answer. We check whether each model's reasoning text engages with negation patterns, exception-clause markers, and quantifier expressions present in the article, using deterministic keyword matching. DeepSeek-R1 exhibits the highest rate of external knowledge errors (19.3% of its incorrect traces), meaning its reasoning frequently references Civil Code articles not present in the given input. Qwen3-235B has the highest negation detection failure rate (56.1%), meaning it frequently fails to mention negation constructions that appear in the article. Llama-3.3-70B is the only model with a false-positive bias (54.2% of its errors are false positives), while the other four models tend toward false negatives. Qwen-2.5-72B has the highest negation interpretation failure rate (71.1%), detecting negation patterns but misinterpreting their scope or application. These complementary error profiles explain why cross-architecture majority voting works: when Qwen3-235B fails to detect a negation, DeepSeek-R1 (with only 25.5% detection failures) is likely to detect it and provide the correct answer, and when DeepSeek-R1 introduces external knowledge, the other four models outvote it. Quantifier errors are corrected 73% of the time, while negation detection failures are corrected only 40%, as detection failures tend to occur across multiple models from the same family. 81% of the remaining ensemble errors occur on questions where both different families and different prompts disagree, identifying the hardest questions.

Fine-tuning comparison. We test whether supervised fine-tuning can match or exceed the zero-shot ensemble. Five fine-tuning configurations are evaluated on the 258-question validation benchmark, using up to 10,525 labelled examples (965 from Task 4 and 9,560 from Task 3). None reaches the best zero-shot expert's accuracy of 84.9%. The best fine-tuned result is ELYZA-8B with LoRA at 77.3%. The worst is Qwen-2.5-32B with QLoRA at 52.8%, barely above random, suggesting that larger models are more susceptible to overfitting when fine-tuned on limited legal data. This finding confirms that for statute entailment at this data scale, the broad legal reasoning capabilities encoded during pretraining are more valuable than task-specific adaptation.

6 PILOT TASK: LEGAL JUDGMENT PREDICTION

Legal judgment prediction is a multi-level problem, since case-level outcomes depend on interactions among multiple claims, counterclaims and factual findings rather than on a single local decision. The Pilot Task requires predicting court verdicts for Japanese tort cases (TP, evaluated by accuracy) and per-claim acceptance labels (RE, evaluated by macro-averaged F1). The training set contains 6,508 cases with 25,231 plaintiff claims and 22,458 defendant claims. The defendant wins in 60.5% of training cases, the median claim count per case is 4 (mean 7.3, maximum 197), and 20.3% of cases have no defendant claims at all. The test set contains 803 cases.

6.1 Method

The system has three stages (Figure 5).

Stage 1: Five claim-level models. Each model scores every claim with a probability between 0 and 1 indicating how likely the court is to accept it. The five models are designed to capture different aspects of claim acceptance.

Models 1 and 2: Jointly trained BERT classifiers. Each starts from a pretrained Japanese BERT encoder [5] (cl-tohoku/bert-base-japanese-v3 [13], 111M parameters) and adds two linear output heads: one for claim acceptance and one for the case verdict. The claim head receives the undisputed facts paired with an individual claim. The verdict head receives the full case text truncated to 512 tokens. Both heads are trained simultaneously with loss $\mathcal{L} = \mathcal{L}_{\text{claim}} + 0.5 \times \mathcal{L}_{\text{verdict}}$, where both terms are binary cross-entropy. Two instances are trained with different random seeds (42 and 7) to produce diverse predictions through convergence to different local optima. Training uses AdamW with learning rate 2×10^{-5} for the encoder and 2×10^{-4} for the heads, 10% warmup, gradient clipping at 1.0, batch size 8, up to 10 epochs with early stopping at patience 3. Out-of-fold RE F1 scores are in the range 0.689–0.693.

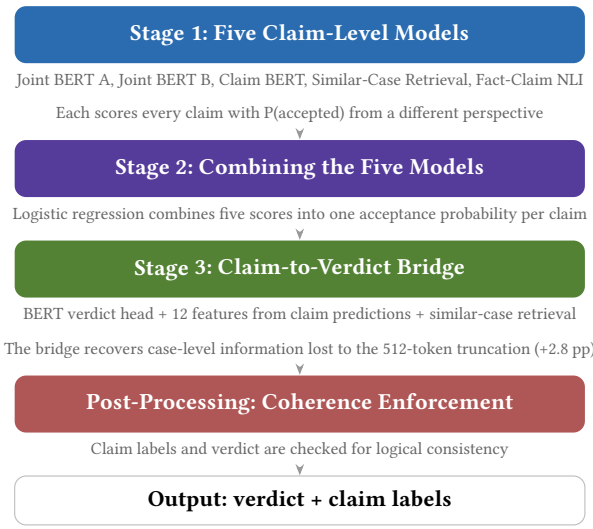


Figure 5: Pilot Task: five models score each claim from different perspectives (Stage 1). A logistic regression combines their scores (Stage 2). The verdict is refined using 12 claim-derived features (Stage 3), then checked for consistency.

Model 3: Claim-only BERT. This model uses the same architecture and input format as the claim head above but is trained exclusively on claim acceptance, without the verdict task. The three supervised models achieve similar claim-level F1 (0.689–0.693), suggesting that joint training benefits verdict prediction more than claim prediction through the shared encoder.

Model 4: Dense retrieval. Rather than learning to classify claims directly, this model finds the most similar claims in the training data and checks whether they were accepted. Qwen3-Embedding-0.6B (600M parameters) encodes each claim together with its case context into a dense vector. The acceptance score is the similarity-weighted average of the gold labels of the K=3 nearest training claims, combined equally with a TF-IDF character n-gram baseline (α=0.5).

Model 5: Textual entailment. A publicly available Japanese entailment model (akiFQC/bert-base-japanese-v3_nli, pre-trained on three NLI datasets by its authors) scores each claim against the case facts. The acceptance score is σ(entailment − contradiction), directly contrasting evidence for and against the claim. This model is the weakest alone (RE F1: 0.503), since legal claim acceptance involves factors beyond textual entailment such as credibility and proportionality. Its value lies in capturing a logical dimension that the other four models do not address: whether the facts support the claim as a matter of inference rather than pattern matching.

Alternative encoders were tested but underperformed cl-tohoku BERT. DeBERTa-v2-base-japanese reduced RE F1 from 0.689 to 0.648. ModernBERT-base (4,096 tokens), despite processing longer inputs, reduced TP accuracy from 0.708 to 0.690.

Stage 2: Combining the five models. A logistic regression combines the five scores into a single acceptance probability per claim, with weights learned from 5-fold cross-validation grouped by case. Table 12 shows the learned weights, which reflect how much we trust each model based on the validation data: the retrieval model receives the highest weight (1.856), meaning it has the most influence on the final prediction, while the NLI model receives the lowest (0.105) but non-zero, confirming a genuine contribution.

Table 12: Logistic regression weights for combining claim-level models.

Model	Weight
(4) Similar-case retrieval	1.856
(2) Joint BERT (seed B)	1.017
(1) Joint BERT (seed A)	0.831
(3) Claim-only BERT	0.618
(5) Textual entailment	0.105

Stage 3: Claim-to-verdict bridge. The BERT verdict head processes only 512 tokens, but 41% of training cases exceed this limit. To recover information lost to truncation, we construct twelve numerical features from the Stage 2 claim predictions (Table 13). These features capture the balance of evidence between the two sides at different levels of granularity. The mean acceptance difference (`re_mean_diff`) measures which side's claims are more likely to be accepted on average. The top-k differences (`re_top1_diff` through `re_top3_diff`) compare the strongest claims from each side, capturing whether the case has a single dominant argument or multiple supporting claims. The threshold-based counts (`re_count_03`, `re_count_05`) measure how many claims from each side pass a moderate or strong acceptance bar, distinguishing cases with many weak claims from cases with a few strong ones. The BERT verdict probability itself appears as a feature (`tp_prob`), allowing the classifier to combine the encoder's direct estimate with the claim-level evidence.

A gradient-boosted decision tree takes these twelve features and produces a refined verdict estimate. This bridge is the most impactful individual component in the entire pipeline, improving TP accuracy by 2.8 percentage points over the BERT verdict head alone. Coefficient analysis confirms that the BERT verdict score and the mean acceptance difference between plaintiff and defendant claims contribute most to the refined prediction. A TF-IDF retrieval component retrieves the three most similar training cases and blends their verdicts with the classifier's estimate (α=0.3). Since claim labels and the verdict are predicted independently, they can contradict each other, for example predicting a plaintiff win while accepting most defendant defences. Post-processing coherence rules resolve such contradictions: when the predicted verdict favours the plaintiff, no defendant defence may be newly accepted, and no more than six claim labels per party may change in a single case.

6.2 Results and Analysis

Table 14 reports leaderboard entries and our component ablation. The full system achieves 73.1% TP accuracy (587 of 803 cases) and 68.2% RE F1, scoring above all official entries on TP and matching the highest on RE.

Each stage provides a measurable improvement. A single BERT model establishes the baseline at 70.9% TP and 64.9% RE F1. Five-model stacking raises RE F1 to 66.9%, a gain of 2.0 pp that confirms the models capture complementary signals about claim acceptance.

Table 13: The twelve bridge features used in the claim-to-verdict classifier (Stage 3).

Feature	Description
`tp_prob`	BERT verdict score
`re_mean_diff`	Mean P acceptance prob. minus mean D
`re_margin_diff`	Confidence-weighted balance
`re_top1_diff`	Strongest P claim minus strongest D
`re_top2_diff`	Sum of top-2 P minus top-2 D
`re_top3_diff`	Sum of top-3 P minus top-3 D
`re_count_03`	#P claims $\geq$0.3 minus #D claims $\geq$0.3
`re_count_05`	#P claims $\geq$0.5 minus #D claims $\geq$0.5
`claim_count`	Number of P claims minus D claims
`total_claims`	Total number of claims (both sides)
`abs_margin`	|margin_diff| (case contentiousness)
`abs_tp_anchor`	|tp_prob $-$ 0.69| (distance from prior)

The claim-to-verdict bridge raises TP to 73.3% (+2.8 pp over the five-model configuration without the bridge), and case-level retrieval adjusts the final system to 73.1% TP and 68.2% RE F1. Among the twelve bridge features, `re_margin_diff` and `re_top1_diff` have the highest gradient-boosted feature importance.

The retrieval model receives the highest logistic regression weight (1.856), suggesting that outcomes of similar past claims are more predictive than features from fine-tuned classifiers. The NLI model receives the lowest weight (0.105) but is retained, confirming it contributes a unique logical entailment dimension. The claim-to-verdict bridge recovers information lost to the BERT verdict head's 512-token truncation (41% of cases are longer), and its +2.8 pp improvement confirms that claim-level predictions carry verdict-relevant information the encoder cannot capture alone. Our entry was inadvertently submitted in development mode rather than leaderboard mode, and the organisers subsequently included it on the results page as an unofficial submission.

7 DISCUSSION

The five tasks provide a compact view of how different modelling choices behave across retrieval, entailment and judgment prediction settings in legal NLP. Three cross-task patterns emerge from our five systems.

Classical retrieval remains competitive for legal documents. In Task 1, the LightGBM ranker with 34 domain features outperforms cross-encoders by 4.42 pp because documents (median 4,573 tokens) exceed encoder context windows. In Task 3, BM25 with character bigrams retrieves the correct article as the top result for 75 of 83 gold instances without supervised training.

Table 14: Pilot Task: top leaderboard entries (official and unofficial) and component ablation.

Configuration	TP Acc.	RE F1
Leaderboard (official entries plus our unofficial run)		
DU1 (ours, unofficial)	**73.1%**	**68.2%**
JNLP	72.7%	64.5%
TortNLP4 (UIT)	72.6%	68.2%
Component ablation		
Single BERT model	70.9%	64.9%
5-model RE, no TP bridge	70.5%	66.9%
5-model RE + bridge, no retrieval	73.3%	66.9%
Full system	73.1%	68.2%

Model choice matters more than pipeline complexity. Across Tasks 3 and 4, we observe that the choice of entailment model produces larger accuracy changes than any amount of pipeline engineering applied to a fixed model. In Task 3, replacing Qwen2.5-72B with Qwen3-235B (using an IRAC prompt) raises accuracy from 79.3% to 91.5%, a gain of 12.2 pp achieved by changing nothing except the model and prompt. By contrast, adding retrieval fusion across three systems, LLM-based reranking, and self-consistency voting to the original Qwen2.5-72B model reduces accuracy from 79.3% to 75.6%, a loss of 3.7 pp. In Task 4, cross-architecture diversity contributes 4.7 pp over prompt diversity within a single model. The error taxonomy from Task 4 confirms this: each model family has a distinct dominant failure mode, and majority voting succeeds because no single error type can accumulate a majority across diverse voters.

Development-set assumptions can severely limit test performance. In Task 2, a single-selection prompt calibrated on a development set where 80% of cases have one correct paragraph caps theoretical F1 at 0.508 on a test set averaging 2.94 correct paragraphs. Changing only the prompt instruction, from "select at most one" to "select all that entail," raises F1 from 0.343 to 0.555. The entire gap between the submitted result and the best official submission is attributable to this single instruction. This finding highlights the importance of validating task assumptions against diverse data splits, particularly in legal NLP where the number of relevant items per query can vary substantially between development and evaluation phases. In Task 3, a related phenomenon occurs: the gold labels are exactly balanced (41 Y, 41 N), but the submitted system predicts 62–66% Y, producing a one-sided error pattern. Both cases illustrate the risk of calibrating output distributions on development data that does not reflect the test distribution.

Tasks 3 and 4 provide a controlled comparison. These two tasks share the same 82 queries and the same Civil Code articles, differing only in whether the system retrieves articles (Task 3) or receives them from the organisers (Task 4). On Task 4, DU1 achieves 96.3%. On Task 3, the same ensemble on BM25-retrieved articles achieves 86.6%, a 9.7 pp gap not explained by retrieval failure (78/82 queries have all gold articles) but by irrelevant articles in the context confusing weaker ensemble members. Qwen3-235B alone achieves 91.5% on Task 3 with five retrieved articles, confirming that a strong model with a well-designed prompt tolerates moderate levels of irrelevant context.

Reproducibility and computational footprint. All systems use open-weight models accessed via published checkpoints, with no proprietary model used for any prediction. For Tasks 3 and 4, every model used was released before 15 July 2025, satisfying the competition cut-off. Inference for the Task 4 ensemble is embarrassingly parallel: nine experts each issue one independent forward pass per question on the 82-question test set, with E_9 adding two extra samples for self-consistency. For practitioners constrained by compute, the strongest single expert (E_9: Qwen3-235B with self-consistency) achieves 84.9% on the 258-question validation set, recovering a substantial part of the ensemble's gain at a fraction of the inference cost. Tables 6, 7 and 9 and the run descriptions in Section 5, together with the official COLIEE data, are sufficient to reconstruct each submitted run.

8 CONCLUSION

This paper presented a unified empirical study of all five COLIEE 2026 tasks through task-specific open-weight systems for retrieval, entailment, and legal judgment prediction. Our strongest performance is obtained on Task 4, where a cross-architecture ensemble of nine open-weight models achieves 96.3% accuracy and first place among 33 submissions, and on the Pilot Task, where a five-model pipeline with a claim-to-verdict bridge achieves 73.1% TP accuracy as an unofficial submission, scoring above all official entries on TP.

Post-competition analysis reveals improvements that require no change to the system architecture. For Task 2, changing a single prompt instruction from single-selection to multi-selection raises F1 from 0.343 to 0.555 in post-competition evaluation, exceeding the best official submission. For Task 3, replacing the entailment model with Qwen3-235B using an IRAC prompt raises accuracy from 79.3% to 91.5% on the same retrieved articles. Both findings were identified only after the gold labels were released, highlighting the value of thorough post-competition analysis.

The error taxonomy constructed from 4,825 reasoning traces in Task 4 provides evidence that cross-architecture diversity succeeds because each model family exhibits a characteristic failure profile. DeepSeek-R1 introduces external legal knowledge not present in the input, Qwen3-235B fails to detect negation patterns, and Llama-3.3-70B exhibits a false-positive bias. These complementary profiles mean that the majority vote corrects errors that no amount of prompt engineering within a single model family can address. This finding may generalise beyond legal entailment to other classification tasks where models from different training backgrounds are available.

ACKNOWLEDGEMENTS

Some experiments in this work used Durham University's NCC cluster, maintained by the Department of Computer Science.

REFERENCES

[1] Amal Saad Alshehri, Can Eken, Nelly Bencomo, and Amir Atapour-Abarghouei. 2026. Neural reranking for UK statutory retrieval: Provision-level evaluation and an open distilled model. *Artificial Intelligence and Law* (2026), 1–31.

[2] Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2024. BGE M3-Embedding: Multi-Lingual, Multi-Functionality, Multi-Granularity Text Embeddings Through Self-Knowledge Distillation. *arXiv preprint arXiv:2402.03216* (2024).

[3] Gordon V. Cormack, Charles L. A. Clarke, and Stefan Buettcher. 2009. Reciprocal Rank Fusion Outperforms Condorcet and Individual Rank Learning Methods. In *Proceedings of the 32nd International ACM SIGIR Conference*. 758–759.

[4] DeepSeek-AI. 2025. DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning. *arXiv preprint arXiv:2501.12948* (2025).

[5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of NAACL-HLT*. 4171–4186.

[6] Randy Goebel, Yoshinobu Kano, Mi-Young Kim, Juliano Rabelo, Ken Satoh, and Masaharu Yoshioka. 2025. An Overview of the COLIEE 2025 Competition: Legal Case Law and Statute Law Information Retrieval and Entailment. In *Proceedings of the Twentieth International Conference on Artificial Intelligence and Law (ICAIL)*.

[7] Aaron Grattafiori et al. 2024. The Llama 3 Herd of Models. *arXiv preprint arXiv:2407.21783* (2024).

[8] Yoshinobu Kano, Randy Goebel, Mi-Young Kim, Calum Kwan, Ken Satoh, Hiroaki Yamada, and Masaharu Yoshioka. 2026. An Overview of the COLIEE 2026 Competition: Legal Case Law and Statute Law Information Retrieval and Entailment. In *Proceedings of the 21st International Conference on Artificial Intelligence and Law (ICAIL)*.

[9] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. 2017. LightGBM: A Highly Efficient Gradient Boosting Decision Tree. In *Advances in Neural Information Processing Systems*, Vol. 30.

[10] Rodrigo Nogueira, Zhiying Jiang, and Jimmy Lin. 2020. Document Ranking with a Pretrained Sequence-to-Sequence Model. *Findings of EMNLP* (2020), 708–718.

[11] Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. *Proceedings of EMNLP-IJCNLP* (2019), 3982–3992.

[12] Stephen Robertson and Hugo Zaragoza. 2009. The Probabilistic Relevance Framework: BM25 and Beyond. *Foundations and Trends in Information Retrieval* 3, 4 (2009), 333–389.

[13] Tohoku University NLP Lab. 2023. BERT base Japanese v3. https://huggingface.co/tohoku-nlp/bert-base-japanese-v3.

[14] Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-Consistency Improves Chain of Thought Reasoning in Language Models. In *Proceedings of ICLR*.

[15] An Yang, Baosong Yang, Beichen Zhang, et al. 2024. Qwen2.5 Technical Report. *arXiv preprint arXiv:2412.15115* (2024).

SIL@COLIEE 2026: Lightweight Citation-Aware Legal Case Retrieval with Ensemble Scoring

Chetan Arora*
chetan@iitbhilai.ac.in
Indian Institute of Technology Bhilai
Bhilai, Chhattisgarh, India

Bhavya Jain*
bhavyaj@iitbhilai.ac.in
Indian Institute of Technology Bhilai
Bhilai, Chhattisgarh, India

Sarika Jain*
jasarika@nitkkr.ac.in
National Institute of Technology,
Kurukshetra, India
India

Abstract

We present the SIL system for COLIEE 2026 Task 1 (Legal Case Retrieval). Building on the observation that judicial citations are motivated by specific passages rather than global document similarity, our approach represents each query using compact citation-context windows — Reason for Citation (RFC) segments — extracted around citation markers. A BM25 index over candidate cases is queried independently with each RFC segment; the resulting ranked lists are merged via round-robin aggregation to form a candidate pool of up to 100 cases per query. Each query–candidate pair is then re-ranked using an ensemble of three MLP networks and three Random Forest classifiers trained on 27 semantic, lexical, and legal-metadata features adapted from prior work. The pipeline runs on CPU without LLM components, making it accessible to resource-constrained groups. Our best submission achieves a micro-F1 of 0.3871 (Precision 0.4186, Recall 0.3600), placing 7th among 54 submissions and 3rd among 22 teams on the COLIEE 2026 Task 1 leaderboard.

CCS Concepts

• **Information systems** → **Document retrieval**; • **Computing methodologies** → *Neural networks*; *Information extraction*; Supervised learning by classification; • **Applied computing** → Law.

Keywords

Legal case retrieval, Citation neighborhoods, Information retrieval, Semantic similarity, Machine learning, Ensemble methods, BM25 retrieval, Feature engineering

ACM Reference Format:
Chetan Arora, Bhavya Jain, and Sarika Jain. 2026. SIL@COLIEE 2026: Lightweight Citation-Aware Legal Case Retrieval with Ensemble Scoring. In *COLIEE 2026, June 12, 2026, Singapore.* ACM, New York, NY, USA, 6 pages.

1 Introduction

Legal case retrieval is a fundamental task in common-law judicial systems, where judges justify their decisions by citing earlier judgments. With the rapid expansion of legal databases, the need for efficient and accurate retrieval systems has become increasingly important, as legal professionals rely on access to relevant precedents to support arguments and judicial decisions. The Competition on Legal Information Extraction and Entailment (COLIEE) Task 1 formalizes this problem: given a new query case, the system must identify all cases that were "noticed", that is, cited within it [3].

Existing retrieval approaches increasingly rely on full-document representations, either through sparse methods like BM25 or dense encoders like MPNet or BERT variants. A fundamental problem with this approach is that full judgments contain large amounts of procedural, narrative, and introductory text that is unrelated to the reason a particular precedent is cited. A judicial citation is almost always motivated by a specific short passage of legal reasoning, not by global similarity between two documents [16]. Encoding an entire case, therefore, dilutes the citation signal considerably.

We address this by building our retrieval around Reason for Citation (RFC) segments —short windows of text surrounding each citation placeholder (e.g., `FRAGMENT_SUPPRESSED`) in the query case. These segments directly reflect the local context in which a judge invokes a precedent, and serve as focused, compact query representations. We build a BM25 index over candidate cases and retrieve top-ranked candidates independently for each RFC segment. We then construct a candidate pool of up to 100 documents per query by round-robin interleaving the ranked lists from all RFC segments and then classify each query–candidate pair using 27 hand-crafted features adopted from [18] with an ensemble of Multi-Layer Perceptron (MLP) and Random Forest (RF) classifiers. The contributions of this paper are as follows:

(1) A round-robin aggregation mechanism that ensures balanced candidate coverage across all citation contexts of a query, reducing the risk of dominant contexts overwhelming the pool.
(2) A citation-aware candidate generation strategy using raw RFC segments as focused query representations, without generative summarization, demonstrated to achieve competitive retrieval performance on a CPU-only pipeline — establishing a reproducible, resource-efficient baseline for future comparison.

The system is the SemIntLab group's entry for COLIEE 2026 and extends our earlier cascading retrieval framework [6], which used full-document BM25 candidate filtering followed by feature-based re-ranking. The present system replaces full-document filtering with citation-neighborhood RFC segments as focused query surrogates. To validate this design before the official submission, we applied it retrospectively to the COLIEE 2025 Task 1 dataset.

*All authors contributed equally to this research.

COLIEE 2026, June 12, 2026, Singapore

2 Problem Statement

Legal Case Retrieval (LCR) is primarily an information retrieval task with domain-specific challenges. The cases referenced by a query case are called noticed cases, which serve as decision-supporting precedents. The objective is to retrieve all noticed cases from a given collection, measured by how accurately the system identifies the relevant supporting cases [10].

Our system is evaluated within the COLIEE 2026 shared task framework, following the benchmark definition and evaluation protocol described in the official overview paper [7]. Formally, given a query case q and a set of candidate cases $C = \{c_1, c_2, \ldots, c_M\}$, $M \in \mathbb{N}^+$, the task is to identify the subset $S = \{r_1, r_2, \ldots, r_k \mid r_i \in C \wedge \mathit{support}(r_i, q)\}$, where $\mathit{support}(r_i, q)$ indicates that the case r_i supports the query case q in at least one legal aspect.

The COLIEE 2026 Task 1 corpus consists of Federal Court of Canada case laws provided in JSON format. The dataset includes a training split of 2,001 query cases drawn from a pool of 7,708 documents, and a test split of 400 query cases drawn from a pool of 1,848 candidate documents. Cases may appear as both queries and candidates in different instances. Performance is evaluated using micro-averaged precision, recall, and F1-score.

3 Related Work

Legal case retrieval has been an active area of research since its inception in the mid-1990s. Existing methods can be broadly categorised into traditional statistical approaches, neural language models, and hybrid architectures.

Early systems represented cases using hand-crafted features such as TF-IDF weights, n-grams, and bag-of-words vectors. When COLIEE introduced its Task 1 benchmark in 2018 [11], the leading approaches relied on BM25 for term-matching retrieval and simple similarity thresholds for relevance judgement [12]. While computationally efficient and interpretable, these methods captured only surface-level textual overlap and missed the deeper semantic and structural relationships that characterise legal citation behavior.

The introduction of transformer-based language models brought a significant shift. BERT-PLI [14] pioneered paragraph-level interaction modeling to manage the length of legal documents, performing pairwise comparisons between short segments of query and candidate cases rather than using document-level representations. This work established the two-stage retrieval paradigm that remains dominant today: a fast lexical first stage using BM25 to produce a manageable candidate set, followed by a computationally heavier neural re-ranking stage. Systems such as DoSSIER at COLIEE 2021 [1] demonstrated that combining Sentence-BERT embeddings with BM25 recall yields substantial improvements over either method alone.

By COLIEE 2022, hybrid retrieval strategies had matured into learning-to-rank (LTR) frameworks. LeiBi [2] aggregated multiple tuned embedding models alongside lexical features using LightGBM and Random Forest classifiers, showing that ensemble aggregation over diverse feature types is an effective strategy for balancing precision and recall. THUIR at COLIEE 2023 [8] took this further by incorporating structural knowledge such as citation graph information and document metadata into pre-trained language models, demonstrating that legal-domain signals improve retrieval beyond what surface text alone can offer.

More recently, large language models (LLMs) have been applied to both retrieval and entailment sub-tasks. At COLIEE 2024, teams such as CAPTAIN [5] used GPT-based models with chain-of-thought prompting for legal reasoning, while TQM [15] proposed multi-perspective re-ranking strategies that combine legal-domain pretraining with pairwise scoring. The ACL 2024 survey on legal case retrieval [4] confirms that LLM-based re-rankers now represent the state of the art, though they carry significant computational cost and are not accessible to all research groups.

A separate and complementary line of work focuses specifically on why legal citations are made, rather than on the broader similarity between cases. Wang et al. [16] observed that the sentences immediately surrounding a citation marker carry the bulk of the citation rationale and proposed isolating these sentences to improve retrieval precision. Liu et al. [9] extended this idea by constructing structured proposition graphs that link extracted citation rationales to specific statutory provisions and legal arguments, showing that modeling the logical hierarchy of legal reasoning improves matching. The UMNLP team at COLIEE 2024 [18] introduced citation-neighborhood propositions—compact declarative summaries of the legal principle associated with each citation—and highlighted the evaluation of raw neighborhood text as a direction for future work. Our system operationalizes this direction: we use RFC segments directly as query representations, without any generative summarization, keeping the pipeline lightweight while retaining the citation-level focus that prior work has identified as important.

4 SIL Methodology

Our system employs a two-stage pipeline. The first and primary stage is a citation-aware candidate generation mechanism that produces a focused pool of at most 100 candidates per query by exploiting the paragraph-level structure of judicial citations. The second stage re-ranks this pool using a supervised ensemble classifier trained on 27 hand-crafted features. Figure 1 gives an overview of the complete pipeline.

4.1 Preprocessing

The raw corpus contains a mix of English and French passages, citation placeholders such as `FRAGMENT_SUPPRESSED` and `REFERENCE_SUPPRESSED`, and heterogeneous formatting artefacts. We first detect the language of each paragraph using the `langdetect` library and retain only English paragraphs. The bilingual COLIEE documents generally contain English translations of French passages; however, we do not verify completeness of these translations and acknowledge that a small number of cases may contain legal reasoning expressed only in French that is consequently excluded. Each retained case is then tokenized into sentence units using spaCy, and headers, footers, and duplicated text markers are removed. Citation placeholders are preserved in place, as they serve as anchors for the citation context extraction step that follows.

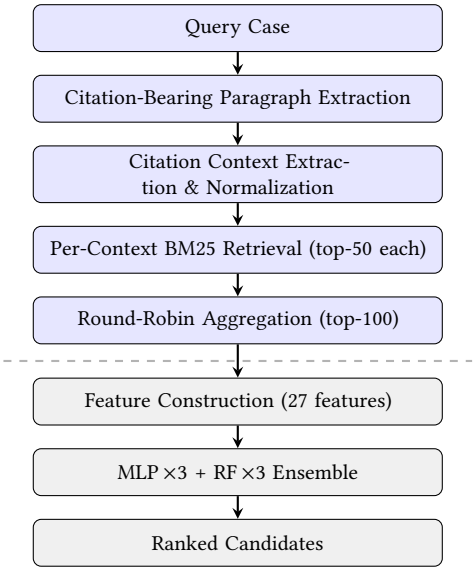


Figure 1: SIL pipeline for COLIEE 2026 Task 1 (Section 4). The upper stage (Sections 4.1–4.2, blue shading) is the citation-aware candidate generation component. The lower stage (Sections 4.3–4.6, grey shading) is the feature-based ensemble re-ranker. The dashed boundary separates the two stages. `MLP` ×3 and `RF` ×3 denote three independently seeded models of each type.

4.2 Candidate Generation

The central contribution of our system is a candidate generation strategy that replaces full-document retrieval with a structured, citation-driven process. The key observation is that relevance in COLIEE Task 1 is determined by explicit judicial citations rather than global document similarity: a candidate case is noticed by a query only because of a specific passage of legal reasoning within that query. Our candidate generation pipeline is designed to exploit this structure at every step.

Citation-Bearing Paragraph Extraction. For each query case q, we leverage the paragraph segmentation provided in the case files. We identify all paragraphs that contain explicit citation markers or suppressed reference placeholders, which signal a citation to a prior case. Let $P_q = \{p_1, p_2, \ldots, p_N\}$ denote the set of such citation-bearing paragraphs extracted from q, where each paragraph p_i corresponds to a distinct citation context. This paragraph-level decomposition is the foundation of our approach: rather than representing the entire query as a single retrieval unit, we treat each citation context as an independent query signal.

Citation Context Extraction and Normalization. A single citation-bearing paragraph may contain multiple suppressed references, each corresponding to a different cited case. To isolate individual citation contexts, we process each paragraph p_i to extract focused windows centred around a single citation marker. All citation markers (`FRAGMENT_SUPPRESSED`, `REFERENCE_SUPPRESSED`, `CITATION_SUPPRESSED`) are first normalized to a generic `REFERENCE` token. For each occurrence, a separate paragraph instance is created in which the corresponding reference is highlighted as `TARGETCASE`. A single paragraph containing k citations therefore yields k citation-specific contexts.

When a resulting context exceeds a predefined upper word limit, it is reduced to a window of 150–250 words centred around `TARGETCASE`. This window size follows the citation-context extraction design of Zhao et al. [18], who found that windows of this length are sufficient to capture the local legal reasoning surrounding a citation without including unrelated content from adjacent paragraphs. This reduction is performed in a sentence-aware manner: the sentence containing `TARGETCASE` is identified as the anchor, and surrounding sentences are symmetrically added before and after until the word limit is satisfied. If the anchor sentence itself is too long, trimming is performed within the sentence while preserving the target reference. Each extracted context is further normalized through token cleaning, removal of non-English text, stop-word removal, and stemming before being used for retrieval. Importantly, this citation-level processing is applied only to query cases. Candidate cases are indexed as full documents and remain unsegmented throughout retrieval, since the citation signal originates exclusively from the query side.

Per-Context BM25 Retrieval. Each normalized citation context is independently used as a query against the full corpus of candidate cases. We employ BM25 as the lexical retrieval function. For each citation context p_i, the top-$K = 50$ candidate cases are retrieved:

$$C_i = \mathrm{BM25}(p_i, C),$$

where C denotes the set of all candidate cases. While retrieval is driven by the paragraph-level citation context, the candidates are scored using their full document representations, ensuring that no information is lost on the candidate side.

Round-Robin Candidate Aggregation. The final candidate set for a query q is constructed by aggregating the per-context ranked lists $\{C_1, \ldots, C_N\}$. A naive score-based merge would allow candidates retrieved from a small number of dominant citation contexts to overwhelm the candidate pool, reducing diversity and increasing the risk of missing cases that are relevant to less prominent citation contexts. To prevent this, we adopt a round-robin aggregation strategy that ensures balanced coverage across all citation contexts.

The average query contains 14.0 ± 13.0 RFC segments, confirming that multi-context aggregation is non-trivial and meaningful for the majority of queries. The aggregation procedure iteratively traverses the ranked lists, selecting one candidate at a time from each list in descending BM25 order. A candidate is added to the aggregated set only if it has not already been selected, ensuring uniqueness. This process continues until either 100 unique candidates are collected or all lists are exhausted, producing the final top-100 candidate pool C_q for query q.

This aggregation design has two important properties. First, it prevents any single citation context from dominating the candidate pool. Second, it improves recall by preserving candidates supported by diverse, localized citation reasoning that would otherwise be discarded by a global score-based merge. The resulting candidate pool is both citation-aware and contextually diverse, and serves as the input to the downstream ranking stage.

Each query–candidate pair (q, c) in the training set is assigned a binary relevance label consistent with the task definition in Section 2:

$$\text{match}(q, c) = \begin{cases} 1 & \text{if } c \text{ is a noticed case in } q, \\ 0 & \text{otherwise.} \end{cases}$$

4.3 Embedding Generation

Dense vector representations are computed using all-MiniLM-L6-v2 [13], a sentence transformer model with 22 million parameters. Each candidate case is encoded at two granularities: at the sentence level and at the paragraph level. For query cases, embeddings are computed from the extracted citation contexts rather than the full document, maintaining consistency with the citation-aware retrieval stage. These embeddings are used to compute the semantic similarity features described below. The model was chosen for its feasibility on modest hardware: the full candidate corpus can be encoded on a standard CPU without GPU acceleration, albeit requiring approximately 22 hours (Table 1). This one-time offline cost is acceptable given that the embeddings are computed once and reused across all queries.

4.4 Feature Construction

For each query–candidate pair in the top-100 pool, we compute 27 features organized into five groups following the feature design introduced by UMNLP [18]. Together, these features capture multiple dimensions of case similarity that go beyond what BM25 term overlap alone can express, and allow the downstream classifier to distinguish genuinely noticed cases from superficially similar ones.

The first group comprises citation-context semantic similarity features (8 features). For every citation context of the query, we compute the maximum cosine similarity, Jaccard similarity, and overlap ratio against each sentence and paragraph in the candidate. Taking the maximum across all context–sentence pairs reflects the intuition that even a single strongly aligned passage is sufficient to motivate a citation, consistent with how judicial citations actually function.

The second group captures full-text lexical similarity (2 features) via Jaccard similarity over sentence word sets and TF-IDF cosine similarity over the concatenated case text. These features capture direct textual resemblance at the document level and complement the citation-context semantic features.

The third group consists of named-entity similarity features (3 features). We extract named entities from both the query and the candidate and compute Jaccard and TF-IDF similarity over the resulting entity strings. An additional binary same-case flag is set when both entity and lexical Jaccard scores exceed a threshold of 0.9, indicating potential near-duplicate or companion cases.

The fourth group contains legal metadata features (4 features). These include a binary year-check feature (set to 1 if the candidate case predates the query), a judge-match binary flag, a judge-pair ratio that captures how frequently the same pair of judges co-occurs in the training data, and a quotation-match flag that checks whether the leading tokens of any quotation in the query appear verbatim in the candidate text.

The fifth group is a sentence-level semantic histogram (10 features). Rather than summarizing sentence-to-sentence similarity with a single aggregate score, we construct a 10-bin histogram of cosine similarity values across all sentence pairs (q_i, c_j). This distributional representation captures whether semantic alignment is concentrated in a few high-similarity sentence pairs or spread broadly across both cases.

4.5 Ensemble Classifier and Training

Ensemble methods have proven to be a consistent source of improvement across COLIEE iterations. NOWJ at COLIEE 2023 [17] employed a multi-task ensemble that jointly optimized lexical, semantic, and structural objectives. The success of ensemble strategies across multiple years and teams reflects the complementary nature of sparse and dense retrieval signals in legal text, and motivates our choice to combine MLP and Random Forest classifiers over a rich feature set rather than relying on a single scoring model.

We adopt MLP and Random Forest as our base learners rather than gradient-boosted trees because Random Forest handles the discrete and binary features in our feature set (year-check, judge-match flags) naturally without requiring normalization, while MLP captures non-linear interactions among the continuous semantic similarity features. Together they address complementary aspects of the feature space. We train an ensemble combining three Multi-Layer Perceptron (MLP) networks and three Random Forest (RF) classifiers, each trained with a different random seed to promote model diversity. Each MLP consists of three hidden layers with 64, 32, and 16 neurons respectively, using ReLU activations and a sigmoid output layer, optimized with Adam (learning rate 10^{-3}, batch size 32, up to 50 epochs with early stopping on validation loss) and binary cross-entropy loss. Each Random Forest uses scikit-learn default settings (100 trees, `max_depth=None`, `min_samples_split=2`, Gini impurity criterion). All hyperparameters follow the defaults established by Zhao et al. [18], whose feature set and experimental protocol we adopt. Positive pairs are oversampled using random oversampling before training to address the severe class imbalance that naturally arises from the far greater number of non-noticed pairs. Each Random Forest operates on the same 27-dimensional feature vector; its tree-based structure handles the discrete and binary features such as the year-check and judge-match flags naturally and without requiring normalization.

The final relevance score for a pair is the average of the two sub-ensemble outputs:

$$\hat{y} = \frac{1}{2}\left(\frac{1}{n_{\text{MLP}}}\sum_{i=1}^{n_{\text{MLP}}} f_{\text{MLP}}^{(i)}(X) + \frac{1}{n_{\text{RF}}}\sum_{j=1}^{n_{\text{RF}}} f_{\text{RF}}^{(j)}(X)\right). \tag{1}$$

where $n_{\text{MLP}} = 3$ and $n_{\text{RF}} = 3$ are the number of MLP and Random Forest models respectively, each trained with a distinct random seed. $f_{\text{MLP}}^{(i)}(X)$ and $f_{\text{RF}}^{(j)}(X)$ denote the predicted relevance probability of the i-th MLP and j-th RF model for feature vector X. Averaging across independently seeded models reduces prediction variance and stabilizes results.

4.6 Inference

At test time, the trained ensemble outputs a relevance probability score for every query–candidate pair in the top-100 pool. We evaluate two inference strategies. In the first, Infer-1, for each query we select the two highest-scoring candidates. The choice of two reflects

Table 1: Approximate runtime per pipeline stage on a 16 GB RAM Intel Core i5 CPU (no GPU). Timings correspond to processing the COLIEE 2026 Task 1 corpus (7,708 candidate documents, 2,001 training queries, 400 test queries).

Stage	Approx. time
BM25 index construction and retrieval	∼2 hours
embedding generation	∼22 hours
Feature construction	∼10 hours
Ensemble training and inference	<1 hour

the average number of noticed cases per query in the COLIEE 2026 training set, which is approximately 2.0, consistent with findings reported by Zhao et al. [18] on prior COLIEE editions. A symmetric fallback is applied in both strategies: for every candidate case c in the pool, if no query has retrieved c as a positive, the query–candidate pair (q^*, c) with the highest predicted score $\hat{y}(q^*, c)$ is additionally selected as positive. This prevents systematic false negatives arising from candidates that score consistently below the selection threshold across all queries. In the second strategy, Infer-2, all pairs with a predicted probability above 0.65 are selected, with the same top-1 fallback applied for both queries and candidates. The threshold of 0.65 follows the value used by Zhao et al. [18], whose system achieved a top-3 rank at COLIEE 2024 with this setting; we adopt it without further tuning. Infer-2 typically yields higher recall by allowing multiple candidates per query when the model assigns them sufficiently high confidence.

5 Results and Discussion

Table 3 shows the official COLIEE 2026 Task 1 leaderboard results. The SIL team submitted two runs and placed 3rd out of 22 participating teams across 54 total submissions. submission_sil2 (Infer-2 thresholding) outperforms submission_sil1 (Infer-1 top-k) by 0.006 F1, driven by a precision gain of 0.033 at the cost of 0.017 recall, confirming that the 0.65 probability threshold is a meaningful control lever.

Proxy ablation via retrospective evaluation. A full within-paper ablation isolating each pipeline component individually is beyond the scope of this competition report. However, two sources of evidence support the value of the RFC-based candidate generation strategy. To further validate the design choice of RFC-based retrieval, we conducted a retrospective study on the COLIEE 2025 Task 1 dataset using progressively richer variants of the proposed pipeline. Table 2 compares our three retrieval configurations against the performance placed first in COLIEE 2025 Task1 by JNLP team *Full cases* uses the entire case text for retrieval and feature computation, representing a traditional document-level baseline. *RFC* replaces full documents with citation-context (Reason for Citation) segments, testing whether focused citation evidence is more effective than full-document matching. Finally, *RFC + ensemble (ours)* applies the proposed MLP+RF ensemble re-ranker on RFC-based features instead of MLP baseline, corresponding to the complete system used in this work. In particular, RFC retrieval already outperforms the full-case baseline (F1 = 0.3436 vs. 0.3103), indicating

Table 2: Retrospective evaluation on the COLIEE 2025 Task 1 test set.

Setting	Inf.	Precision	Recall	F1
JNLP	1	0.3267	0.2945	0.3667
	2	0.3353	0.3042	0.3735
Full cases	1	0.3030	0.3127	0.3078
	2	0.3272	0.2951	0.3103
RFC	1	0.3241	0.3655	0.3436
	2	0.3475	0.3491	0.3483
RFC + ensemble (ours)	1	0.3668	0.4053	0.3851
	2	**0.3961**	**0.3934**	**0.3948**

Table 3: COLIEE 2026 Task 1 official leaderboard results (top-12 of 54 submissions shown; full leaderboard available at [7]). SIL entries are shaded. Teams may submit multiple runs; 22 teams participated in total.

Team	Run	F1	Prec.	Recall
NOWJ	submission_2	0.4220	0.4235	0.4206
NOWJ	submission_3	0.4200	0.4140	0.4263
JNLP	random_forest	0.4126	0.4341	0.3931
JNLP	ensemble	0.4040	0.4174	0.3914
JNLP	xgboost	0.4000	0.4059	0.3943
NOWJ	submission_1	0.3880	0.3772	0.3994
SIL	submission_sil2	0.3871	0.4186	0.3600
SIL	submission_sil1	0.3810	0.3856	0.3766
mezza	task_1_mezzanino	0.3289	0.3161	0.3429
INTIT	3.intit_bm25_year	0.3165	0.3249	0.3086
DU	du3	0.3141	0.2945	0.3366
DU	du2	0.3136	0.2940	0.3360

that citation-focused retrieval is more effective than using entire judgments as queries. Combining RFC retrieval with the ensemble classifier achieves the best result (F1 = 0.3948), surpassing the top-performing COLIEE 2025 submission (JNLP, F1 = 0.3735). These findings motivated our adoption of RFC-based candidate generation as the foundation of the COLIEE 2026 system.

The task is competitive, with the top submission achieving an F1 of 0.4220. The gap between our best F1 (0.3871) and the top submission is only 0.035, which is a competitive margin given that our system uses no GPU and no large language model. Notably, `submission_sil2` achieves the third highest precision on the entire leaderboard (0.4186), behind only JNLP's `random_forest` (0.4341) and NOWJ `submission_2` (0.4235), indicating that our system is notably conservative and accurate when it does retrieve. The difference between our two runs reflects a precision–recall trade-off: `submission_sil2` gains 0.033 in precision at the cost of 0.017 in recall compared to `submission_sil1`, showing that the probability threshold of 0.65 is a meaningful lever for controlling this trade-off.

5.1 Limitations

Informal error analysis suggests that the candidate pool resulting from BM25 retrieval may not cover all noticed cases, particularly when the legal principle motivating a citation is expressed in abstract or paraphrased terms rather than through direct lexical overlap with the candidate text. In such cases, the relevant candidate may not appear in the top-100 pool at all, making recovery impossible during re-ranking. Among candidates that do appear in the pool, missed retrievals tend to occur when the legal principle is stated implicitly or through general doctrine rather than in a form that produces strong feature scores, leading the ensemble to assign them a low relevance probability. A further limitation of the experimental setup is that COLIEE 2026 does not provide a dedicated validation split; accordingly, all reported numbers reflect test-set performance on the official leaderboard, and we are unable to assess overfitting or perform threshold tuning on held-out data. This is a shared constraint for all COLIEE participants and is acknowledged as a limitation of competition-style evaluation.

6 Conclusion

The COLIEE 2026 competition has provided a valuable opportunity for the SIL team to investigate citation-neighborhood representations for legal case retrieval. Our system suggests that representing queries through RFC segments rather than full documents, combined with a feature-rich ensemble classifier, is an effective and resource-efficient approach. Our best submission achieves F1 = 0.3871 on the official leaderboard, placing 7th out of 54 submissions and 3rd out of 22 teams. The system runs entirely on modest CPU hardware, showing that competitive results do not necessarily require large-scale GPU infrastructure or LLM-based re-rankers, and serves as a strong, lightweight baseline for future comparison.

In future work, we intend to investigate legal-domain pre-trained encoders such as LegalBERT to strengthen the semantic similarity features. We also plan to explore better representations for candidate cases: the current system indexes candidates as full documents without removing noise, and constructing cleaner candidate representations could improve matching quality. Since our system deliberately avoids LLM-based re-rankers, a controlled comparison against an LLM re-ranking stage would quantify the exact performance–cost trade-off. Feature importance analysis of the 27-feature set (e.g., via permutation importance or SHAP values) would clarify which feature groups drive the ensemble's decisions and guide future feature engineering. Finally, we intend to incorporate citation graph structure by treating noticed-case relationships as edges in a graph, which may capture legal relevance patterns that purely textual features miss.

References

[1] S. Althammer et al. 2021. DOSSIER@COLIEE 2021: Leveraging Dense Retrieval and Summarization-Based Re-ranking for Legal Case Retrieval. In *COLIEE Workshop*.

[2] M. Askari et al. 2022. LeiBi@COLIEE 2022: Aggregating Tuned Lexical Models with Cluster-Driven BERT-Based Models for Case Law Retrieval. In *COLIEE Workshop*.

[3] COLIEE. 2026. COLIEE 2026 — Task 1: Legal Case Retrieval. https://coliee.org/tasks/task1.

[4] Y. Feng, C. Li, and V. Ng. 2024. Legal Case Retrieval: A Survey of the State of the Art. In *Proceedings of ACL 2024*.

[5] M. Fujita, T. Onaga, and Y. Kano. 2024. LLM Tuning and Interpretable CoT: KIS Team in COLIEE 2024. In *New Frontiers in Artificial Intelligence*. Springer.

[6] B. Jain, P. Harde, T. Sadikot, E. N. Kujur, and S. Jain. 2025. SIL@COLIEE 2025: A Cascading Framework for Finding Relevant Case Laws. In *Proceedings of the COLIEE 2025 Workshop*. Chicago, USA.

[7] Yoshinobu Kano, Randy Goebel, Mi-Young Kim, Calum Kwan, Ken Satoh, Hiroaki Yamada, and Masaharu Yoshioka. 2026. An Overview of the COLIEE 2026 Competition: Legal Case Law and Statute Law Information Retrieval and Entailment. In *Proceedings of the 21st International Conference on Artificial Intelligence and Law (ICAIL 2026)*.

[8] C. Li et al. 2023. THUIR@COLIEE 2023: Incorporating Structural Knowledge into Pre-trained Language Models for Legal Case Retrieval and Entailment. In *COLIEE Workshop*.

[9] X. Liu, J. Zhang, and V. Ng. 2023. Linking Legal Rationales and Factual Elements Through Structured Propositional Representations. In *Proceedings of ICAIL 2023*.

[10] J. Rabelo, R. Goebel, M.-Y. Kim, Y. Kano, M. Yoshioka, and K. Satoh. 2022. Overview and Discussion of the Competition on Legal Information Extraction/Entailment (COLIEE) 2021. *The Review of Socionetwork Strategies* 16, 1 (2022), 111–133.

[11] J. Rabelo, Y. Kano, T. Okada, et al. 2018. COLIEE-2018: Evaluation of the Competition on Legal Information Extraction and Entailment. In *CEUR Workshop Proceedings*.

[12] Juliano Rabelo, Mi-Young Kim, Randy Goebel, Yoshinobu Kano, Masaharu Yoshioka, and Ken Satoh. 2019. A Summary of the COLIEE 2019 Competition. In *Proceedings of the 6th Competition on Legal Information Extraction and Entailment (COLIEE 2019)*.

[13] Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. 3982–3992.

[14] Yiming Shao, Yansong Ma, Bingfeng Liu, et al. 2020. BERT-PLI: Modeling Paragraph-Level Interactions for Legal Case Retrieval. In *Proceedings of IJCAI 2020*.

[15] TQM Team. 2024. TQM@COLIEE 2024: Towards an In-depth Comprehension of Case Relevance in Legal Retrieval and Entailment. In *COLIEE Workshop*.

[16] H. Wang, Z. Sun, and C. Li. 2023. Reason-Aware Citation Modeling for Legal Case Retrieval. In *COLIEE Workshop*.

[17] J. Wang, M. Zhang, et al. 2023. NOWJ@COLIEE 2023: Multi-task Ensemble Learning for Legal Case Retrieval. In *COLIEE Workshop*.

[18] W. Zhao, R. Chen, et al. 2024. UMNLP@COLIEE 2024: Citation Neighborhoods for Capturing Legal Rationale in Case Retrieval. In *COLIEE Workshop*.

An Empirical Study on Hybrid Large Language Model Methods for Legal Information Retrieval and Entailment Tasks

Euijin Baek
H M Quamran Hasan
Yeji Kim
University of Alberta
Canada
{euijin1,hmquamra,yeji7}@ualberta.ca

Housam Khalifa Bashier Babiker
University of Waikato
New Zealand
housam.babiker@waikato.ac.nz

Mi-Young Kim
University of Alberta
Canada
miyoung2@ualberta.ca

Randy Goebel
Alberta Machine Intelligence Institute, Department of Computing Science
University of Alberta
Canada
rgoebel@ualberta.ca

Abstract

In this paper, we present the approaches developed by the UA team in the Competition on Legal Information Extraction and Entailment (COLIEE 2026). The UA team participated in Legal Case Retrieval, Legal Case Entailment, and Statute Law Entailment. For Task 1, we employ a hybrid retrieval pipeline combining sparse retrieval with LLM-based judging and supervised reranking. For Task 2, we investigate zero-shot LLM reasoning and supervised pair classification over hybrid retrieval candidates. For Task 4, we adopt a training-centric approach based on continued pretraining and parameter-efficient fine-tuning of a Japanese LLM. Our best results were an F1 score of 26.66% for retrieval, an F1 score of 42.54% for case entailment, and an accuracy of 92.68% for statute-law entailment.

CCS Concepts

• **Computing methodologies → Natural language processing; Heuristic function construction; Neural networks; Classification.**;

Keywords

legal textual retrieval, semantic text representation, document similarity, binary classification, imbalanced datasets

ACM Reference Format:
Euijin Baek, H M Quamran Hasan, Yeji Kim, Housam Khalifa Bashier Babiker, Mi-Young Kim, and Randy Goebel. 2026. An Empirical Study on Hybrid Large Language Model Methods for Legal Information Retrieval and Entailment Tasks. In *Proceedings of Workshop on Competition on Legal Information Extraction and Entailment (COLIEE 2026).* ACM, New York, NY, USA, 10 pages.

COLIEE 2026, Singapore

1 Introduction

Current Artificial intelligence applications like large language models (LLMs) are increasingly used in the legal domain and have attracted growing attention from both AI researchers and legal professionals [16, 39]. Legal NLP systems must often process large collections of case law, statutes, and other legal documents [4]. Despite recent progress in LLMs, legal natural language processing remains challenging because it requires accurate retrieval, fine-grained entailment, and reasoning over long, structured texts. Shared benchmarks are therefore essential for evaluating progress in realistic legal settings.

The Competition on Legal Information Extraction and Entailment (COLIEE) provides such a benchmark for legal retrieval and reasoning [29]. In COLIEE 2026, our team participated in Task 1, Task 2, and Task 4. Together, these tasks cover complementary aspects of legal NLP, including case retrieval, paragraph-level case entailment, and statute-law reasoning. Rather than using a single unified architecture, we have designed task-specific systems tailored to the different demands of each task. For Task 1, we use a hybrid retrieval pipeline with multi-stage candidate filtering, LLM-based judging, and supervised reranking. For Task 2, we investigate paragraph selection pipelines based on hybrid retrieval, zero-shot LLM reasoning, LLM-based candidate scoring, and pairwise classification. For Task 4, we adopt a training-centric approach based on continued pretraining and supervised fine-tuning.

2 Task Description

In COLIEE 2026, team UA participated in three tasks: Task 1, Task 2, and Task 4. In this section, we introduce the tasks.

2.1 Task 1

Legal Case Retrieval (Task 1) focuses on building systems that can accurately find supporting legal documents. Given a specific case,

the goal is to search a large database of candidates and pull out "candidate cases" that the original case should officially cite.

Since every legal situation is different, the number of relevant results isn't always the same. Previous submissions have shown that post-processing is a vital step [8] and helps rank the findings and reduce them to a final, high-quality shortlist.

For Task 1, the evaluation metrics consist of precision, recall, and F1-measure:

$$\text{Precision} = \frac{TP}{TP + FP}$$

$$\text{Recall} = \frac{TP}{TP + FN}$$

$$F_1\text{-measure} = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

where TP refers to True Positives (correctly retrieved candidates), FP refers to False Positives (incorrectly retrieved candidates), and FN denotes False Negatives (missed predictions).

2.2 Task 2

In this task, the system is given a base case, an entailed fragment from that case, and a noticed case that is already assumed to be relevant. The goal is to identify the paragraph or paragraphs in the noticed case that entail the queried fragment.

The evaluation metrics for Task 2 are the same as those used in Task 1, as described above.

2.3 Task 4

Task 4 is a Japanese statute-law entailment task. Given a yes/no legal query and the relevant Japanese Civil Code articles, the system must decide whether the proposition is confirmed (`Y`) or refuted (`N`) by the supplied statute text. Unlike Task 3, the organizers provide the gold relevant articles for each query, so Task 4 evaluates yes/no legal reasoning over fixed evidence rather than relying on accurate article retrieval. As specified by the organizers, the official evaluation metric is accuracy [1].

Because the evidence is already supplied, the main challenge is statutory reasoning rather than document retrieval. Many questions hinge on provisos, narrow conditions, negation, paragraph-level structure, and statute-internal references. The official task also forbids human intervention after the formal queries are released, so the system must handle these phenomena with a fixed, fully automatic pipeline. Task 4 therefore tests whether a system can follow fine-grained statutory logic rather than merely relying on lexical overlap between the query and the cited articles.

3 Related Work

3.1 Task 1

Prior submissions to COLIEE Task 1 show a clear transition from relatively direct retrieval baselines to hybrid systems that combine lexical matching, dense retrieval, reranking, and explicit modeling of legal structure. In COLIEE 2023, for example, Task 1 focused on retrieving relevant supporting cases for a new query case from a case law corpus, and the organizers noted that overall performance remained challenging, with generally low F1 scores across participating systems [10]. Among the strongest 2023 approaches, THUIR incorporated structural knowledge into pre-trained language models for legal case retrieval, thus illustrating the value of encoding legal document structure rather than relying only on surface-form similarity [18].

This direction became more pronounced in COLIEE 2024, where competitive systems increasingly adopted multi-stage retrieval and ranking pipelines. For example, the TQM [17] submission followed the general framework of the previous year's leading system by first reducing noise in case documents, then combining classical lexical retrieval with semantic retrieval, fusing multiple signals through learning-to-rank, and finally applying heuristic post-processing; their best official run achieved an F1 score of 0.4432. The COLIEE 2024 summary also showed TQM's continued participation and confirmed that hybrid retrieval architectures had become the dominant pattern for Task 1 submissions [11].

Submissions in COLIEE 2025 extended this line of work further by introducing stronger reranking methods, graph-based reasoning, and large language model components. Among the representative systems, NOWJ [21] proposed a multi-stage framework combining preprocessing, LLM-based summarization, and several retrieval and reranking components built around modern language models. UQLegalAI introduced *CaseLink*, a graph-based retrieval method that leverages relationships among cases and charges to improve relevance estimation [34]. The 2025 leaderboard further suggests that the best-performing systems *no longer rely on a single retrieval model*, but instead combine complementary retrieval signals and downstream reranking modules [9].

Our submission builds on prior years' findings and recent LLM developments to build a retrieval pipeline that integrates sparse retrieval methods, followed by LLM-as-a-Judge evaluation and reranking, motivated by the strong alignment between LLM and human evaluation [26, 38].

3.2 Task 2

Prior work on COLIEE Task 2 suggests that lexical matching remains a strong starting point for paragraph-level legal entailment. Nigam and Goel [24] combined Best Matching 25 (BM25), Sentence-BERT, and Sent2Vec in a cascading framework and reported that BM25 remained a strong baseline. Official COLIEE overviews [8] likewise show that traditional IR methods continue to play an important role as first-stage filters, even when stronger neural models are used later in the pipeline.

Zero-shot transfer has also proven competitive for legal case entailment. Rosa et al. [32] reported that zero-shot models achieved the highest scores in COLIEE 2021 and could be more robust than task-specific fine-tuning when labeled data are limited. Later COLIEE analyses and team reports similarly emphasize that sparse supervision, severe class imbalance, and distribution shift make this task difficult to solve with retrieval or supervised ranking alone; they also note that more parameters and stronger legal knowledge can improve performance [19].

More recent Task 2 systems increasingly adopt multi-stage pipelines that separate candidate generation from final entailment verification [21]. The official COLIEE 2025 overview [8] highlights the broader move toward multi-stage architectures, and a recent Task 2

system description reports that top-performing 2024 approaches relied heavily on BM25 pre-ranking, fine-tuned monoT5/BERT-style models, hard negatives, and LLM-based analysis over a top-k candidate set. This trend motivates our design, which likewise separates recall-oriented retrieval, structured candidate scoring, and final pair classification.

3.3 Task 4

Prior Task 4 systems mainly follow two directions: prompt-centric inference and training-centric adaptation. In the prompt-centric line, systems try to improve reasoning at test time with carefully designed instructions, few-shot exemplars, or prompt ensembles. In COLIEE 2024, CAPTAIN combined few-shot prompting, Auto-CoT, and data augmentation; NOWJ emphasized prompt collection and structured reasoning; and AMHR used a mixture-of-experts prompting scheme with weighted reconciliation [22, 23, 25]. These systems show that statute-law entailment can benefit from explicit reasoning scaffolds even without heavy task-specific retraining.

A second line of work moves most of the adaptation effort offline. The KIS 2025 Task 4 system paper reports official Swallow-70B formal runs with zero-shot, few-shot, and label-balanced few-shot prompts; the best of these, KIS3, achieved 67/74 on the official benchmark [27]. In contrast, our 2025 system explored a Japanese-first, training-centric pipeline based on continued pretraining, instruction tuning, and fixed-prompt inference, showing that much of the gain can come from domain and language alignment rather than from per-query exemplar retrieval [2].

These observations motivate our 2026 design. We again keep the runtime interface simple, use a fixed Japanese prompt, and place the main adaptation burden on the model and the training data. This lets us test whether the remaining errors are better addressed by stronger representation and supervision than by additional prompt variation.

4 Method

4.1 Task 1

Our submission builds on the same pre-processing pipeline Team UA followed last year [2]. During pre-processing, we remove procedural details, references, and HTML/XML tags, followed by chunking each document. Then the chunks are translated into English and then merged. The merged chunks are then summarized using a Qwen-3-8B [37] model.

Following the idea that a cited case must exist prior to the query case [17], we filter candidates with dates after the query case. To do this, we extract the latest date within each chunk using a Qwen-3-8B model. We subsequently compare all the dates among the chunks for a particular case and pick the latest date.

Run 1: Run 1 begins with using TF-IDF similarity on the preprocessed queries and candidate cases to retrieve the top 100 cases per query. Following the constraint that a query case cannot serve as a candidate case for other queries [17], we removed any query case from the pool of retrieved candidates. Then, TF-IDF similarity is applied on the summaries of the retrieved candidates and the queries to shortlist the candidate space to 50 per query. Following this, we use BM25 similarity on the query and candidates and reduce the candidate space to 20 per query. The date filtering logic is then applied to the retrieved candidates to ensure that a referred candidate exists prior to the query. Finally we use two LLMs, Llama-3.3-70B [12] and Qwen-3-32B [37], to determine whether the candidate should be cited by the query. The models were provided with the summaries of the cases. Following the self-consistency framework [36], we apply a voting logic where a candidate case is only retained when both models reach a consensus on its relevance. In cases where a consensus cannot be established, the system defaults to the judgment of the larger model (Llama-3.3-70B) to leverage its superior reasoning capacity and parameter scale.

Run 2: Run 2 follows the same pipeline as Run 1. However, we add one more LLM, NVIDIA-Nemotron-Nano-9B-v2 [3] to the majority voting pipeline. This time, we apply a majority voting logic where a candidate case is only retained if at least two out of the three models reach a consensus on its relevance (2-out-of-3 majority). Similar to Run 1, in cases where a consensus cannot be established, the system defaults to the judgment of Llama-3.3-70B.

Run 3: Run 3 utilizes the same candidate set retrieved from Run 2, and uses a supervised reranker based on the T5 architecture [30]. The model was fine-tuned on the COLIEE 2026 training set using a 1:1 sampling ratio of positive citations to random negative samples. At inference time, the model computes a relevance probability P by applying a softmax over the logits of the target tokens "true" and "false":

$$P(\text{relevant} \mid q, d) = \frac{e^{s_{\text{true}}}}{e^{s_{\text{true}}} + e^{s_{\text{false}}}} \tag{1}$$

A decision threshold of $\tau = 0.4$ (chosen heuristically) is applied to select the final citations.

4.2 Task 2

Our submission investigates three Task 2 runs under a shared paragraph selection framework. All runs first construct a high-recall candidate pool through hybrid retrieval. Run1 performs zero-shot selection over the retrieved candidates, whereas Run2 and 3 add LLM-based candidate scoring and make final predictions through binary classification over (fragment, paragraph) pairs. The three runs differ in the size of the candidate pool, the use of structured LLM scoring, and the final prediction model.

4.2.1 Data Preprocessing and Candidate Generation. To ensure a single English retrieval space and consistent model inputs, we normalize whitespace, remove paragraph identifiers, and translate all detected French text into English using the deep-translator package with the Google Translate [5] backend. Language detection is performed using both langdetect and langid.

All three runs begin with a hybrid retrieval stage. Given the query fragment and the paragraphs of the noticed case, we retrieve candidate paragraphs using both BM25 [31] and a BAAI General Embedding (BGE) [1]-based dense retriever, and then take the union of the two result sets. This hybrid design is intended to combine the strengths of lexical and semantic retrieval. BM25 is effective at finding paragraphs with strong surface-level lexical overlap, whereas dense retrieval can recover semantically relevant paragraphs even when the wording differs substantially from that

[1] https://huggingface.co/BAAI/bge-large-en

of the fragment. Using the union of the two result sets therefore improves early-stage recall and provides a more diverse pool of candidate evidence [21, 24].

4.2.2 Run1: Simple Hybrid Retrieval with Zero-shot Reasoning. We retrieve the top 5 paragraphs with BM25 and the top 5 paragraphs with BGE, and use their union as the final candidate set. We then present all candidate paragraphs jointly to LLaMA3.3-70B-Instruct [2] in a zero-shot setting. The prompt provides the target decision fragment and a numbered list of candidate paragraphs, and asks the model to select the paragraph or paragraphs that best entail the decision. We adopt a zero-shot setting since zero-shot prompting can be highly competitive for legal case entailment [32]. In addition, we deliberately restrict the candidate pool to a small size (top-5 from each retriever) to reduce potential confusion in the LLM. Providing too many candidate paragraphs can introduce noise and dilute the model's attention, making it harder to identify the most relevant supporting evidence [20]. All LLM-based inference in Run1, Run2, and Run3 is performed with temperature set to 0 to ensure deterministic outputs.

4.2.3 Run2: Expanded Retrieval, LLM-based Candidate Scoring, and Supervised Pair Classification. Run2 expands the candidate pool to improve recall by retrieving the top 10 paragraphs with BM25 and top 10 with BGE, and taking their union. While this improves coverage, it also introduces noise. To address this, we introduce an LLM-based candidate scoring stage using Qwen2.5-72B-Instruct [3], which evaluates how directly each paragraph supports the query fragment. We retain the top 7 candidates to preserve recall after reranking while keeping the final classification stage compact. This step is motivated by the observation that retrieval relevance does not necessarily imply entailment. In legal texts, many paragraphs may discuss similar topics or cite related authorities, but only a few directly support the specific proposition in the fragment [8, 19]. To better capture this distinction, we use a structured scoring scheme that decomposes support into multiple aspects, including direct support, specificity to the fragment, and whether the candidate alone provides sufficient justification, while penalizing paragraphs that mainly serve as background or authority. This is consistent with prior work showing that relevance judgments are inherently multi-criteria and that LLM-based evaluation benefits from explicit, task-specific criteria [7, 28]. In addition, to help interpret the candidate accurately without rewarding surrounding relevance, we provide the previous and next paragraph as context only and explicitly instruct the model to evaluate the candidate paragraph itself [15, 33]. The LLM is used as a semantic judge to assess these criteria, and the resulting scores are combined to rerank candidates. The final prediction is performed using a supervised pair-classification model based on DeBERTa-v3-base [4]. We cast the task as binary classification over (fragment, paragraph) pairs, where the model predicts whether a candidate paragraph supports the target fragment. Final predictions are obtained by thresholding (0.4 was chosen by validation set) the support probabilities assigned to the retained candidates.

[2]https://huggingface.co/meta-llama/Llama-3.3-70B-Instruct
[3]https://huggingface.co/Qwen/Qwen2.5-72B-Instruct
[4]https://huggingface.co/microsoft/deberta-v3-base

4.2.4 Run3: Expanded Retrieval, LLM-based Candidate Scoring, and PEFT-based LLM Fine-tuning. Run3 follows the same retrieval and candidate scoring pipeline as Run2, including hybrid retrieval and Qwen-based reranking to obtain the top 7 candidates, thus ensuring a high-quality and compact candidate set. The key difference lies in the final stage: instead of using a lightweight encoder classifier, Run3 fine-tunes Qwen2.5-32B-Instruct [5] with a sequence-classification head for the same binary (fragment, paragraph) prediction task. We adopt parameter-efficient fine-tuning (PEFT) [6, 14] and imbalance-aware training through controlled negative sampling and weighted losses. This setup allows the model to leverage richer representations and instruction-tuned reasoning capabilities.

4.3 Task 4

4.3.1 Method. Our Task 4 system follows the same training-centric direction that was promising in our 2025 study [2], but updates the base model and expands the internal evaluation protocol. We use Llama-3.3-Swallow-70B-Instruct-v0.4 as the base model. The model card lists 10 March 2025 as the release date, which satisfies the COLIEE 2026 rule that Tasks 3 and 4 may use only open models released before 15 July 2025 (JST) [35][6] . We adapt the model in two stages: domain-adaptive continued pretraining (Stage A) on Japanese Civil Code text, followed by parameter-efficient supervised fine-tuning (Stage B) for yes/no statute entailment.

Overview. Our design goal is to keep inference simple and reproducible while moving the main adaptation burden into training. This choice is consistent with our earlier observation that many residual Task 4 errors arise from condition parsing, exception handling, and reference resolution rather than from missing evidence [2]. Accordingly, we use a single fixed Japanese prompt at inference and focus our modeling effort on domain alignment and task-specific fine-tuning.

Data construction. We use only the organizer-provided 2026 Japanese Task 4 training materials. Stage A uses a Civil Code corpus serialized as 800 single-text pretraining records, each corresponding to one statutory unit. Stage B uses year-wise statute-entailment instances stored in a uniform supervised schema with fields `year`, `qid`, `instruction`, `input`, and `output`. For each instance, the organizer-provided gold article identifiers are resolved against the Civil Code corpus, and the resulting article texts are concatenated in the organizer-provided order inside `input`; the `output` field is always exactly `Y` or `N`. This yields 1,216 Task 4 fine-tuning examples in total. We keep all statute text and query text in Japanese during training and do not use any external retrieval corpus or manually curated exemplars.

Stage A: domain adaptation. We first continue pretraining the Swallow-70B model on the Civil Code corpus so that the model better captures statute-specific wording, article structure, and legal terminology. This step follows the general motivation of domain-adaptive pretraining, but is implemented here with LoRA (Low-Rank Adaptation)-based parameter-efficient adaptation [13, 14]. The objective of Stage A is not direct entailment supervision, but better alignment with Japanese statutory language before the Stage

[5]https://huggingface.co/Qwen/Qwen2.5-32B-Instruct
[6]https://coliee.org/COLIEE2026/rules

Table 1: English gloss of the Task 4 Stage B input format used by the submitted baseline. The actual runs use the original Japanese prompt and the full statute text; the example here is shortened for readability and LaTeX portability.

Field	Content
Instruction	Based on the following statutory text, answer Y if the proposition is true and N if it is false. Output only Y or N.
Input	**Statute**: Article 5 (Acts by a minor). A minor must obtain the consent of the statutory agent to perform a juridical act; however, this does not apply to an act that merely acquires a right or exempts the minor from an obligation. (2) A juridical act in violation of the preceding paragraph may be rescinded. **Query**: Even if a sales contract made by a minor lacks parental consent, it cannot be rescinded when the contract concerns ordinary daily-life matters.
Output	N

B fine-tuning stage. We did not submit a separate no-DAPT ablation, so in this paper Stage A should be interpreted as part of the submitted system recipe rather than as an independently isolated source of improvement.

Stage B: supervised fine-tuning. Beginning with the Stage A adapter, we fine-tune the model on Task 4 yes/no entailment examples. In the submitted baseline, the relevant articles are concatenated into a single evidence block, followed by the query statement, and the model is instructed to output only Y or N. We deliberately keep this template fixed across all evaluation settings so that improvements arise primarily from training and representation changes rather than prompt tuning. Table 1 shows an English gloss of the baseline input format.

Submitted run families. At inference time, we use two decoding modes. The first is deterministic greedy decoding. The second is sampling-based self-consistency with $n = 15$ generations, followed by majority voting over the resulting Y/N predictions [36]. Our three official runs are as follows. **Run 1** (submission tag UA1) uses the model family trained without year R06 and predicts with vote-15. **Run 2** (submission tag UA2) uses the same no-R06 family with greedy decoding. **Run 3** (submission tag UA3) uses a second model family trained on all available years through R06 and predicts with vote-15. This gives us both a conservative year-holdout family for model selection and a full-data family for the final submission.

Analysis variants. In addition to the submitted baseline, we evaluate three deterministic, analysis-driven variants. First, a *structured + cue-tagged* variant makes paragraph boundaries, condition markers, exception markers, negation cues, and reference cues more explicit in the input. Second, a *paragraph + one-hop reference* variant reformats the evidence by paragraph and appends one-hop statute-internal references when they are not already present among the gold articles. Third, a *targeted augmentation* variant duplicates selected training examples with paragraph/reference-oriented rendering for cases involving *junyo* (incorporation by reference), setoff, and local-reference plus condition/negation or exception/negation patterns. These variants are not separate submission families; they are evaluated to diagnose where additional gains might come from.

Table 2: Task 4 corpora and internal evaluation splits. "Train" and "Eval" denote the number of examples used in each setting.

Setting	Train	Eval
Civil Code DAPT corpus	800	–
All-years SFT corpus	1,216	–
Holdout R06	1,143	73
Formal H30	573	67
Formal R01	640	110
Formal R02	750	81

Only the baseline and paragraph-plus-reference variants were rerun on all formal H30/R01/R02 settings; the other variants are reported as holdout diagnostics.

4.3.2 Experiment Setup.

Corpora and year-based splits. All Task 4 experiments use only the organizer-provided 2026 Japanese materials. The Japanese Civil Code corpus is used for Stage A, while the year-wise entailment sets are converted into Stage B instruction-tuning examples. Development and formal-run evaluation use the same serialized schema as fine-tuning: a fixed instruction, a concatenated statute-and-query input, and a gold Y/N output. Because the official Task 4 labels were unavailable during system development, model selection relied on year-based held-out settings rather than random train/dev splits. First, we use a holdout split that trains on H18–H30, R01–R05 and evaluates on labeled R06 data. Second, following the COLIEE 2026 submission policy, we reproduce the required formal-run settings for H30, R01, and R02, where each target year is evaluated against all earlier years[7]. In the Results section, we report both the official leaderboard outcome and these internal development splits. Table 2 summarizes the resulting corpora and labeled splits.

Training and decoding configuration. All Swallow-70B runs use LoRA with rank 8 over all linear modules, a context length of 2048 tokens, and bf16 training under FSDP on two A100 80GB GPUs. For both Stage A and Stage B, the per-device batch size is 1 and gradient accumulation is 4, giving an effective batch size of 8 examples across two GPUs. Stage A uses cosine learning-rate decay with warmup ratio 0.03, learning rate 1×10^{-5}, and one epoch of continued pretraining on the Civil Code corpus. Stage B keeps the same distributed setup and scheduler family, but uses learning rate 1×10^{-4} and three epochs of supervised fine-tuning. The FSDP (Fully Sharded Data Parallel) configuration uses full sharding with auto-wrapping over LlamaDecoderLayer, CPU-efficient loading, and activation checkpointing. Greedy inference uses temperature 0. Vote-15 uses 15 stochastic generations with temperature 0.6 and top-p 0.9, followed by majority voting over Y/N. Unless a table or sentence explicitly refers to the official leaderboard, Task 4 accuracy values in this section are computed on labeled internal splits using the same accuracy metric as the official evaluation [1].

Comparison policy. We distinguish between model-selection runs and submission-only runs. The no-R06 family (UA1/UA2) is the main

[7] https://coliee.org/COLIEE2026/submission

comparative baseline because R06 is excluded from its training data and can therefore serve as a holdout year. The all-years family (UA3) is trained on all available years through R06 for final submission, but because R06 is included in its training set, we use it only for submission and sanity checking rather than for comparative model selection.

5 Results

5.1 Task 1

Table 3 reports the official COLIEE 2026 Task 1 results. This year, Task 1 had a total of 54 submissions, with Team UA securing 18th, 19th, and 22nd spots. The rankings of the runs indicate that our incremental additions did positively affect the scores. Run 1 used two LLMs as judges (F1 0.2576), Run 2 used three LLMs as judges (F1 0.2647), and finally Run 3 used a finetuned T5 reranker on top of the candidate set from Run 2 (F1 0.2666). We also observe that our approaches yielded very high recalls, scoring 7th position in terms of recall. This suggests that future work should focus on improving precision.

Table 3: Official COLIEE 2026 Task 1 results (Top 25 systems).

Team	Run	F1	Precision	Recall
NOWJ	submission_2	0.4220	0.4235	0.4206
NOWJ	submission_3	0.4200	0.4140	0.4263
JNLP	random_forest	0.4126	0.4341	0.3931
JNLP	ensemble	0.4040	0.4174	0.3914
JNLP	xgboost	0.4000	0.4059	0.3943
NOWJ	submission_1	0.3880	0.3772	0.3994
SIL	submission_sil2	0.3871	0.4186	0.3600
SIL	submission_sil1	0.3810	0.3856	0.3766
mezza	task_1_mezzanino	0.3289	0.3161	0.3429
INTIT	3.intit_bm25_year	0.3165	0.3249	0.3086
DU	du3	0.3141	0.2945	0.3366
DU	du2	0.3136	0.2940	0.3360
INTIT	1.intit_bm25_pys	0.3063	0.3309	0.2851
DU	du1	0.3035	0.2845	0.3251
FLNLP	task1_flnlpltr	0.2935	0.2619	0.3337
INTIT	2.intit_bm25_judge	0.2854	0.3147	0.2611
FLNLP	task1_flnlpembed	0.2709	0.2540	0.2903
UA	**ua_run3**	**0.2666**	**0.2038**	**0.3851**
UA	**ua_run2**	**0.2647**	**0.2056**	**0.3714**
UCD-CS	ucdcs3	0.2645	0.2480	0.2834
UCD-CS	ucdcs1	0.2607	0.2131	0.3354
UA	**ua_run1**	**0.2576**	**0.2114**	**0.3297**
UCD-CS	ucdcs2	0.2565	0.2405	0.2749
JUNLLP	task1_run2_results	0.2525	0.2193	0.2977
JUNLLP	task1_run1_results	0.2515	0.2369	0.2680

5.2 Task 2

Table 4 shows the official COLIEE 2026 Task 2 results. Among our three submissions, Run 1 achieved the best performance with an F1 of 0.4254, outperforming Run 2 (0.3721) and Run 3 (0.3529). Interestingly, the simplest zero-shot pipeline yielded the strongest result, suggesting that a compact candidate set with direct reasoning was more effective in our setting than the more complex reranking and fine-tuning variants. Run 2 achieved the highest precision among our submissions (0.5882) but at the cost of substantially lower recall, while Run 3 showed a more balanced but overall weaker trade-off. These results indicate that controlling candidate noise remains critical in Task 2.

Table 4: Official COLIEE 2026 Task 2 results (Top 10 systems and our submissions).

Team	Run	F1	Precision	Recall
IAI	task2_run2	0.4899	0.4501	0.5374
AIIRLab	task2_aiirfusion3	0.4706	0.5120	0.4354
JNLP	submission (3)	0.4507	0.4174	0.4898
IAI	task2_run3	0.4477	0.4769	0.4218
JNLP	submission (2)	0.4457	0.3879	0.5238
NOWJ	nowj003	0.4429	0.7037	0.3231
IAI	task2_run1	0.4424	0.4877	0.4048
AIIRLab	task2_aiirfusion2	0.4271	0.4257	0.4286
UA	**result_run1_ua**	**0.4254**	**0.4711**	**0.3878**
JNLP	submission (1)	0.4239	0.3381	0.5680
	Additional submissions from our team			
UA	result_run2_ua	0.3721	0.5882	0.2721
UA	result_run3_ua	0.3529	0.3592	0.3469

5.3 Task 4

Table 5 situates our submissions within the official COLIEE 2026 Task 4 leaderboard. The top score is 0.9634 (79 correct), achieved by DU1 and DU2; the next best score is 0.9512 (78 correct), achieved by JNLP-3 and IAIrun2/IAIrun3. Our best run, UA2, reaches 76 correct answers and 0.9268 accuracy, tying for rank 8 with JNLP-1, RUG-1, and RUG-2. UA1 and UA3 each achieve 75 correct answers and 0.9146 accuracy, tying for rank 12[8].

To interpret the official outcome, we also report the internal year-based results used during system development. Table 6 reports the development-time results for the submitted no-R06 baseline family (UA2 = greedy, UA1 = vote-15). On these internal splits, vote-15 is slightly better on holdout R06, but the difference between greedy and voting is small and inconsistent across years.

Table 6 shows two immediate patterns. First, the no-R06 baseline generalizes reasonably well to the newest labeled holdout year, which supports the use of year-based splitting for development. Second, R01 is the hardest internal split by a wide margin, suggesting that the remaining error types are not evenly distributed across years. At the same time, the official leaderboard shows that greedy UA2 slightly outperforms vote-15 UA1, whereas holdout R06 slightly favored vote-15; the all-years family UA3 also does not improve on the no-R06 family[9]. We therefore treat the internal splits as informative but imperfect predictors of final run ordering. For the all-years family, the labeled R06 sanity evaluation reaches 73/73, but since R06 is included in that model's training data, we treat that number

[8] https://coliee.org/COLIEE2026/results/task4
[9] https://coliee.org/COLIEE2026/results/task4

Table 5: Official COLIEE 2026 Task 4 results (top systems and our submissions).

Team	Run	Accuracy	Correct
DU	DU1	0.9634	79
DU	DU2	0.9634	79
JNLP	JNLP-3	0.9512	78
IAI	IAIrun2	0.9512	78
IAI	IAIrun3	0.9512	78
DU	DU3	0.9390	77
IAI	IAIrun1	0.9390	77
JNLP	JNLP-1	0.9268	76
RUG	RUG-1	0.9268	76
RUG	RUG-2	0.9268	76
UA	**UA2**	**0.9268**	**76**
Additional submissions from our team			
UA	UA1	0.9146	75
UA	UA3	0.9146	75

Table 6: Development-time internal Task 4 results for the submitted no-R06 baseline family (UA2 = greedy, UA1 = vote-15).

Split	Greedy	Vote-15
Holdout R06	65/73 (0.8904)	66/73 (0.9041)
Formal H30	61/67 (0.9104)	61/67 (0.9104)
Formal R01	84/110 (0.7636)	84/110 (0.7636)
Formal R02	77/81 (0.9506)	76/81 (0.9383)

Table 7: Task 4 analysis variants on labeled internal splits.

Variant	Holdout R06	Formal total
Baseline (vote-15)	66/73 (0.9041)	221/258 (0.8566)
Structured cues	66/73 (0.9041)	–
Paragraph + ref.	67/73 (0.9178)	220/258 (0.8527)
Targeted aug.	67/73 (0.9178)	–

as an in-sample check rather than evidence of generalization and exclude it from comparative tables.

To probe the remaining error modes, we evaluated several analysis-driven variants beyond the no-R06 vote-15 baseline. Table 7 summarizes the holdout and formal-run outcomes. The paragraph-plus-reference variant improves holdout R06, while the targeted-augmentation variant reaches the same holdout score only with checkpoint selection. The structured cue-tagging variant only matches the baseline. Because none of these variants improves the formal-run aggregate, we retain the simpler baseline family. This conservative choice is consistent with the official leaderboard, where greedy UA2 outperforms both vote-based submissions UA1 and UA3 [10].

Note. "Formal total" is reported only for variants rerun on H30, R01, and R02; "–" means that no corresponding formal rerun was executed. For checkpoint-sensitive variants, we report the best observed checkpoint rather than the final epoch.

[10] https://coliee.org/COLIEE2026/results/task4

Table 8: Error-category counts for greedy no-R06 baseline errors on labeled internal splits. Counts are non-exclusive because one query can trigger multiple tags.

Category	R06	H30	R01	R02	Total
Cross-reference	8	6	26	4	44
Condition	8	5	25	4	42
Negation	7	5	26	2	40
Multiple items/paragraphs	5	3	19	2	29
Exception/proviso	2	1	10	2	15
Junyo	2	1	6	0	9
Setoff	1	0	0	0	1

The exploratory variants are informative even when they are not selected as the final recipe. The paragraph-plus-reference variant improves R06 and slightly improves R01 (85/110 vs. 84/110), but loses ground on R02, so its formal-run aggregate remains just below the baseline. The targeted-augmentation variant can reach 67/73 on holdout R06, but regresses at the final epoch; the structured cue-tagging variant is less volatile, but its best checkpoint only ties the baseline vote result at 66/73. This split-wise instability is the main reason they were not promoted to the default submission family.

A final diagnostic comes from the confidence analysis. On holdout R06, direct scoring of $\log p(\texttt{Y})$ and $\log p(\texttt{N})$ matches the vote-based accuracy of 66/73, but with mean confidence 0.9835 and expected calibration error 0.1018. On R01, score-based accuracy drops to 83/110 while mean confidence remains 0.9737 and expected calibration error rises to 0.2192. On these internal splits, many errors are therefore high-confidence wrong answers rather than low-confidence undecided cases.

Table 8 summarizes the main error categories observed in the greedy no-R06 baseline across the labeled internal splits. The feature counts are not mutually exclusive: one wrong prediction can involve several legal phenomena at once.

One representative error illustrates why these categories are difficult. In a holdout R06 example about a non-written gift contract, the statute states that a non-written gift may be rescinded, but adds a proviso excluding portions that have already been performed. The query asks whether the donee may rescind after the object has already been delivered but before the burden has been performed. The gold answer is N, because the delivered portion falls under the proviso, but both greedy and vote-15 predict Y with unanimous votes. This example combines an exception clause, a condition, and negation, and shows that the remaining errors often require tracking the scope of a limiting proviso rather than recognizing the general rule alone.

6 Discussion

6.1 Task 1

The rankings provide a clear trend of how our increased complexity improved the F1 score over the runs. All runs use the same initially retrieved candidates before applying run-specific logic. Run 1 was the simplest approach using the agreement between two LLMs to get the final candidate set. Run 2 increased the complexity by

introducing another LLM, and this significantly boosted the Recall from 0.3297 to 0.3714. Finally, using the T5 reranker did improve the scores to some extent, but not significantly. We speculate the reason to be the usage of a base model which consists of only 250M parameters, which resulted in this small gain. The LLMs used were also capped at 70B, leaving room for improvements using bigger models.

This year's results show that a combination of sparse retrieval along with LLMs and reranking provides a valuable proof-of-concept for addressing COLIEE Task 1. Future work on Task 1 should utilize larger state-of-the-art models and finetune larger models for reranking to fully demonstrate the benefits of this approach.

6.2 Task 2

Our results provide several insights into the design of Task 2 systems. First, the best performance is achieved by Run 1, which uses the simplest pipeline. This suggests that a compact candidate set combined with direct zero-shot reasoning can be highly effective. Restricting the candidate pool appears to reduce noise and help the model focus on the most relevant evidence [20].

Second, expanding the candidate pool in Run 2 and Run 3 improves recall at the retrieval stage but introduces many hard negative examples. Although LLM-based candidate scoring reduces some noise, it does not fully eliminate weak or ambiguous candidates. As a result, the downstream classification task becomes more difficult. This is reflected in Run 2, which achieves higher precision but significantly lower recall, indicating a more conservative prediction behavior.

Overall, our findings indicate that controlling candidate quality is critical for Task 2, and that improvements in retrieval and filtering stages may be more impactful than increasing pipeline complexity.

6.3 Task 4

Official Task 4 results are now available. The best leaderboard score is 79 correct answers (0.9634), achieved by DU1 and DU2. Our best submitted run, `UA2`, achieves 76 correct answers and 0.9268 accuracy, tying for rank 8 with JNLP-1, RUG-1, and RUG-2; `UA1` and `UA3` each achieve 75 correct answers and 0.9146 accuracy, tying for rank 12 [11]. These official outcomes support our main conclusion that the training-centric baseline is strong, but they also sharpen one point: on the unseen official test set, greedy decoding generalized slightly better than vote-15, and the full-data `UA3` family did not improve over the no-`R06` family.

Table 8 and the concrete holdout example make the error profile more explicit. Across the 44 greedy errors made by the no-`R06` baseline on the labeled internal splits, all 44 involve explicit cross-references, 42 involve condition expressions, and 40 involve negation. Multiple-item or multi-paragraph reasoning appears in 29 cases, while exception/proviso cues appear in 15. The hardest split, `R01`, contributes 26 of the 44 greedy errors and contains nearly all of these phenomena. This quantitative pattern supports the same conclusion as our case-level inspection: the dominant failures are compositional statute-logic failures rather than missing-evidence failures or simple lexical mismatches.

The exploratory variants sharpen this diagnosis. Structured serialization with cue tagging makes legal markers more explicit, but its best checkpoint only matches the baseline holdout score of 66/73. Paragraph-level serialization with one-hop reference expansion improves holdout `R06` to 67/73 and slightly improves `R01` from 84/110 to 85/110, but its formal-run aggregate remains below the baseline (220/258 vs. 221/258) because it regresses on `R02`. Targeted augmentation reaches the same 67/73 holdout score only with checkpoint selection and does not show a stable advantage at the final epoch. Taken together, these variants suggest that richer structural rendering helps on some hard cases, but not yet in a way that is robust across year shifts.

The confidence analysis reinforces the same point from a different angle. On holdout `R06`, mean confidence remains 0.9835 despite an expected calibration error of 0.1018, and on `R01` mean confidence is still 0.9737 while expected calibration error rises to 0.2192. Many wrong predictions are therefore high-confidence mistakes rather than low-confidence undecided cases. This explains why simple confidence-thresholding or low-margin voting is unlikely to solve the main residual errors. The more plausible path forward is better training-time handling of condition scope, provisos, negation, paragraph structure, and statute-internal references.

Taken together, the official and internal results suggest a more specific practical lesson for Task 4. Our strongest reliable recipe is not a heavily engineered inference procedure, but a Japanese-first training pipeline consisting of one epoch of Civil Code continued pretraining, three epochs of LoRA-based entailment fine-tuning, and a fixed `Y/N`-only prompt at inference time. This recipe is strong enough for `UA2` to tie for rank 8 on the official leaderboard [12]. At the same time, because we did not isolate Stage A from Stage B in a separate ablation, we treat continued pretraining as a component of the final system design rather than as a standalone explanation for the result. The internal results also show that year-based model selection is useful, but not sufficient to predict final run ordering. The mismatch between the holdout preference for vote-15 and the official preference for greedy decoding makes that limitation explicit. Our ablations therefore point to a narrow, evidence-based future direction: improve structural handling of cross-references, conditions, negation, and multi-paragraph reasoning at training time, rather than relying on additional prompt engineering alone.

7 Conclusion and Future Work

We have presented our approaches to COLIEE 2026 for Legal Case Retrieval (Task 1), Legal Case Entailment (Task 2), and Statute Law Entailment (Task 4). Our methods employ LLMs in different contexts, including using them as judges, feature extractors, and end-to-end entailment classifiers. Our experimental results provided valuable insights that we can build on in future work. These include, but are not limited to, the importance of domain pretraining, as shown in Task 4; the usefulness of imbalance-aware training for learning discriminative representations; and the use of LLMs as a proxy for retrieval and re-ranking.

[11] https://coliee.org/COLIEE2026/results/task4

[12] https://coliee.org/COLIEE2026/results/task4

Acknowledgments

This research was supported by the Alberta Machine Intelligence Institute (Amii) at the University of Alberta, NSERC (grants DGECR-2022-00369 and RGPIN-2022-0346), and Alberta Innovates.

References

[1] 2026. COLIEE 2026 Competition Homepage. https://coliee.org/COLIEE2026/overview.

[2] Euijin Baek, Jiayi Dai, HM Quamran Hasan, Yeji Kim, Housam Babiker, Mi-Young Kim, and Randy Goebel. 2025. From TF-IDF to instruction-tuned LLMs: Hybrid legal reasoning systems for COLIEE 2025. In *Workshop of the Tenth Competition on Legal Information Extraction/Entailment (COLIEE'2025) in the 21th International Conference on Artificial Intelligence and Law (ICAIL)*.

[3] Aaron Blakeman et al. 2025. Nemotron-H: A Family of Accurate and Efficient Hybrid Mamba-Transformer Models. arXiv:2504.03624 [cs.CL] https://arxiv.org/abs/2504.03624

[4] Ilias Chalkidis, Abhik Jana, Dirk Hartung, Michael Bommarito, Ion Androutsopoulos, Daniel Katz, and Nikolaos Aletras. 2022. LexGLUE: A benchmark dataset for legal language understanding in English. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 4310–4330.

[5] Nicolas Constant. 2023. deep-translator: A flexible free and unlimited python tool to translate between different languages. https://pypi.org/project/deep-translator/.

[6] Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms. *Advances in neural information processing systems* 36 (2023), 10088–10115.

[7] Naghmeh Farzi and Laura Dietz. 2025. Criteria-Based LLM Relevance Judgments. In *Proceedings of the 2025 International ACM SIGIR Conference on Innovative Concepts and Theories in Information Retrieval (ICTIR)*. 254–263.

[8] Randy Goebel, Yoshinobu Kano, Mi-Young Kim, Calum Kwan, Juliano Rabelo, Ken Satoh, Hiroaki Yamada, and Masaharu Yoshioka. 2026. The COLIEE 2025 Competition on Legal Information Extraction and Entailment: Overview, Discussion, and Dataset Expansion. *The Review of Socionetwork Strategies* (2026), 1–31.

[9] Randy Goebel, Yoshinobu Kano, Mi-Young Kim, Calum Kwan, Ken Satoh, Hiroaki Yamada, and Masaharu Yoshioka. 2026. An Overview of the COLIEE 2025 Competition: Legal Case Law and Statute Law Information Retrieval and Entailment. In *Proceedings of the Twentieth International Conference on Artificial Intelligence and Law (ICAIL '25)*. Association for Computing Machinery, New York, NY, USA, 506–515. doi:10.1145/3769126.3785016

[10] Randy Goebel, Yoshinobu Kano, Mi-Young Kim, Juliano Rabelo, Ken Satoh, and Masaharu Yoshioka. 2023. Summary of the Competition on Legal Information, Extraction/Entailment (COLIEE) 2023. In *Proceedings of the Nineteenth International Conference on Artificial Intelligence and Law* (Braga, Portugal) *(ICAIL '23)*. Association for Computing Machinery, New York, NY, USA, 472–480. doi:10.1145/3594536.3595176

[11] Randy Goebel, Yoshinobu Kano, Mi-Young Kim, Juliano Rabelo, Ken Satoh, and Masaharu Yoshioka. 2024. Overview of Benchmark Datasets and Methods for the Legal Information Extraction/Entailment Competition (COLIEE) 2024. In *New Frontiers in Artificial Intelligence*, Toyotaro Suzumura and Mayumi Bono (Eds.). Springer Nature Singapore, Singapore, 109–124.

[12] Aaron Grattafiori et al. 2024. The Llama 3 Herd of Models. arXiv:2407.21783 [cs.AI] https://arxiv.org/abs/2407.21783

[13] Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't Stop Pretraining: Adapt Language Models to Domains and Tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Online, 8342–8360. doi:10.18653/v1/2020.acl-main.740

[14] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Liang Wang, Weizhu Chen, et al. 2022. Lora: Low-rank adaptation of large language models. *Iclr* 1, 2 (2022), 3.

[15] Di Jin and Peter Szolovits. 2018. Hierarchical neural networks for sequential sentence classification in medical scientific abstracts. In *Proceedings of the 2018 conference on empirical methods in natural language processing*. 3100–3109.

[16] Daniel Martin Katz, Dirk Hartung, Lauritz Gerlach, Abhik Jana, and Michael J Bommarito II. 2023. Natural language processing in the legal domain. *arXiv preprint arXiv:2302.12039* (2023).

[17] Haitao Li, You Chen, Zhekai Ge, Qingyao Ai, Yiqun Liu, Quan Zhou, and Shuai Huo. 2024. Towards an In-Depth Comprehension of Case Relevance for Better Legal Retrieval. arXiv:2404.00947 [cs.IR] https://arxiv.org/abs/2404.00947

[18] Haitao Li, Weihang Su, Changyue Wang, Yueyue Wu, Qingyao Ai, and Yiqun Liu. 2023. THUIR@COLIEE 2023: Incorporating Structural Knowledge into Pre-trained Language Models for Legal Case Retrieval. arXiv:2305.06812 [cs.IR] https://arxiv.org/abs/2305.06812

[19] Haitao Li, Changyue Wang, Weihang Su, Yueyue Wu, Qingyao Ai, and Yiqun Liu. 2023. THUIR@ COLIEE 2023: more parameters and legal knowledge for legal case entailment. *arXiv preprint arXiv:2305.06817* (2023).

[20] Nelson F Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024. Lost in the middle: How language models use long contexts. *Transactions of the association for computational linguistics* 12 (2024), 157–173.

[21] Hoang-Trung Nguyen, Tan-Minh Nguyen, Xuan-Bach Le, Tuan-Kiet Le, Khanh-Huyen Nguyen, Ha-Thanh Nguyen, Thi-Hai-Yen Vuong, and Le-Minh Nguyen. 2025. NOWJ@ COLIEE 2025: A Multi-stage Framework Integrating Embedding Models and Large Language Models for Legal Retrieval and Entailment. *arXiv preprint arXiv:2509.08025* (2025).

[22] Phuong Nguyen, Cong Nguyen, Hiep Nguyen, Minh Nguyen, An Trieu, Dat Nguyen, and Le-Minh Nguyen. 2024. CAPTAIN at COLIEE 2024: Large Language Model for Legal Text Retrieval and Entailment. In *New Frontiers in Artificial Intelligence*. Lecture Notes in Computer Science, Vol. 14741. Springer, Singapore, 125–139. doi:10.1007/978-981-97-3076-6_9

[23] Tan-Minh Nguyen, Hai-Long Nguyen, Dieu-Quynh Nguyen, Hoang-Trung Nguyen, Thi-Hai-Yen Vuong, and Ha-Thanh Nguyen. 2024. NOWJ@COLIEE 2024: Leveraging Advanced Deep Learning Techniques for Efficient and Effective Legal Information Processing. In *New Frontiers in Artificial Intelligence*. Lecture Notes in Computer Science, Vol. 14741. Springer, Singapore, 183–199. doi:10.1007/978-981-97-3076-6_13

[24] Shubham Kumar Nigam, Navansh Goel, and Arnab Bhattacharya. 2022. nigam@ coliee-22: Legal case retrieval and entailment using cascading of lexical and semantic-based models. In *Jsai international symposium on artificial intelligence*. Springer, 96–108.

[25] Animesh Nighojkar, Kenneth Jiang, Logan Fields, Onur Bilgin, Stephen Steinle, Yernar Sadybekov, Zaid Marji, and John Licato. 2024. AMHR COLIEE 2024 Entry: Legal Entailment and Retrieval. In *New Frontiers in Artificial Intelligence*. Lecture Notes in Computer Science, Vol. 14741. Springer, Singapore, 200–211. doi:10.1007/978-981-97-3076-6_14

[26] Shuai Niu, Jing Ma, Hongzhan Lin, Liang Bai, Zhihua Wang, Richard Yi Da Xu, Yunya Song, and Xian Yang. 2025. Knowledge-Augmented Multimodal Clinical Rationale Generation for Disease Diagnosis with Small Language Models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (Eds.). Association for Computational Linguistics, Vienna, Austria, 11011–11024. doi:10.18653/v1/2025.acl-long.540

[27] Takaaki Onaga and Yoshinobu Kano. 2026. KIS: COLIEE 2025 Task 4 Solver Using Japanese LLM. *The Review of Socionetwork Strategies* (2026). doi:10.1007/s12626-026-00209-w Published online 18 March 2026.

[28] Qian Pan, Zahra Ashktorab, Michael Desmond, Martin Santillan Cooper, James Johnson, Rahul Nair, Elizabeth Daly, and Werner Geyer. 2024. Human-Centered Design Recommendations for LLM-as-a-judge. In *Proceedings of the 1st Human-Centered Large Language Modeling Workshop*. 16–29.

[29] Juliano Rabelo, Randy Goebel, Mi-Young Kim, Yoshinobu Kano, Masaharu Yoshioka, and Ken Satoh. 2022. Overview and Discussion of the Competition on Legal Information Extraction/Entailment (COLIEE) 2021. *Journal of Review of Socionetwork Strategies* 16(1) (2022), 111–133.

[30] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2023. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. arXiv:1910.10683 [cs.LG] https://arxiv.org/abs/1910.10683

[31] Stephen Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: BM25 and beyond. *Foundations and Trends in Information Retrieval* 3, 4 (2009), 333–389.

[32] Guilherme Moraes Rosa, Ruan Chaves Rodrigues, Roberto de Alencar Lotufo, and Rodrigo Nogueira. 2021. To tune or not to tune? zero-shot models for legal case entailment. In *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Law*. 295–300.

[33] Xingyi Song, Johann Petrak, and Angus Roberts. 2018. A deep neural network sentence level classification method with context information. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. 900–904.

[34] Yanran Tang, Ruihong Qiu, and Zi Huang. 2025. UQLegalAI@COLIEE2025: Advancing Legal Case Retrieval with Large Language Models and Graph Neural Networks. arXiv:2505.20743 [cs.IR] https://arxiv.org/abs/2505.20743

[35] tokyotech-llm. 2025. Llama-3.3-Swallow-70B-Instruct-v0.4 Model Card. Hugging Face model card. https://huggingface.co/tokyotech-llm/Llama-3.3-Swallow-70B-Instruct-v0.4 Release history lists 10 March 2025; accessed 2026-03-21.

[36] Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-Consistency Improves Chain of Thought Reasoning in Language Models. arXiv:2203.11171 [cs.CL] https://arxiv.org/abs/2203.11171

[37] An Yang et al. 2025. Qwen3 Technical Report. arXiv:2505.09388 [cs.CL] https://arxiv.org/abs/2505.09388

[38] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena. In *Advances in neural information processing systems*. 46595–46623. arXiv:2306.05685 [cs.CL] https://arxiv.org/abs/2306.05685

[39] Haoxi Zhong, Chaojun Xiao, Cunchao Tu, Tianyang Zhang, Zhiyuan Liu, and Maosong Sun. 2020. How does NLP benefit legal system: A summary of legal artificial intelligence. In *Proceedings of the 58th annual meeting of the association for computational linguistics*. 5218–5230.

Optimization over Complexity: A Hybrid Multi-Stage Pipeline for Legal Case Retrieval

Marco Billi*
marco.billi3@unibo.it
University of Bologna
Bologna, Italy

Alessandro Parenti*
alessandro.parenti3@unibo.it
University of Bologna
Bologna, Italy

Giuseppe Pisano*
g.pisano@unibo.it
University of Bologna
Bologna, Italy

Marco Sanchi*
marco.sanchi4@unibo.it
University of Bologna
Bologna, Italy

Abstract

Legal Case Retrieval (LCR) is a core task in Legal Information Retrieval, aimed at identifying prior judicial decisions that are relevant to a given query case. In this work, we propose a hybrid multi-stage retrieval pipeline designed to balance effectiveness and computational complexity. Our approach combines a tuned lexical retriever (BM25+) with a lightweight neural re-ranking model based on a LoRA-finetuned cross-encoder. An initial retrieval stage narrows the candidate space, followed by a neural re-ranking step that captures deeper semantic relationships between cases. Additionally, we incorporate heuristic filtering based on temporal constraints to improve precision. To enhance training, we employ dynamically updated hard negatives, which strengthen the model's ability to distinguish relevant from non-relevant cases. Experimental results on the COLIEE Task 1 dataset demonstrate that our method achieves competitive performance while remaining suitable for resource-constrained environments.

CCS Concepts

• **Applied computing** → **Law**; • **Computing methodologies** → **Information extraction**.

Keywords

Legal Case Retrieval, Cross Encoder, Lexical Similarity, Hard Negatives, Coliee Competition

ACM Reference Format:
Marco Billi, Alessandro Parenti, Giuseppe Pisano, and Marco Sanchi. 2026. Optimization over Complexity: A Hybrid Multi-Stage Pipeline for Legal Case Retrieval. In *Proceedings of COLIEE (COLIEE 2026, June 12, 2026).* ACM, New York, NY, USA, 5 pages.

*All authors contributed equally to this research.

COLIEE 2026, June 12, 2026,

1 Introduction

Legal Case Retrieval (LCR) is a core task in Legal Information Retrieval, aimed at identifying prior judicial decisions that are relevant to a given query case. This task plays a role in supporting legal professionals by enabling efficient access to precedent cases that inform legal reasoning and decision-making. With the continuous growth of digital legal repositories, developing retrieval systems that are both accurate and computationally efficient has become increasingly important. In the context of the Competition on Legal Information Extraction and Entailment (COLIEE) [3], LCR is formulated as a judgment retrieval problem. Specifically, given a query case, the objective is to retrieve all *noticed cases*, i.e., those cases that are cited as supporting precedents. Unlike traditional keyword-based search, this task requires capturing complex semantic relationships and citation patterns within legal texts, making it particularly challenging.

Classical lexical retrieval methods such as BM25 remain strong baselines due to their efficiency and robustness. However, they often fail to fully capture semantic similarity between legal documents. On the other hand, neural approaches based on transformer architectures, such as cross-encoders, have shown improved effectiveness by modeling fine-grained interactions between queries and candidate documents, albeit at a higher computational cost.

To address these challenges, we propose a hybrid multi-stage retrieval pipeline that balances effectiveness and efficiency. Our approach combines a tuned lexical retriever (BM25+) with a lightweight neural re-ranking component based on a LoRA-finetuned cross-encoder. To further reduce computational overhead, we adopt a cascading strategy in which an initial retrieval stage narrows down the candidate set, followed by a re-ranking stage that refines the final results. Additionally, we incorporate heuristic filtering based on temporal constraints to improve precision.

The present paper showcases the methodology and results of the team `mezza` for the run `task_1_mezzanino` submitted at COLIEE 2026. Our contribution can be summarized as follows: (1) a hybrid retrieval framework that integrates lexical and neural ranking techniques in a computationally efficient pipeline; (2) a training strategy based on dynamically updated hard negatives to improve ranking performance; (3) a practical methodology for legal case retrieval that is suitable for local deployment with limited computational resources.

1.1 Task Description

The task investigates the performance of systems that search a set of case laws that support the unseen case law. The goal of the task is to return "noticed case" in the given collection to a query. A case is "noticed" to a query if the case is referenced by the query case. In this task, the references are redacted from the query case contents using expressions such as "FRAGMENT_SUPPRESSED" directly in the text. The dataset consists of legal decisions from the Federal Court of Canada and is divided into a training set, where query cases are paired with their noticed cases, a validation set for internal testing, and a final test set containing only query cases. The goal is to evaluate how effectively a system can recover the relevant supporting cases from the collection for the final test set. The corpus is given as a flat list of files containing all query and noticed cases, for both the training and validation datasets. These datasets are described in a json file containing a mapping between the query case and a list of noticed cases, as in the example below:

```
{
  "000001.txt": ["000005.txt", "012101.txt"],
  "012831.txt": ["000001.txt"],
  ...
}
```

The above is an example of a golden labels file for Task 1 containing three query cases (or "base cases"). The first query case is the file "000001.txt", which has 2 noticed cases ("000005.txt" and "012101.txt"). The second query case ("012831.txt") has only one noticed case: "000001.txt".

1.2 Evaluation Metrics

For this task the evaluation metrics used are precision, recall, and F-measure.

Precision. Precision is defined as the ratio between the number of correctly retrieved cases across all queries and the total number of retrieved cases across all queries:

$$\text{Precision} = \frac{\text{N of correctly retrieved cases for all queries}}{\text{N of retrieved cases for all queries}}$$

Recall. Recall is defined as the ratio between the number of correctly retrieved cases across all queries and the total number of relevant cases across all queries:

$$\text{Recall} = \frac{\text{N of correctly retrieved cases for all queries}}{\text{N of relevant cases for all queries}}$$

F-measure. The F-measure is the harmonic mean of precision and recall:

$$\text{F1} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

In the evaluation of this task micro-averaging is used. This means that the evaluation metrics are computed globally by aggregating the results over all queries, rather than computing the metrics for each query and averaging them (macro-averaging).

1.3 Datasets Description

We use the COLIEE Task 1 dataset for both training and validation. The dataset consists of legal case documents paired with their corresponding *noticed cases*, which serve as ground-truth labels.

For training the cross-encoder, we rely on the Task 1 training split, which contains 7231 documents (slightly fewer than the original 7350 due to the exclusion of fully non-English cases). Among these, 1678 query cases have redacted noticed cases. The distribution of relevant cases per query is reported in Table 1. On average, each query is associated with approximately 4 relevant cases, although the distribution is skewed, with some queries having significantly more associated precedents.

For internal validation, we use the 400 query cases from the Task 1 test set (out of 2159 total target entries). This subset is used for hyperparameter tuning and model selection. The corresponding distribution, also shown in Table 1, is consistent with the training data, supporting reliable validation.

The final evaluation is conducted on a separate test set of 1848 documents, of which 400 contain relevant noticed cases.

Set	Count	Mean	Std	Min	Median	Max
Training	1678	4.10	3.55	1	3	34
Validation	400	4.40	3.32	1	4	25

Table 1: Distribution of relevant (noticed) cases per query in the training and validation sets.

2 Related Work

LCR has been extensively studied within the broader field of Legal Information Retrieval, particularly in the context of benchmark initiatives such as COLIEE [1]. These shared tasks have played a key role in advancing retrieval methodologies by providing standardized datasets and evaluation protocols for identifying precedent cases.

Traditional approaches to document retrieval rely heavily on lexical matching techniques. Among these, BM25 [8] remains one of the most widely adopted methods due to its simplicity, efficiency, and strong baseline performance. Similarly, earlier probabilistic language modeling approaches [7] have contributed foundational techniques for estimating document relevance based on term distributions. However, such methods are limited in their ability to capture deeper semantic relationships between legal texts.

Recent research has increasingly focused on neural methods for legal retrieval. Transformer-based models, in particular Large Language Models (LLMs), have shown strong performance by finding contextual and semantic interactions between queries and documents [9]. The GTE ModernBERT model [10] represents a recent advancement in this direction, offering efficient text representations suitable for ranking tasks, and a great compromise between performance and computational complexity. These models are often used within re-ranking frameworks to refine the outputs of initial lexical retrieval stages.

Comprehensive surveys on case law retrieval [4] highlight the shift from purely lexical systems to hybrid approaches that combine symbolic and neural methods. Such hybrid pipelines leverage the efficiency of traditional retrieval techniques while incorporating the semantic understanding provided by neural models.

In line with these developments, our work adopts a hybrid retrieval strategy that integrates lexical retrieval with neural re-ranking, aiming to balance effectiveness and computational efficiency in this task.

3 Retrieval Pipeline

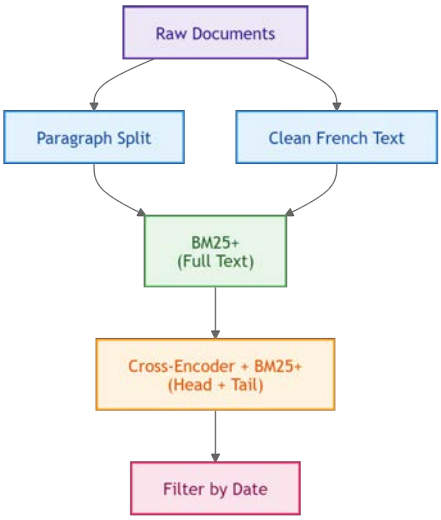


Figure 1: Retrieval Pipeline

Our system employs a multi-stage retrieval architecture as shown in Figure 1.

3.1 Preprocessing

Before scoring, documents are split into paragraphs, and French-language content is identified and removed to retain only the relevant English portions. Language identification is performed using the tool proposed in [5], which provides a set of pre-trained models for this task. This step is necessary because judgments in the dataset often include repeated legal and factual descriptions in French, reflecting the bilingual nature of Canadian courts.

3.2 BM25+

In the first stage, we use BM25+[6] to retrieve an initial set of candidate documents. BM25+ is an extension of the classical BM25 ranking function that introduces a small additive term δ to the frequency of the document term, which mitigates the underestimation of short documents. The BM25+ score for a query Q in document D is defined as:

$$\text{BM25+}(Q,D) = \sum_{i=1}^{n} \text{IDF}(q_i) \cdot \left[\frac{f(q_i,D) \cdot (k_1+1)}{f(q_i,D) + k_1 \left(1 - b + b \frac{|D|}{\text{avgdl}}\right)} + \delta \right]$$

where:

- $f(q_i, D)$ is the term frequency of query term q_i in document D,
- $|D|$ is the document length and avgdl is the average document length,
- k_1 controls term frequency saturation,
- b controls document length normalization,
- δ is the BM25+ additive smoothing constant.

For our experiments, we set $k_1 = 3$ and $b = 1.3$, which are higher than typical defaults ($k_1 = 1.5$, $b = 0.75$). A higher k_1 increases the impact of term frequency, allowing documents with repeated query terms to score more strongly. A higher b amplifies the effect of document length normalization, giving shorter documents less penalty and longer documents slightly more adjustment, which is useful in legal retrieval where relevant content may appear in brief paragraphs within long documents.

3.3 Re-ranking and Hybrid Scoring

For each query, the top $n = 20$ results are retained and passed to subsequent stages of the pipeline. This cut-off point has been chosen due to a compromise between expected recall and re-ranking performance. Each candidate is assigned a hybrid score ($H(Q,D)$), computed as a weighted combination of a cross-encoder score ($CE(Q,D)$) and the normalized BM25+ score ($\overline{\text{BM25+}(Q,D)}$):

$$H(Q,D) = \alpha \cdot CE(Q,D) + (1-\alpha) \cdot \overline{\text{BM25+}(Q,D)}.$$

To ensure that only temporally valid precedents are considered, we apply a post-processing step based on temporal consistency. Dates are extracted from candidate case texts using a regular expression designed to capture common Canadian date formats, including:

- ISO-style numeric dates: YYYY-MM-DD or YYYY/MM/DD,
- Numeric day-first or month-first dates: DD/MM/YYYY or MM/DD/YYYY,
- Textual month representations: e.g., "Jan 5, 2020" or "January 5 2020".

Extracted date strings are parsed into standardized datetime objects using a day-first convention. For each candidate document, we consider the *most recent date* mentioned in its text. This choice is motivated by the desire to ensure that any decision in the document does not occur after the query case, while also avoiding discarding useful documents unnecessarily. If a document has no identifiable date, it is retained to preserve recall, as some older cases may still be valid precedents even if dates are missing.

Formally, for a query case with most recent date d_{Q} and a candidate document with most recent date d_{D} (or missing), the candidate is retained if $d_{\text{D}} \le d_{\text{Q}}$ or d_{D} is unavailable. This procedure enforces real-world legal reasoning, ensuring that only past decisions, or documents without explicit dates, are considered as valid precedents.

A candidate is selected only if it exceeds a predefined hybrid score threshold τ and satisfies the temporal consistency condition:

$$f(Q,D) = \begin{cases} 1 & \text{if } \text{H}(Q,D) > \tau \text{ and } (d_D \le d_{\text{Q}} \text{ or } d_D \text{ is missing}), \\ 0 & \text{otherwise.} \end{cases}$$

The optimal hyperparameters α and τ are determined via 10-fold cross-validation on the validation set. For each fold, a grid search is performed over candidate parameter values, and the final parameters are obtained by averaging the best-performing values across all folds. In our experiments, we set $\alpha = 0.2$ and the threshold $\tau = 0.16$.

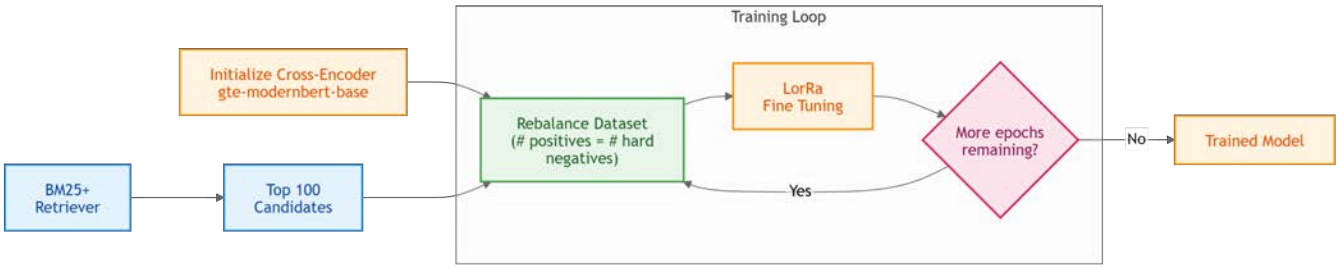


Figure 2: Cross-Encoder Training Pipeline

4 Cross-Encoder Model

We fine-tune the *Alibaba-NLP/gte-modernbert-base*[1] [10] model as a cross-encoder for binary relevance classification. A cross-encoder jointly encodes the query and candidate document, allowing the model to capture rich interactions between query and document tokens, which is especially important in the legal domain where subtle phrasing and factual details determine relevance. In contrast, a bi-encoder approach that encodes queries and documents separately may miss these fine-grained token-level dependencies, leading to lower ranking quality.

Due to the length of legal judgments and hardware limitations, we could not process the full document context. Instead, we adopt a *head–tail* strategy, selecting 384 tokens from the beginning and 128 tokens from the end of each document, with a maximum sequence length of 1024 tokens. The head portion often contains summary information, including a brief recap of the factual scenario, which is frequently useful for relevance estimation. Given the token limitations, attempting to select relevant sentences from the entire document, or from the middle sections, resulted in worse performance, likely because important sections were missed or the model focused on less relevant content. Although this segmentation strategy could be further improved, it yielded the best F1 score under our computational constraints.

To ensure computational efficiency, we apply LoRA (Low-Rank Adaptation) [2] with rank 16 for parameter-efficient fine-tuning. LoRA modules are applied to key attention and feed-forward layers. Training was conducted on a single NVIDIA T4 GPU.

The training dataset is balanced such that the number of positive and negative samples is equal, rather than relying on a weighted loss function, although it may not fully reflect the naturally imbalanced retrieval setting. This approach was chosen to ensure that the model receives a consistent learning signal from both relevant and non-relevant examples, avoiding potential instability or bias that can arise when using heavily weighted losses.

Hard negatives are drawn from a fixed global pool of candidates obtained from the initial BM25+ retrieval; this pool is not updated using predictions from the evolving model. This design simplifies training, reduces computational overhead, and ensures that the model is consistently exposed to unseen, but still lexically similar, non-relevant examples.

Within this fixed pool, the specific hard negatives sampled for each epoch are periodically varied to increase diversity in the training signal. In our setup, hard negatives are refreshed at intervals of 8, 4, 4, 4, and 4 epochs, meaning that while the pool remains constant, the subset of negatives used for training changes over time. This approach exposes the model to a wide variety of difficult negatives throughout training while maintaining a stable and controlled learning process. In total, the model is trained for 24 epochs. 4 Epoch training lasted approximately 3 hours and 20 minutes, thus full fine-tuning approximately 20 hours. The main training hyperparameters are summarized in Table 2.

Parameter	Value
LoRA rank	16
Max sequence length	1024
Batch size (per device)	64
Learning rate	2×10^{-4}
Weight decay	0.001
Num. epochs	24
Warmup ratio	0.1
Precision	FP16 / BF16

Table 2: Cross-encoder fine-tuning hyperparameters.

The overall training loop is illustrated in Figure 2, showing BM25+ retrieval, candidate selection, dataset rebalancing, and iterative LoRA fine-tuning.

5 Results & Discussion

We evaluate our system using the F1 score on the validation dataset. Table 3 summarizes the performance of different configurations.

Configuration	F1 Score
BM25 (default parameters)	0.12
BM25+ (tuned parameters)	0.24
BM25+ & Cross-encoder	0.31
BM25+ & Cross-encoder & Date filtering	**0.32**

Table 3: F1 performance on the validation dataset for different system configurations.

[1] https://huggingface.co/Alibaba-NLP/gte-modernbert-base

Tuning BM25+ parameters results in a substantial improvement over the default configuration. Incorporating the cross-encoder provides the second largest performance gain, while temporal filtering yields a minor further improvement.

Since ground truth labels are not available for the test set, we additionally analyze the distribution of predicted positive cases to assess generalization. Table 4 shows that the predicted distributions on the validation and test sets are highly consistent, with comparable means, standard deviations, and quartiles. This suggests that, at least from the perspective of label distribution, the model exhibits stable behavior across different splits and does not overfit to the validation set.

Set	Count	Mean	Std	Min	Median	Max
Validation	400	5.24	3.54	0	5	16
Evaluation	400	4.75	3.23	0	4	17

Table 4: Distribution of predicted positive cases for validation and test sets.

Table 5 presents the final results of our system in the competition. Our approach demonstrates consistent performance on the test dataset, achieving results comparable to those observed during validation. This indicates good generalization capability. In the official ranking, our team placed 4th overall out of 22 participants, while this system corresponds to the 9th submission out of 54 total[2].

	F1	Precision	Recall
Final Evaluation	0.3289	0.3161	0.3429

Table 5: Final performance of our system in the competition.

5.1 Limitations

Several limitations affect the proposed approach. First, hardware constraints impose a strict limit on the maximum sequence length, preventing full-document modeling (e.g., up to 8192 tokens). As a result, the head–tail segmentation strategy may overlook relevant information located in the middle sections of long documents.

Second, the dataset exhibits substantial variability in document length, ranging from very short to extremely long cases. This heterogeneity can negatively impact both lexical and neural components, as shorter documents may lack sufficient context, while longer ones are only partially observed.

Third, the reliance on lexical retrieval for initial candidate generation may limit recall, particularly in cases where relevance depends on deeper semantic understanding rather than surface-level term overlap. While the hybrid approach mitigates this issue to some extent, it remains a constraint of the overall pipeline.

Finally, beyond the specific approach used, the task itself suffers from an inherent limitation concerning the notion of what constitutes a "noticed case", as it is not entirely uniform across the dataset. Differences in writing styles, citation practices, and legal reasoning across judges and time periods introduce additional variability, making the retrieval task inherently more challenging.

[2]Team *mezza*, run *task_1_mezzanino*: https://coliee.org/COLIEE2026/results/task1 accessed on 24th of March 2026

6 Conclusion

Our findings highlight a pragmatic trade-off between performance and computational efficiency. While more advanced neural rankers could potentially yield higher performance, their training and inference costs remain prohibitive in constrained environments. The proposed pipeline demonstrates that combining classical retrieval methods with lightweight neural enhancements can achieve competitive results in legal retrieval tasks.

Future work may explore more sophisticated document segmentation strategies or efficient long-context models to better capture the structure of legal texts. More avenues of research may include exploring slightly imbalanced ratios or multiple hard negatives per positive example, to better represent dataset structure, as well as increasing computational cost to verify when the performance reaches a point of diminishing returns.

Acknowledgments

This work was supported by the Italian Ministry of University and Research (MUR) under the FIS project GenAI4Law, Grant No. FIS-2024-07323.

References

[1] Randy Goebel, Yoshinobu Kano, Mi-Young Kim, Calum Kwan, Ken Satoh, Hiroaki Yamada, and Masaharu Yoshioka. An overview of the COLIEE 2025 competition: Legal case law and statute law information retrieval and entailment. In Juliano Maranhão, editor, *Proceedings of the Twentieth International Conference on Artificial Intelligence and Law, ICAIL 2025, Chicago, IL, USA, June 16-20, 2025*, pages 506–515. ACM, 2025.

[2] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022.

[3] Yoshinobu Kano, Randy Goebel, Mi-Young Kim, Calum Kwan, Ken Satoh, Hiroaki Yamada, and Masaharu Yoshioka. An overview of the coliee 2026 competition: Legal case law and statute law information retrieval and entailment. In *Proceedings of the 21st International Conference on Artificial Intelligence and Law (ICAIL 2026)*, 2026.

[4] Daniel Locke and Guido Zuccon. Case law retrieval: problems, methods, challenges and evaluations in the last 20 years. *CoRR*, abs/2202.07209, 2022.

[5] Marco Lui and Timothy Baldwin. langid.py: An off-the-shelf language identification tool. In *The 50th Annual Meeting of the Association for Computational Linguistics, Proceedings of the System Demonstrations, July 10, 2012, Jeju Island, Korea*, pages 25–30. The Association for Computer Linguistics, 2012.

[6] Yuanhua Lv and ChengXiang Zhai. Lower-bounding term frequency normalization. In Craig Macdonald, Iadh Ounis, and Ian Ruthven, editors, *Proceedings of the 20th ACM Conference on Information and Knowledge Management, CIKM 2011, Glasgow, United Kingdom, October 24-28, 2011*, pages 7–16. ACM, 2011.

[7] Jay M. Ponte and W. Bruce Croft. A language modeling approach to information retrieval. In W. Bruce Croft, Alistair Moffat, C. J. van Rijsbergen, Ross Wilkinson, and Justin Zobel, editors, *SIGIR '98: Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, August 24-28 1998, Melbourne, Australia*, pages 275–281. ACM, 1998.

[8] Stephen E. Robertson and Hugo Zaragoza. The probabilistic relevance framework: BM25 and beyond. *Found. Trends Inf. Retr.*, 3(4):333–389, 2009.

[9] Marco Siino, Mariana Falco, Daniele Croce, and Paolo Rosso. Exploring llms applications in law: A literature review on current legal NLP approaches. *IEEE Access*, 13:18253–18276, 2025.

[10] Xin Zhang, Yanzhao Zhang, Dingkun Long, Wen Xie, Ziqi Dai, Jialong Tang, Huan Lin, Baosong Yang, Pengjun Xie, Fei Huang, Meishan Zhang, Wenjie Li, and Min Zhang. mgte: Generalized long-context text representation and reranking models for multilingual text retrieval. In Franck Dernoncourt, Daniel Preotiuc-Pietro, and Anastasia Shimorina, editors, *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 1393–1412. Association for Computational Linguistics, 2024.

ABAI at COLIEE 2026 Task 1: Multi-Stage Retrieval with GraphRAG-Enhanced Meta-Learning for Case Law Retrieval

Minhan Cho*
zvezda@g.skku.edu
AlphaBridge
Seongnam-si, Gyeonggi-do, Republic of Korea
Sungkyunkwan University
Seoul, Republic of Korea

Soyoung Park*
attorney.soyoung@gmail.com
National Assembly Research Service
Seoul, Republic of Korea
Sungkyunkwan University
Seoul, Republic of Korea

Daejin Choi†
djchoi@ewha.ac.kr
Ewha Womans University
Seoul, Republic of Korea

Jinyoung Han‡
jinyounghan@skku.edu
Sungkyunkwan University
Seoul, Republic of Korea

ABSTRACT

We present a multi-stage hybrid retrieval pipeline for COLIEE 2026 Task 1 (case law retrieval), submitted under team ABAI. The task requires identifying cited cases from a corpus of legal documents where citation passages are suppressed, creating a fundamental lexical gap between query and target. Our system addresses this through four independently trained stages: (i) multi-view BM25 with citation context windows and reciprocal rank fusion, (ii) neural reranking via a contrastive bi-encoder, BGE-M3 multi-signal retrieval, and a cross-encoder with selective truncation, (iii) graph-based reranking using entity-aware community detection (GraphRAG Lite) and a Graph Attention Network, and (iv) a LightGBM meta-learner that aggregates 34 features from all stages. We submitted three runs differing only in the meta-learner decision threshold: run 1 (balanced), run 2 (recall-favouring), and run 3 (precision-favouring). While our cross-validated F1 of 0.311 suggested competitive performance, our best submission (run 2) achieved F1=0.177 on the official test set, ranking 35th of 54 official runs and 15th of the 22 participating teams (by each team's best run). We present an ablation study showing the cross-encoder contributes 24% of the discriminative power of the pipeline, and analyze the substantial gap between cross-validation and test performance.

CCS CONCEPTS

• **Information systems** → **Information retrieval**; **Retrieval models and ranking**; **Language models**.

*Both authors contributed equally to this research.
†Corresponding author.
‡Corresponding author.

COLIEE 2026, June 12, 2026, Singapore

KEYWORDS

legal case retrieval, learning to rank, graph neural networks, community detection, meta-learning

1 INTRODUCTION

The great advance of the technology of artificial intelligence and the increasing amount of legal data have driven the improvement for diverse tasks in legal domains. One of the popular tasks is case retrieval, which searches for the relevant cases for a given query document, so that legal professionals can use and cite the retrieved cases in their final decision.

In line with this, the Competition on Legal Information Extraction and Entailment (COLIEE) 2026 Task 1 has been introduced [7]. The goal of the task is to find the cited cases from a set of case documents, given a query case document. In the 2026 challenge, the provided dataset consists of a corpus of 9,556 documents, where the actual cited cases have been replaced with the `<FRAGMENT_SUPPRESSED>` markers. This structural constraint severely limits the effectiveness of traditional lexical matching, as the direct textual overlap between a query and its gold-standard citations is inherently diminished. Consequently, robust systems must look beyond simple keyword matching and leverage both semantic representations and the latent structural relations within the legal corpus.

In this paper, we propose a multi-stage hybrid pipeline to mitigate the lexical gap caused by citation suppression. Our approach integrates four distinct components: (i) a multi-view BM25 retriever that employs citation context windows to capture local relevance signals, (ii) a neural reranking stage consisting of bi-encoders, BGE-M3, and cross-encoders, to capture deep semantic interactions, (iii) a graph-based reranking stage whose goal is to capture latent structural relationships between cases through shared legal entities and propagate relevance signals across the corpus graph, and (iv) a LightGBM meta-learner that aggregates 34 features extracted from three stages to decide whether each case should be predicted as a citation or not.

Our empirical evaluation shows a notable performance discrepancy: the cross-validated F1 score on the provided training data is 0.311 while the one on the official test set is 0.177, which ranked 35th

among 54 official runs (15th of the 22 participating teams by each team's best run). We conducted an ablation study for the proposed model, demonstrating that the cross-encoder plays a significant role in performance improvement (24% relative). The source code and submission materials are available at https://github.com/rabqatab/coliee2026_ABAI.

2 RELATED WORK

The increasing popularity of deep learning models has encouraged the research community to develop diverse models for case retrieval tasks [4]. A large portion of this research has focused on the structural characteristics of legal documents, identifying and exploiting the key parts for better retrieval. For example, Tran et al. [18] demonstrated that extracting lexical features from different parts of query and candidate documents improved retrieval over whole-document comparison. SAILER [9] suggested an asymmetric encoder-decoder model that encodes the fact section and reconstructs aggressively-masked reasoning and decision sections, forcing the encoder to capture cross-section dependencies. Similarly, Ma et al. [13] proposed Structured Legal Retrieval (SLR) framework, which first splits a legal document into functional segments (facts, reasoning, ruling), and then computes embeddings of each segment text. By incorporating an external charge-wise relation graph, which captures co-occurrence and hierarchical relationships between legal charges, the framework finally estimates the rank of a given document.

Recently, integration of graph-based models to prior work has been suggested. GraphRAG [16], which models and uses the relations among legal documents, has gained significant attention. Louis et al. [11] apply GNNs to statutory article retrieval, showing that propagating information through a citation graph improves dense retrieval by capturing inter-article dependencies. Zhang et al. [21] propose CFGL-LCR, a counterfactual graph learning framework for legal case retrieval that disentangles causal and shortcut features on case graphs.

Our framework differs from these prior studies in several important ways. First, instead of segmenting legal documents by legal function (e.g., facts, reasoning, and ruling), the proposed framework employs citation context windows around <FRAGMENT_SUPPRESSED> markers to form a retrieval-focused structural view. We also propose a GraphRAG Lite module that consists of two lightweight components: (i) regex-based entity extraction (statutes, judges, legal domains, outcomes) and (ii) Leiden community detection [17], which together capture latent structural relations without LLM overhead or explicit citation graphs. We use Graph Attention Network [19] to generate graph-level features.

3 METHOD

Figure 1 illustrates the overall architecture of the proposed framework, which consists of four stages. Stages 1 to 3 are responsible not only for choosing relevant cases, but also for extracting features from the given query and candidate cases, which are then fed into Stage 4 for the final decision. Note that the components in each stage are trained independently on different tasks.

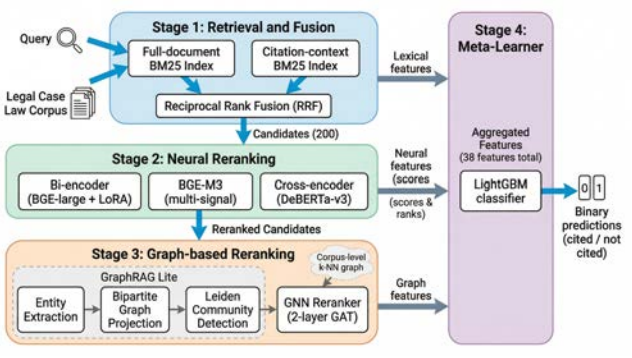


Figure 1: Architecture of the proposed multi-stage pipeline.

3.1 Stage 1: Retrieval and Fusion

Inspired by the observation that legal documents exhibit strong lexical characteristics dominated by legal terminology [4], we first design a lexicon-based case retrieval mechanism as the initial step for filtering case candidates. In particular, we employ BM25, which computes the similarity index between a pair of documents. Here, we calculate two indices: one for a pair of a query and a full case document, and the other for a pair of a query and a key part of a case document. We consider ±150 words around a <FRAGMENT_SUPPRESSED> marker as the key part. The rationale behind this choice is that the marker indicates the location where cited text was redacted, so the surrounding context contains the strongest relevance signal for identifying the cited case. Note that this approach extends the strategy proposed by Tran et al. [18], who suggested extracting lexical features from different structural parts of legal documents for improved retrieval, rather than comparing whole documents. Throughout the process, we select top 200 and 30 documents in terms of BM25 indices for full document and key part. Both indices are then fused by Reciprocal Rank Fusion [2] (RRF), which is robust to score distribution mismatches between heterogeneous retrieval systems. Formally, for a candidate document d ranked at position $r_i(d)$ under index $i \in \{\text{full}, \text{ctx}\}$, the RRF score is

$$\text{RRF}(d) = \sum_{i \in \{\text{full},\text{ctx}\}} \frac{1}{k + r_i(d)},$$

with $k = 60$. Top 200 candidate cases are finally selected and forwarded to Stage 2.

Computed Features. In addition to BM25 scores, we compute five lexical feature values (cosine similarity of TF-IDF vectors, Jaccard coefficient, shared bigrams, length ratio, and shared legal terms) for all query–document pairs based on token-level statistics. We also extract two citation-context features per candidate: the number of citation context windows whose BM25 top-30 results include that candidate, and the maximum BM25 score across those windows. Together, Stage 1 contributes 9 features to the meta-learner (Table 1, #1–9).

3.2 Stage 2: Neural Reranking

Stage 2 is responsible for reranking the candidate cases and extracting neural features. To this end, we design three components: a bi-encoder, BGE-M3, and a cross-encoder.

Bi-encoder. We fine-tune BGE-large-en-v1.5 [20] with LoRA [6] (r=16, α=32) as a *contrastive retrieval model* (trained independently from other stages). Given a query q, a positive document d^+ (a gold citation), and 7 hard negatives $\{d_1^-, \ldots, d_7^-\}$ sampled from the BM25 top-50 non-gold candidates, the per-query InfoNCE loss is

$$\mathcal{L}_{\text{NCE}}(q) = -\log \frac{\exp(s\ \cos(e_q, e_{d^+}))}{\exp(s\ \cos(e_q, e_{d^+})) + \sum_{i=1}^{7} \exp(s\ \cos(e_q, e_{d_i^-}))},$$

where e_q and e_d are the query and document embeddings produced by the LoRA-adapted encoder and s is the scale factor. The loss pulls the cosine similarity $\cos(e_q, e_{d^+})$ above that of every hard negative. Training uses AdamW for 3 epochs (lr=2e-5, batch size 16).

BGE-M3. To capture complementary retrieval signals beyond dense cosine similarity, we additionally score candidates using BAAI/bge-m3 [1] *without further fine-tuning*; the released checkpoint provides dense, sparse, and ColBERT similarity signals within a single model. The sparse signal recovers exact lexical matches that dense retrieval may miss, while ColBERT captures fine-grained token-level interactions without full cross-attention cost.

Cross-encoder. We select DeBERTa-v3-large [5] as our cross-encoder because its disentangled attention mechanism explicitly models content and positional information separately, which is beneficial for matching legal facts across structurally different case documents. It is fine-tuned as a binary classifier with cross-entropy loss using AdamW for 3 epochs (lr=1e-5, batch size 16). Because legal documents (2,000–5,000 words) far exceed the 512-token input limit, naive truncation discards most of the document. We therefore use *selective truncation*: for each query-candidate pair, we concatenate document head/tail (100 words each) with the top-5 citation context windows and top-10 paragraphs by TF-IDF similarity, within a 500-word budget. This prioritizes the passages most likely to contain citation-relevant information over arbitrary positional truncation, following the intuition from context-aware passage selection [3] and structured legal retrieval [13].

Computed Features. Each neural model produces a relevance *score* and a within-query *rank* (the ordinal position when all candidates for the same query are sorted by that model's score). The rank is a derived feature computed after scoring, not a direct model output. Stage 2 contributes 8 features in total: bi-encoder score and rank, four BGE-M3 signal scores, and cross-encoder score and rank (Table 1, #10–17).

3.3 Stage 3: Graph-Based Reranking

While citation graphs are the natural structure for case law retrieval [15], the citation information is suppressed in our setting, and explicit citation links therefore cannot be used. We instead construct proxy citation structures through shared legal entities, following the GraphRAG paradigm [16]. Stage 3 consists of two independent sub-modules—GraphRAG Lite and a GNN reranker—that operate on *different graphs* and produce complementary features.

3.3.1 GraphRAG Lite.

Entity extraction. We extract four entity types from each document via regex: statute citations (e.g., "R.S.C. 1985, c. I-2, s. 3(1)"), judge names (e.g., "Gagné J."), legal domain (8 categories by keyword matching: immigration, tax, IP, aboriginal, criminal, administrative, labour, environmental), and case outcome (dismissed/allowed).

Bipartite graph. Let $D = \{d_1, \ldots, d_n\}$ (n=9,556) denote documents and $S = \{s_1, \ldots, s_m\}$ the extracted entities. We construct a weighted bipartite graph $G_{\text{bip}} = (D \cup S, E)$ where an edge $(d, s) \in E$ exists if document d contains entity s. Each edge is weighted by entity type:

$$w(d, s) = w_{\tau(s)} \cdot \mathbf{1}[(d, s) \in E]$$

with type weights $w_{\text{statute}} = 0.50$, $w_{\text{judge}} = 0.15$, $w_{\text{domain}} = 0.10$, and $w_{\text{outcome}} = 0.05$.

Document graph projection. We project G_{bip} onto a document–document graph $G_{\text{doc}} = (D, E_{\text{doc}})$ via weighted one-mode projection:

$$A_{\text{doc}}(d_i, d_j) = \sum_{s \in S} w(d_i, s) \cdot w(d_j, s)$$

Two documents sharing more (or higher-weighted) entities receive a stronger edge.

Community detection. Leiden community detection [17] is applied to G_{doc} at three resolutions ($\gamma = 0.5, 1.0, 2.0$), each producing a complete partition of all 9,556 documents into non-overlapping communities. Lower resolutions yield larger communities (broad legal domains); higher resolutions yield smaller clusters (fine-grained case groups). We use all three partitions.

Computed Features. For each query–candidate pair (q, c), GraphRAG Lite produces 9 features: three same-community indicators (one per resolution, with value 1 if q and c belong to the same community and 0 otherwise), the community Jaccard (the fraction of the three resolutions where q and c co-occur), the shared statute count and shared judge count, the same-domain and same-outcome indicators, and an entity overlap score computed as a weighted Jaccard over all shared entities (Table 1, #18–26).

3.3.2 GNN Reranker.

The GNN reranker operates on a corpus-level graph G_{knn} that is distinct from the entity-projected graph G_{doc} used for community detection.

Graph construction. G_{knn} has 9,556 document nodes and 61,645 edges constructed by combining two edge sources: (1) k-nearest-neighbor edges (k=8) based on cosine similarity of bi-encoder embeddings, capturing semantic relatedness; and (2) entity overlap edges from G_{doc}, weighted at 30% of total edge weight.

Node features. Each node is represented by a 35-dimensional feature vector: 32 dimensions from PCA-compressed bi-encoder embeddings concatenated with 3 retrieval scores (BM25 RRF, bi-encoder similarity, cross-encoder probability).

Architecture. A 2-layer Graph Attention Network [19] (d_{hidden}=64, 4 attention heads, dropout 0.1) processes G_{knn}. We choose GAT over GCN because its attention mechanism learns to weight neighbor contributions differently, distinguishing highly relevant corpus neighbors from coincidentally similar documents [11]. The GNN outputs a relevance score per document node, trained with binary cross-entropy loss on gold citation pairs using Adam for 50 epochs

(lr=1e-3, full-batch transductive). To avoid information leakage, we use 5-fold out-of-fold predictions.

Computed Features. The GNN produces 2 features per query–candidate pair: the GNN relevance score and the within-query GNN rank. In total, Stage 3 contributes 11 features to the meta-learner (Table 1, #18–28).

3.4 Stage 4: Meta-Learner

We adopt a learning-to-rank meta-learner rather than a fixed score combination because the relative importance of retrieval signals varies across legal domains—e.g., statute overlap matters more for tax cases, while judge co-citation matters more for immigration cases [4]. A gradient-boosted tree captures these non-linear feature interactions without manual weight tuning. A LightGBM [8] binary classifier trained with binary cross-entropy loss—chosen for its ability to handle heterogeneous feature types (continuous scores, binary indicators, counts) and its robustness to the class imbalance inherent in retrieval [8] (typically 4 positives among 200 candidates per query)—combines 34 features from all preceding stages (Table 1).

Table 1 lists all 34 features grouped by pipeline stage.

Training uses GroupKFold (5 folds, grouped by query), with 3 random seeds for stability (15 models total). Key training strategies include: (1) gold positive injection—adding the 3,484 gold positives that lie outside the BM25 top-200 to the candidate pool; (2) stratified negative sampling (50% hard / 30% medium / 20% easy, 10:1 ratio, cap 50 per query); and (3) threshold re-optimization on the full candidate pool. Letting $\phi(q, c) \in \mathbb{R}^{34}$ denote the feature vector for the query–candidate pair (q, c) and f the trained LightGBM probability, the predicted citation set for query q is

$$\hat{C}_q = \{\, c \in C_q : f(\phi(q, c)) \geq t \,\},$$

where C_q is the BM25-RRF top-200 candidate pool from Stage 1 and the threshold t is optimised to maximise micro-F1 on training data.

Scale and regularisation. Although the dataset contains 2,001 training queries, the meta-learner is trained at the granularity of query–candidate pairs: 2,001 queries with up to 200 candidates per query yield on the order of $4{\times}10^5$ pair-level examples after the stratified negative-sampling cap. The ratio of training examples to features (roughly $1.2{\times}10^4$ examples per feature) places the meta-learner in the regime where gradient-boosted trees are robust to feature dimensionality. We further mitigate overfitting through query-grouped cross-validation (GroupKFold by query ID prevents within-query leakage across folds), early stopping at 80 rounds on a per-fold validation set, feature and row subsampling (column and row sampling rates both set to 0.8, minimum samples per leaf set to 20), and averaging across three random seeds.

Hyperparameter choices. Several hyperparameters were fixed without ablation due to compute budget. Their values are motivated as follows. The citation-context window of ±150 words follows Tran et al. [18] and corresponds to a typical paragraph length in Canadian federal case judgments. The entity type weights $w_{\text{statute}} = 0.50$, $w_{\text{judge}} = 0.15$, $w_{\text{domain}} = 0.10$, $w_{\text{outcome}} = 0.05$ reflect an information-content ordering: statute citations are the most discriminative legal entities (typically unique to a doctrinal area), whereas case outcomes (dismissed/allowed) are coarse binary attributes that appear in roughly half of all judgments and are the least discriminative. The Leiden resolutions $\gamma \in \{0.5, 1.0, 2.0\}$ form a $\sqrt{2}$-spaced sweep chosen to capture broad legal domains (low γ) and fine-grained case clusters (high γ) without manual tuning; using all three simultaneously is robust to misspecification of any single resolution. The 7 hard negatives per positive in bi-encoder training follow the default of the BGE recipe [20]. A systematic sensitivity sweep of these hyperparameters is left for future work.

4 RESULTS

4.1 Internal Validation

Table 2 compares our pipeline against simplified reproductions of prior COLIEE winning systems on the same candidate pool and evaluation protocol, isolating the contribution of each reranking strategy. We do not retrain the original systems end-to-end; instead, we reimplement the published feature set and reranking pipeline of each baseline as follows. **BM25 (vanilla)** is Okapi BM25 over full documents only, without RRF, without neural reranking, returning the top-200 candidates per query and predicting by a single global score threshold tuned on the validation set. **TQM 2024 (LTR)** combines BM25 with a LightGBM ranker over the 14-feature subset reported in the TQM system paper [10] (BM25, TF-IDF cosine, document and query lengths, simple lexical overlaps, and a single SBERT cosine), without SAILER, graph features, or our citation-context-aware cross-encoder. **JNLP 2025 (BM25+SAILER)** uses BM25 as the first stage and SAILER [9] as a reranker (using the publicly released checkpoint without retraining), followed by a LightGBM aggregator over the 8-feature set described by the JNLP authors [14]. The three baselines use the same training/validation split, the same candidate pool size (top-200 from BM25), and the same LightGBM hyperparameters where applicable, so differences among them in Table 2 reflect the published feature differences rather than tuning. The row for our system in the same table is included for ease of reference and follows a different protocol described below.

We use two complementary evaluation protocols. For the baseline comparisons reported in Table 2, we hold out the most recent 20% of training queries (401 of 2,001) by case ID as a chronological validation set, training each system on the remaining 1,600 queries. For the ablation and feature analyses reported in Tables 4 and following, we use 5-fold GroupKFold cross-validation on all 2,001 training queries, grouped by query ID to prevent within-query leakage, and report out-of-fold (OOF) micro-F1. The F1=0.311 reported for our system in Table 2 is the 5-fold OOF result on the full 2,001-query training set, reproduced in the table for direct comparison; a chronological hold-out evaluation that retrains the meta-learner on the earliest 80% of queries and evaluates on the most recent 20% would more directly quantify temporal generalisation, and we discuss this limitation in Section 4.5.

We submitted three runs at different thresholds: *balanced* (t=0.547, the CV-optimal threshold), *recall* (t=0.400, favoring coverage), and *precision* (t=0.700, favoring accuracy).

4.2 Official Test Results

Table 3 shows our three submissions alongside the top performers among the 22 participating teams, which together produced 54 official runs. Our best run (run 2, recall-favouring) reached F1=0.177 and ranked 35th of the 54 per-run submissions (15th among the 22 teams when each team is represented by its best run), substantially

Table 1: Feature inventory of the LightGBM meta-learner (34 features per query–candidate pair). Type: cont. = continuous, int = integer count or rank, bin. = binary.

Stage	#	Feature	Type	Description
Stage 1	1	bm25_score	cont.	Raw BM25 score (full-document query)
	2	bm25_rrf_score	cont.	RRF-fused score from multi-view BM25
	3	n_context_matches	int	Number of citation-context windows matching c
	4	max_context_bm25	cont.	Highest BM25 score among context windows for c
	5	tfidf_cosine	cont.	Cosine similarity of TF-IDF vectors (50K vocab)
	6	jaccard	cont.	Word-level Jaccard: $\lvert T_q \cap T_c \rvert / \lvert T_q \cup T_c \rvert$
	7	shared_bigrams	cont.	Bigram-level Jaccard overlap
	8	length_ratio	cont.	$\min(\lvert q \rvert, \lvert c \rvert)/\max(\lvert q \rvert, \lvert c \rvert)$ in words
	9	shared_legal_terms	int	Shared words from 42-term legal vocabulary
Stage 2	10	biencoder_score	cont.	Cosine similarity of BGE-large embeddings
	11	biencoder_rank	int	Within-query rank by bi-encoder score
	12	m3_dense_score	cont.	BGE-M3 dense embedding similarity
	13	m3_sparse_score	cont.	BGE-M3 sparse lexical matching score
	14	m3_colbert_score	cont.	BGE-M3 ColBERT late-interaction score
	15	m3_fused_score	cont.	Weighted fusion of M3 dense/sparse/ColBERT
	16	crossencoder_score	cont.	DeBERTa softmax $P(\text{cited} \mid q, c)$
	17	crossencoder_rank	int	Within-query rank by cross-encoder score
Stage 3	18	same_community_0.5	bin.	Co-membership at Leiden resolution 0.5
	19	same_community_1.0	bin.	Co-membership at Leiden resolution 1.0
	20	same_community_2.0	bin.	Co-membership at Leiden resolution 2.0
	21	community_jaccard	cont.	Fraction of resolutions with co-membership
	22	shared_statutes	int	Count of shared statute citations
	23	shared_judges	int	Count of shared judge names
	24	same_domain	bin.	Same legal domain (8 categories)
	25	same_outcome	bin.	Same case outcome (dismissed/allowed)
	26	entity_overlap_score	cont.	Weighted Jaccard over all shared entities
	27	gnn_score	cont.	GAT output relevance score
	28	gnn_rank	int	Within-query rank by GNN score
Stage 4	29	bm25_rrf_rank_norm	cont.	Normalized BM25 rank within query (0=best)
	30	bm25_rrf_score_gap	cont.	BM25 score minus query mean
	31	biencoder_score_gap	cont.	Bi-encoder score minus query mean
	32	crossencoder_score_gap	cont.	Cross-encoder score minus query mean
	33	top_score_ratio	cont.	BM25 score / query maximum score
	34	score_above_median	bin.	BM25 score > query median

Table 2: Internal validation results against simplified prior winners.

System	F1	Prec.	Recall
BM25 (vanilla)	0.033	0.018	0.217
TQM 2024 (LTR)	0.119	0.145	0.101
JNLP 2025 (BM25+SAILER)	0.138	0.123	0.157
Ours (t=0.547, balanced)	0.311	0.368	0.270
Ours (t=0.400, recall)	0.285	0.264	0.310
Ours (t=0.700, precision)	0.273	0.472	0.192

below our cross-validated estimate of F1=0.311. The winner (NOWJ) achieved F1=0.422, and JNLP, the COLIEE 2025 winner at F1=0.335, improved to 0.413.

Table 3: Official COLIEE 2026 Task 1 results (selected runs).

Rank	Team / Run	F1	Prec.	Recall
1	NOWJ / sub_2	0.422	0.424	0.421
3	JNLP / random_forest	0.413	0.434	0.393
7	SIL / sil2	0.387	0.419	0.360
35	ABAI / run2 (recall)	0.177	0.160	0.199
36	ABAI / run1 (balanced)	0.167	0.195	0.146
41	ABAI / run3 (precision)	0.131	0.220	0.093

4.3 Per-Stage Retrieval Quality

A natural diagnostic for a cascaded retrieval pipeline is how recall and ranking quality evolve across stages. By construction, every reranker and feature extractor in Stages 2–3 operates on the fixed pool of 200 candidates produced by Stage 1, so the downstream stages can only change the ordering of candidates within

the pool, not the recall of the pool itself. Recall@200 is therefore determined entirely by the BM25-RRF first stage, which retrieves 57.8% of gold positives on the training data; this value is an upper bound on the achievable recall of every subsequent stage in our system. The reranking contribution of each stage is captured by the cross-validated F1 deltas reported in the ablation (Table 4), which measure the impact of each component on the final binary decision. Test-set recall@k and NDCG@k per stage would have been the most informative diagnostic for first-stage failures; the gold labels for the official test set were not released to participants in time for this analysis, and we report only the official test F1 in Section 4.2.

4.4 Ablation Study

Table 4 shows the contribution of each pipeline component, measured by cross-validation with out-of-fold predictions on RRF-only candidate pools.

Table 4: Component ablation (5-fold cross-validation). Feature # refers to Table 1.

Configuration	Features	F1	Recall	ΔF1
Full pipeline	all 34	0.311	0.270	—
− Cross-encoder	#16, 17	0.236	0.248	−0.075
− Lexical features	#5–9	0.289	0.276	−0.023
− GNN + BGE-M3	#12–15, 27, 28	0.299	0.286	−0.012
− GraphRAG	#18–26	0.305	0.295	−0.006
− Bi-encoder	#10, 11	0.310	0.281	−0.001

The cross-encoder is the dominant component (−24% relative F1). Lexical features rank second (−7.3%), followed by GNN + BGE-M3 (−3.8%) and GraphRAG (−2.0%). The bi-encoder contributes minimally (−0.3%), as its signal is largely subsumed by the cross-encoder and BM25.

The GraphRAG Lite and GNN reranker components together contribute $\Delta F1 \approx -0.012$ on cross-validation. Their training-time cost is modest: entity extraction is regex-only, Leiden community detection runs once on the 9,556-document graph in roughly 107 seconds on a single CPU core, and the GNN trains in approximately 25 minutes on a single NVIDIA DGX Spark GB10 GPU (one-off; the model is cached for inference). Inference cost on the same hardware is negligible: community lookups are $O(1)$ per pair and a single forward pass of the 9,556-node GAT completes in about 3 seconds for the 400 test queries. The graph components are therefore best characterised as inexpensive structural retrofits to the meta-learner feature set, rather than as core ranking components.

To quantify the discriminative signal carried by graph features, we compare their distribution across gold and non-gold candidate pairs within the BM25 top-200 candidate pool on the training data (Table 5). Graph features show consistent enrichment in gold pairs across all four categories, with ratios of 1.43–1.72× over non-gold pairs. This signal is real but modest, and it is correlated with rather than redundant to lexical similarity, consistent with the additive contribution of graph features in the ablation.

The niche in which graph features help most is the deeper portion of the BM25-RRF candidate pool: 470 gold pairs that BM25-RRF ranks below position 50 still exhibit a strongly positive aggregate graph signal (same community at all three resolutions, multiple shared statutes, and shared judges). A representative example is the citation from query 057595 (*Olvera v. Canada (M.C.I.)*, 2012, a Federal Court judicial review of a Refugee Protection Division decision concerning a Mexican family) to gold case 033043 (a 2010 Federal Court refugee-protection decision in the same area decided by the same presiding judge). BM25-RRF ranks 033043 only 63rd for this query, yet GraphRAG features capture the latent affinity through three shared Leiden communities, three shared statute citations, two shared judge names, and the same legal domain. Such pairs are precisely the cases that pure semantic reranking is least able to recover, and graph features provide an inexpensive complementary signal for the meta-learner.

Table 5: GraphRAG feature distribution in gold vs. non-gold pairs (BM25 top-200, training data). Feature # refers to Table 1.

#	Feature	Gold	Non-gold	Ratio
20	same_community ($\gamma = 2.0$)	26.8%	17.3%	1.55×
23	shared_judges (mean count)	0.22	0.13	1.72×
22	shared_statutes (mean count)	0.90	0.59	1.53×
26	entity_overlap (mean score)	0.171	0.120	1.43×

4.5 Analysis: CV-to-Test Gap

The 43% relative drop from CV F1=0.311 to test F1=0.177 is the primary finding of our work. We identify three contributing factors and report the empirical basis for each.

Recall ceiling. On the training queries, only 57.8% (4,767 of 8,251) of gold positives appear in the BM25 top-200 candidate pool, so the achievable post-reranking recall is bounded above by this fraction even for a perfect downstream reranker. This ceiling is a property of the first-stage retrieval design rather than of any particular evaluation split: our cross-validated post-threshold recall of 0.270 already sits well below the ceiling, and the test recall of 0.199 (run 2, the recall-favouring run) leaves further headroom against the same upper bound. The recall ceiling therefore acts as a baseline constraint shared by both cross-validation and test, and the drop from 0.270 to 0.199 in observed recall is attributable not to a collapse of the ceiling but to the two factors discussed below.

Temporal distribution shift. Citation patterns in Canadian federal case law evolve as new statutes take effect and doctrinal interpretations are refined. The 2026 test queries are drawn from the most recent cases in the corpus, while our training queries span a broader and earlier time range. We partition the 2,001 training queries into four equal quartiles by case ID (a proxy for chronological order) and find that the BM25-RRF top-200 recall is reasonably stable across the partition: 53.7%, 57.6%, 59.7%, and 60.2% from oldest to newest quartile, with average gold citations per query in a narrow band of 3.89–4.28. First-stage retrieval quality therefore does not appear to degrade systematically with case age within the training window, suggesting the temporal component of the CV-to-test gap is driven primarily by changes in the discriminative geometry of the reranking and threshold stages rather than

by a collapsing recall ceiling. Our 5-fold cross-validation mixes earlier and later cases in each fold, which can still overestimate generalisation to a strictly temporally out-of-distribution test split; a chronological hold-out evaluation that retrains the meta-learner on the earliest 80% of queries and evaluates on the most recent 20% would quantify this effect directly, and is left as future work.

Threshold miscalibration under gold injection. During training, we injected the 3,484 gold positives that lie outside the BM25 top-200 into the candidate pool (Section 3.4) and optimised the decision threshold on this gold-injected pool. At inference time, no such injection occurs and the LightGBM score distribution shifts downward: on the test queries the meta-learner output scores have a median of 0.021, a 95th percentile of 0.231, and a 99th percentile of 0.635. The threshold $t = 0.547$ that maximised training F1 therefore selects only the most confident predictions at inference: run 1 produced 1,316 predictions in total (mean 3.3 per query), against a training gold density of 4.1 per query. This shift accounts for the lower precision and recall of the balanced run on the test set relative to its cross-validated estimate, and is consistent with the recall-favouring run 2 at $t = 0.400$ being our best official submission. Ma et al. [12] formalise distribution-aware list truncation for ranking, and adopting such calibration is a natural next step.

5 CONCLUSION

We proposed a four-stage hybrid pipeline for legal case retrieval that combines lexical retrieval, neural reranking, graph-based community features, and gradient-boosted meta-learning. The ablation study confirms the cross-encoder with selective truncation as the most impactful component (−24% relative F1 when removed), while graph-based features from GraphRAG Lite and the GNN reranker provide modest but consistent gains (−2.0% and −3.8%, respectively).

Our system achieved F1=0.177 on the COLIEE 2026 test set, with our best run (run 2, recall-favouring) ranking 35th of 54 official runs and 15th of the 22 participating teams (by each team's best run)—substantially below our cross-validated estimate of 0.311. This gap reveals three systematic issues that generalize beyond our system: (1) a recall ceiling of 57.8% imposed by BM25 top-200 retrieval, bounding the performance of all downstream stages; (2) temporal distribution shift, where the newest test queries exhibit citation patterns not captured by random cross-validation folds; and (3) threshold miscalibration from training on gold-injected candidate pools whose score distribution differs from the test-time pool.

Future work should address the recall ceiling through hybrid dense-lexical first-stage retrieval, replace random cross-validation with temporal splits that mirror test conditions, develop threshold calibration methods that are robust to the absence of gold positive injection at inference time, and conduct a qualitative case-level analysis of when graph-based features are decisive.

ACKNOWLEDGMENTS

This work was supported by the IITP(Institute of Information & Communications Technology Planning & Evaluation)-ITRC(Information Technology Research Center) grant funded by the Korea government(Ministry of Science and ICT)(IITP-2026-RS-2024-00436936). This research was supported by the MSIT(Ministry of Science and ICT), Korea, under the ICAN(ICT Challenge and Advanced Network of HRD) support program(IITP-2026-RS-2023-00259497) supervised by the IITP(Institute for Information & Communications Technology Planning & Evaluation). This research was supported by the MSIT(Ministry of Science and ICT), Korea, under the Graduate School of Virtual Convergence support program(IITP-2026-RS-2023-00254129) supervised by the IITP(Institute for Information & Communications Technology Planning & Evaluation).

REFERENCES

[1] Jianlyu Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2024. M3-Embedding: Multi-Linguality, Multi-Functionality, Multi-Granularity Text Embeddings Through Self-Knowledge Distillation. In *Findings of the Association for Computational Linguistics: ACL 2024*. 2318–2335.

[2] Gordon V Cormack, Charles LA Clarke, and Stefan Buettcher. 2009. Reciprocal rank fusion outperforms condorcet and individual rank learning methods. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*. 758–759.

[3] Zhuyun Dai and Jamie Callan. 2020. Context-Aware Term Weighting For First Stage Passage Retrieval. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1533–1536.

[4] Yi Feng, Chuanyi Li, and Vincent Ng. 2024. Legal case retrieval: A survey of the state of the art. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 6472–6485.

[5] Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. Deberta: Decoding-enhanced bert with disentangled attention. *arXiv preprint arXiv:2006.03654* (2020).

[6] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-Rank Adaptation of Large Language Models. In *International Conference on Learning Representations*.

[7] Yoshinobu Kano, Randy Goebel, Mi-Young Kim, Calum Kwan, Ken Satoh, Hiroaki Yamada, and Masaharu Yoshioka. 2026. An Overview of the COLIEE 2026 Competition: Legal Case Law and Statute Law Information Retrieval and Entailment. In *Proceedings of the 21st International Conference on Artificial Intelligence and Law (ICAIL 2026)*.

[8] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. 2017. Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems* 30 (2017).

[9] Haitao Li, Qingyao Ai, Jia Chen, Qian Dong, Yueyue Wu, Yiqun Liu, Chong Chen, and Qi Tian. 2023. Sailer: structure-aware pre-trained language model for legal case retrieval. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1035–1044.

[10] Haitao Li, You Chen, Zhekai Ge, Qingyao Ai, Yiqun Liu, Quan Zhou, and Shuai Huo. 2024. Towards an in-depth comprehension of case relevance for better legal retrieval. In *JSAI International Symposium on Artificial Intelligence*. Springer, 212–227.

[11] Antoine Louis, Gijs Van Dijck, and Gerasimos Spanakis. 2023. Finding the law: Enhancing statutory article retrieval via graph neural networks. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*. 2761–2776.

[12] Yixiao Ma, Qingyao Ai, Yueyue Wu, Yunqiu Shao, Yiqun Liu, Min Zhang, and Shaoping Ma. 2022. Incorporating retrieval information into the truncation of ranking lists for better legal search. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 438–448.

[13] Yixiao Ma, Yueyue Wu, Qingyao Ai, Yiqun Liu, Yunqiu Shao, Min Zhang, and Shaoping Ma. 2023. Incorporating structural information into legal case retrieval. *ACM Transactions on Information Systems* 42, 2 (2023), 1–28.

[14] Hai Nguyen et al. 2025. JNLP at COLIEE 2025: Hybrid Large Language Model-based Framework for Legal Information Retrieval and Entailment. In *Proceedings of the Workshop on the Twelfth International Competition on Legal Information Extraction and Entailment COLIEE 2025 in association with the 20th International Conference on Artificial Intelligence and Law*, Randy Goebel et al. (Eds.).

[15] Shounak Paul, Pawan Goyal, and Saptarshi Ghosh. 2022. Lesicin: A heterogeneous graph-based approach for automatic legal statute identification from indian legal documents. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 36. 11139–11146.

[16] Boci Peng, Yun Zhu, Yongchao Liu, Xiaohe Bo, Haizhou Shi, Chuntao Hong, Yan Zhang, and Siliang Tang. 2025. Graph retrieval-augmented generation: A survey. *ACM Transactions on Information Systems* 44, 2 (2025), 1–52.

[17] Vincent A Traag, Ludo Waltman, and Nees Jan Van Eck. 2019. From Louvain to Leiden: guaranteeing well-connected communities. *Scientific reports* 9, 1 (2019), 5233.

[18] Vu Tran, Minh Le Nguyen, and Ken Satoh. 2019. Building legal case retrieval systems with lexical matching and summarization using a pre-trained phrase scoring model. In *Proceedings of the seventeenth international conference on artificial intelligence and law*. 275–282.

[19] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph Attention Networks. In *International Conference on Learning Representations*.

[20] Shitao Xiao, Zheng Liu, Peitian Zhang, Niklas Muennighoff, Defu Lian, and Jian-Yun Nie. 2024. C-pack: Packed resources for general chinese embeddings. In *Proceedings of the 47th international ACM SIGIR conference on research and development in information retrieval*. 641–649.

[21] Kun Zhang, Chong Chen, Yuanzhuo Wang, Qi Tian, and Long Bai. 2023. Cfgl-lcr: A counterfactual graph learning framework for legal case retrieval. In *Proceedings of the 29th ACM SIGKDD Conference on knowledge discovery and data mining*. 3332–3341.

GraphRAG for Statutory Retrieval and Entailment

Liam Kaan Cody
Graz University of Technology
Graz, Austria
lcody@student.tugraz.at

Ratko Savic
Graz University of Technology
Graz, Austria
ratko.savic@tugraz.at

Roman Kern
Know Center Research GmbH
Graz University of Technology
Graz, Austria
rkern@tugraz.at

Abstract

We present Team Cody's graph-based retrieval and entailment pipeline for Tasks 3 and 4 of the 2026 Competition on Legal Information Extraction and Entailment (COLIEE). We do not treat statutes as isolated articles. Instead, we model the Japanese Civil Code as a directed multigraph. Its edges capture explicit citations, implicit references, definition links, and links between articles later added under the same main article number. For Task 3, we first retrieve candidate articles using a hybrid dense-lexical ranking method. We then expand these candidates using the statute graph and rerank them with a fine-tuned cross-encoder. Next, a breadth-first traversal serializes the graph into a linear text representation. Finally, a large language model (LLM) uses this context to perform entailment via majority voting. Task 4 starts from the provided golden articles. The system optionally expands these articles to include their surrounding graph neighborhoods. This serialized context is then fed into the same LLM entailment module used in Task 3. Our best configurations achieved 85% accuracy on Task 3, placing fifth overall. On Task 4, the system reached 90% accuracy. Analysis suggests that graph expansion is more effective for contextualization than for retrieval, that pipeline recall outweighs precision, and that inference-time majority voting consistently improves predictions.

CCS Concepts

• **Information systems** → **Retrieval models and ranking**; **Question answering**; • **Applied computing** → *Law*; • **Computing methodologies** → *Natural language processing*.

Keywords

legal information retrieval, statutory entailment, GraphRAG, Japanese Civil Code, cross-encoder reranking, legal NLP

ACM Reference Format:
Liam Kaan Cody, Ratko Savic, and Roman Kern. 2026. GraphRAG for Statutory Retrieval and Entailment. In *Proceedings of COLIEE 2026 workshop, June 12, 2026, Singapore*. ACM, New York, NY, USA, 10 pages.

1 Introduction

Advances in Natural Language Processing (NLP) have had a substantial impact on the legal domain, influencing both professional practice and academic research. Recent surveys characterize Legal NLP as a broad research area spanning question answering, text summarization, text classification, named entity recognition, argument mining, and judgment prediction [1]. Progress in this area has come primarily from domain-adapted legal language models and shared evaluation benchmarks. However, advances across these tasks remain uneven. At the same time, researchers in legal information retrieval (IR) and NLP continue to emphasize the distinctive challenges posed by legal text. These include document length, specialized vocabulary, and the need for reliable and interpretable reasoning [1, 5, 6, 12].

Legal Question Answering (LQA) studies how legal sources can be reviewed, interpreted, and applied to specific facts. Its practical goal is to provide individuals and businesses with guidance on complex legal matters [1]. LQA primarily includes knowledge-driven (KD) inquiries into legal concepts and case-analysis (CA) problems grounded in real-world situations. The need for advanced reading comprehension and multi-step reasoning across both categories makes LQA a particularly challenging NLP task [42]. One subfield of LQA is question answering over statutes. This task can be framed as entailment: Given a set of facts and relevant statutory provisions, the system must decide whether a specific legal conclusion follows. Under this view, statutory reasoning becomes a natural language inference problem [37].

Within the broader Legal NLP landscape, LQA has emerged as a field for combining retrieval with reasoning over authoritative legal sources. A variety of approaches currently exist in this domain. Some methods utilize knowledge-graph-based question answering over legislation [35], while others focus on discourse-aware answer retrieval [34]. Additionally, retrieve-then-read pipelines are used to process retrieved statutory provisions, generating interpretable long-form answers [23]. As summarized by Ariai et al. [1], LQA is now one of the most active subareas of legal NLP, and current systems increasingly depend on identifying the right legal provisions before downstream answer generation or verification.

For LQA over statutes, a single article is often insufficient to determine the answer. The meaning and legal effect of a provision usually depend on definitions, exceptions, and hierarchical context. Often, there are also many cross-references located elsewhere in the code. Consequently, methods relying solely on article-level text matching often miss the context required for accurate entailment [18, 22, 24]. These challenges are especially pronounced in civil-law codes, whose organization is explicitly hierarchical and referential [22, 24].

A common benchmark for evaluating how well computational models handle these deeply connected legal structures is the Competition on Legal Information Extraction and Entailment (COLIEE) [17]. Each year, the competition challenges researchers to develop state-of-the-art NLP systems capable of retrieving relevant legal

COLIEE 2026, June 12, 2026, Singapore

documents and performing logical reasoning over complex statutes and case law. Prior work in COLIEE highlights the mutual relationship between statute retrieval and entailment, demonstrating that jointly modeling the two tasks improves both retrieval quality and the accuracy of the final yes/no decision [27]. Reflecting this, the statute branch of COLIEE 2026 features two complementary scenarios. Task 3 integrates retrieval and entailment: systems must return both relevant civil-law articles and a yes/no decision for unseen queries. Task 4 focuses strictly on controlled statute entailment when the golden articles are already provided [8, 17].

In this paper, we address both tasks using a shared graph-based pipeline. We represent the Japanese Civil Code as a graph of statutes and use it to construct a task-specific legal context. For Task 3, the system retrieves and reranks candidate articles from the full corpus before expanding them into a local subgraph. For Task 4, it starts from the provided golden articles and optionally expands their local graph neighborhoods. Inspired by Graph Retrieval-Augmented Generation (GRAG), we serialize the selected subgraph into a linear context for entailment with a large language model (LLM). We further reduce inference variance through majority voting across multiple model calls, following the broader intuition of self-consistency decoding [15, 38].

1.1 Contributions

This paper contributes an end-to-end study of how graph context can be used for COLIEE 2026 Tasks 3 and 4. At the center of the system is an article-level directed multigraph of the Japanese Civil Code whose typed edges capture explicit citations and implicit dependencies. In Task 3, this graph is inserted between hybrid ranking and cross-encoder reranking, broadening the candidate set. In Task 4, where the relevant articles are provided, the evidence is expanded before entailment. In both settings, the selected neighborhoods are serialized using a BFS-derived hierarchy so that the entailment model receives both the article text and the local structure. The novelty of the approach lies in adapting GraphRAG to statutory entailment over the Japanese Civil Code: an article-level graph is placed at different stages of the Task 3 and Task 4 pipelines, and its neighborhoods are serialized for LLM reasoning. The evaluation tests suggest that graph expansion is more useful for contextualization around known articles than for broadening at the retrieval stage.

2 Related Work

Our approach combines four converging lines of work: (a) representing legislation as a structured graph rather than an independent-text corpus [9, 22, 39], (b) high-recall hybrid retrieval with neural reranking [10, 28, 32, 33], (c) GraphRAG-style conversion of retrieved subgraphs into LLM-consumable linear contexts [11, 15, 31], and (d) robust LLM inference via ensembling and voting [29, 38, 40].

2.1 Statute-graph modeling

A recurring observation in statutory reasoning is that an article's meaning often depends on surrounding provisions, definitions, and cross-references. This motivates retrieval methods that exploit legislative structure rather than treating statutes as isolated documents. For instance, Louis et al. [22] represent legislation as a graph with nodes for section headings and articles. They then use a GNN so that each article's retrieval representation reflects not only its own text but also its position in the surrounding legislative hierarchy. Similarly, the legal knowledge management literature emphasizes that modeling both explicit citations and implicit relationships (e.g., shared concept hierarchies) improves retrieval coverage [9, 39]. Within the COLIEE benchmark specifically, Mizuno and Kano [24] construct a legal structure graph capturing hierarchical placement and referential dependencies. However, their approach recursively inlines cited provisions into the citing article's text, which can lead to redundant, exponentially expanding contexts. In contrast, our design keeps the graph at the article level. It represents the civil code as a directed multigraph with explicit citations and implicit relationships, and uses neighborhood expansion as a targeted, recall-oriented mechanism. This avoids the context bloat caused by recursive inlining.

2.2 Hybrid retrieval and neural reranking

Hybrid retrieval is a standard baseline in legal information retrieval. It combines lexical matching, such as BM25, with dense semantic retrieval, then merges the two rankings with Reciprocal Rank Fusion [10], helping cover both exact legal terms of art and paraphrased queries. Following high-recall retrieval, cross-encoder rerankers are widely used to improve precision [28]. Recent benchmark-specific systems, such as JLGR [24], have demonstrated the effectiveness of pairing graph-augmented bi-encoders with cross-encoder reranking to balance this precision–recall trade-off. Our pipeline extends the standard "retrieve-then-rerank" paradigm by inserting a novel graph expansion step *between* hybrid retrieval and reranking, followed by a cross-encoder that selectively filters and restores precision over the structurally expanded candidate set.

2.3 Graph-Augmented LLM Generation (GraphRAG)

GraphRAG methods generalize classical retrieval-augmented generation (RAG) by retrieving subgraphs and presenting both textual and topological evidence to an LLM [15, 31]. For scaling to large corpora, local-to-global approaches use staged summarization to consolidate evidence before answering [11].

However, standard GraphRAG often expects semantic entities and struggles with the rigid, hierarchical cross-references inherent to statutory law. To address this domain-specific challenge, we introduce a breadth-first search (BFS) serialization method and LLM-based context consolidation. This translates general graph-to-LLM principles into a statute-specific pipeline: we (i) retrieve a statute subgraph, (ii) serialize it into a hierarchical linear context suited for prompting without losing structural integrity, and (iii) consolidate redundant statements while strictly preserving legal exceptions and conditions.

2.4 Robust LLM inference via ensembling

Legal entailment in COLIEE Task 4 has increasingly been addressed via LLM prompting due to limited labeled data and the need for faithful reasoning. Prompt engineering, including structured reasoning templates (e.g., IRAC-style prompts), significantly impacts performance [2, 29, 40]. Beyond prompting, robustness can be improved

via self-evaluation signals [16] or by issuing multiple independent LLM calls and taking a majority vote [2]. Self-consistency samples diverse reasoning paths and selects the most consistent answer, effectively reducing stochastic decoding errors [38].

3 Dataset

We use the 2025 English translation of the Japanese Civil Code together with the COLIEE query dataset [17]. The COLIEE query dataset consists of annual legal bar examination problems from 2006 to 2024 and is distributed as one XML file per year. Each XML file contains a set of queries, each with a query text, a yes/no label, and a list of ground-truth articles. The query files were translated with Gemini 3 Pro [13, 17].

4 Methods

This section presents the methods used for COLIEE Tasks 3 and 4. In both settings, we represent the civil code as a graph and rely on LLM-based entailment to produce the final yes/no decision. The two tasks share the same overall architecture, differing primarily in the way the legal context for inference is constructed. For Task 3, the system first retrieves candidate articles from the corpus, whereas Task 4 begins from the provided golden articles and may further expand their local graph neighborhoods. The following subsection introduces this shared pipeline and then highlights the task-specific differences where relevant.

4.1 Pipeline

4.1.1 Statute Graph. For both tasks, we model the civil code as a directed multigraph $\mathcal{G}$, allowing multiple directed edges between the same ordered pair of articles to represent different relation types.

$$\mathcal{G} = (\mathcal{V}, \mathcal{E}, \tau),$$

where $\mathcal{V}$ is the set of article nodes, $\mathcal{E}$ is the set of typed edges between articles, and $\tau : \mathcal{E} \to \mathcal{R}$ maps each edge to a relation type from the set $\mathcal{R}$. Each article node $v \in \mathcal{V}$ is identified by its article ID and stores text in the following format:

```
Article {id}
{optional caption}
{text content}
```

Edges in $\mathcal{E}$ use four relation types:

Explicit citation. An article directly cites another article by naming its article number or relative position to itself.

Caption-based implicit dependency. An article refers to another article by mentioning its caption rather than by naming its article number directly.

Definition-based implicit dependency. An article uses a term whose definition is given in another article.

Branch-numbered insertion. Two articles are linked when they appear in the same branch-numbered sequence, such as Articles 424-1 and 424-2. In the Civil Code, such branch numbers mark provisions inserted later. They leave the surrounding article numbering untouched, so that existing cross-references do not need to be rewritten [14].

The first three relation types correspond to the relation categories discussed by Mizuno and Kano [24]: caption or hierarchical context, implicit neighboring dependencies, and explicit inter-article references. In their representation, structural units such as parts, chapters, sections, and captions are modeled as separate nodes linked to article nodes through the legal hierarchy. Our graph is structured differently: instead of introducing separate hierarchy nodes and recursively inlining cited provisions, we use an article-level graph over inter-article relations. As an additional heuristic relation, we add links between articles in the same branch-numbered sequence. We do so because their placement in the code often signals a close relationship to the neighboring articles. For explicit citations and definition-based dependencies, we also add reverse edges so that these relations can be traversed in both directions.

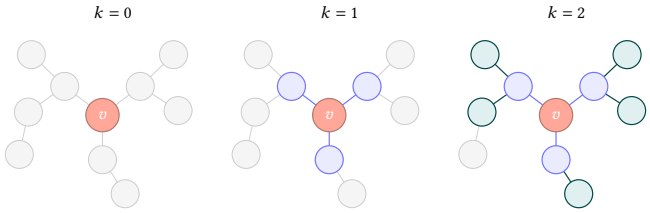


Figure 1: Cumulative k-hop ego-graph expansion around a seed article.

4.1.2 Retrieval.

Hybrid Article Ranking. We encode each article text with a pre-trained embedding model, which maps it to a vector in $\mathbb{R}^d$. For a query vector $\mathbf{q} \in \mathbb{R}^d$, we then retrieve candidate articles with a hybrid dense-lexical procedure. We first build two ranked shortlists: one based on cosine similarity and one based on BM25 over article text [33]. For an article node v, the fused score under Reciprocal Rank Fusion (RRF) [10] is

$$s(v) = \mathbf{1}_{\{v \in D\}} \frac{1}{k_{\text{RRF}} + \text{rank}_{\text{dense}}(v)} + \mathbf{1}_{\{v \in B\}} \frac{1}{k_{\text{RRF}} + \text{rank}_{\text{BM25}}(v)}.$$

Here, D and B denote the dense and BM25 shortlists. $\text{rank}_{\text{dense}}(v)$ and $\text{rank}_{\text{BM25}}(v)$ are the positions of article v in those lists, and k_{RRF} is the RRF constant. Sorting $D \cup B$ by the fused score gives the final hybrid shortlist V_{hyb}.

Ego-Graph Expansion. To enlarge the candidate set, we perform a k-hop ego-graph expansion around each article in V_{hyb}. This adds all connected articles within k hops, following the ego-graph retrieval idea used in GRAG [15]. We then merge the expanded sets into a single candidate subgraph defined as

$$\mathcal{G}_{\text{cand}} = \bigcup_{v \in V_{\text{hyb}}} \text{EgoGraph}(\mathcal{G}, v, k).$$

where $\text{EgoGraph}(\mathcal{G}, v, k)$ returns the subgraph of $\mathcal{G}$ containing all nodes reachable from v within k hops. As k increases, the ego graph around a seed article expands outward through successive layers of the local statutory neighborhood. A 0-hop ego graph contains only the seed article; a 1-hop ego graph additionally includes directly adjacent provisions; and a 2-hop ego graph further includes provisions reachable through a single intermediate relation. Figure 1 shows the process of cumulative expansion.

Node Reranking. The merged candidate graph $\mathcal{G}_{\text{cand}}$ may now still contain articles that are only loosely related to the query. Following the retrieve-then-refine pattern from GRAG [15], and as in the Task 3 system of Mizuno and Kano [24], we use a second-stage cross-encoder model [28], which was fine-tuned on the training queries to rerank the graph nodes. The model takes the query and article text as input and outputs a relevance score. After reranking, a minimum score threshold θ can be applied. If no article exceeds the threshold, we fall back to the top M articles. The retained articles form V_{sel}. The induced subgraph on these articles is

$$\mathcal{G}_{\text{sel}} = \mathcal{G}_{\text{cand}}[V_{\text{sel}}].$$

4.1.3 Entailment Decision. For the entailment stage, we first construct a breadth-first spanning forest over the retained subgraph $\mathcal{G}_{\text{sel}}$, rooted at the highest-ranked retained article [25, Section 12.3.1]. We then linearize this forest as hierarchical text following GRAG [15]. The traversal is therefore BFS-derived, although the rendered text is grouped hierarchically rather than printed in strict level order. This yields a legal context such as:

```
[ROOT] Article A.
Article A is connected to:
  1.1 Article B via explicit citation from Article A.
  Article B is connected to:
   1.1.1 Article D via explicit citation from Article B.
  1.2 Article C via explicit citation from Article A.

[Additional Related Articles]
[ROOT] Article E.
```

When the serialized context is large, we optionally consolidate it into a shorter single bulleted list that preserves distinct facts, conditions, and exceptions. This step is intended to reduce the risk that relevant information will be overlooked when it appears in the middle of a long input sequence, a position that LLMs handle less reliably [20]. When consolidation is applied, we use a fixed prompt template that instructs the model to rewrite the provided Japanese Civil Code text as a single bulleted list. The prompt directs the model to rely exclusively on the supplied text, remove redundant material, and retain all distinct facts and exceptions. We then run multiple independent LLM calls on each query-context pair and determine the final label by majority vote [2]. To simplify output parsing, the model is asked to return `true` or `false` rather than `yes` or `no`. Given the predictions $y_i \in \{\text{true}, \text{false}\}$ from the n independent LLM calls, the final majority-vote decision $\hat{y}$ is obtained as

$$\hat{y} = \arg \max_{y \in \{\text{true,false}\}} \sum_{i=1}^{n} \mathbf{1}_{\{y_i = y\}}.$$

Figure 2 outlines the shared pipeline. The task-specific sections below explain how Tasks 3 and 4 use this pipeline.

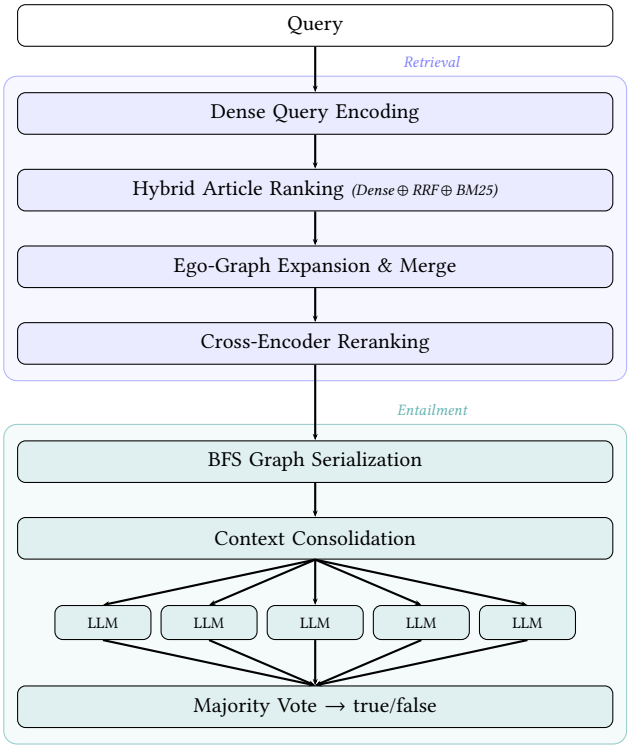


Figure 2: Shared pipeline for candidate construction and entailment over the statute graph.

4.2 Task 3: Statute Retrieval and Entailment for Yes/No Questions

4.2.1 System Overview. For Task 3, we use the full retrieval-to-entailment pipeline shown in Figure 2. The system first forms an initial candidate set from the full statute corpus, combining dense retrieval with BM25 so that both semantic and exact lexical matches enter the search space. It then expands these seeds across the statute graph, thereby recovering nearby provisions that provide definitions, exceptions, or cross-referenced conditions. The merged candidate set is reranked with a cross-encoder, serialized by a breadth-first traversal, and consolidated to remove redundancy while retaining legal dependencies. We then take a majority vote over the independent LLM calls to obtain the final yes/no prediction. This design treats Task 3 as a joint retrieval and reasoning problem, with article selection and legal inference treated as consecutive stages of the same workflow.

4.2.2 Cross-Encoder Training. We train and validate the reranker on the COLIEE query dataset [17]. Rather than using a random split, we adopt a chronological partition: queries from 2006–2022 are used for training, whereas queries from 2023–2024 are reserved for validation. This setup provides a more realistic estimate of performance on later examination questions and reduces the risk of overly optimistic results under temporal concept drift [4].

The cross-encoder is trained to rescore articles that have already passed the first-stage retrieval pipeline and to promote the most relevant ones in the final ranking. For each training query, candidate articles are generated by the same hybrid retrieval and ego-graph expansion procedure used at inference time. As a result, the candidate subgraphs reflect the types of candidates generated by the retrieval pipeline, rather than relying solely on randomly sampled corpus negatives. Within each candidate subgraph, articles whose IDs match a ground-truth article for the query are labeled as positives, whereas all other retrieved articles are labeled as negatives.

We rank negative candidates using two signals: the stage at which they enter the merged candidate subgraph and their similarity to the query. We then construct a mixed negative set by combining harder negatives from the top of this ranking with a larger pool of randomly sampled negatives. This strategy exposes the reranker to more challenging negative examples while avoiding overreliance on a small fixed set of hard negatives. The model is optimized with a listwise ranking loss, and checkpoints are selected based on validation Hit@1, that is, whether any ground-truth article is ranked first for the query.

4.2.3 Implementation Details. We implement the statute graph as a directed multigraph in NetworkX [26]. Dense retrieval is based on Qwen3-Embedding-4B [41] via the SentenceTransformers framework [32], while BM25 provides the lexical scores. The codebase also supports mean-pooling an article embedding with embeddings of its ego-graph neighbors, following the pooling idea in GRAG [15]. The submitted system, however, uses only the article's embedding. For hybrid ranking, we set the RRF constant to $k_{\text{RRF}} = 60$ and draw $\lceil \frac{N}{2} \rceil$ candidates from the dense shortlist and $\lfloor \frac{N}{2} \rfloor$ from the BM25 shortlist, allowing overlap. The merged candidates are scored with BGE-reranker-v2-m3 [7, 19], which we fine-tune on the training queries using a ListNet-style listwise cross-entropy loss [3]. During training, each query retrieves 50 seed articles, and each seed is expanded by 1 graph hop to form the candidate subgraph, which is scored by the reranker. Negatives are sampled with a 30/70 split between hard negatives and articles from the residual pool; the sample is refreshed every 2 epochs to avoid overfitting to a fixed set of distractors. We optimize the model with AdamW [21], using a learning rate of 2×10^{-5}, weight decay of 10^{-2}, and a warmup fraction of 0.1. Training ran for 12 epochs, and the best checkpoint reached a validation Hit@1 of 0.926.

Table 1 lists four run-specific parameters:

- N Number of articles returned by hybrid ranking.
- k Ego-graph expansion depth.
- M Number of reranked articles retained for entailment.
- θ Minimum reranker score when thresholding is enabled.

If thresholding is enabled and no article exceeds θ, we fall back to the top M articles. Both context consolidation and final entailment use Qwen3-32B [36]. We access it through OpenRouter's chat/completions endpoint under the model designation Qwen/Qwen3-32B [30]. We set the LLM temperature to 0.5 because, in internal evaluation, this value provided enough diversity for majority voting to be useful without introducing excessive variance into individual generations. For voting runs, we issue up to $n = 5$ independent calls and stop early once a majority is reached.

Submitted Runs. We submitted three experimental runs with different hyperparameter settings, as detailed in Table 1. These settings were chosen from the region of the hyperparameter space that yielded the strongest end-to-end accuracy in our internal evaluation. Because the pipeline relies on stochastic LLM inference, repeated runs show some variance. We therefore did not observe a single deterministic optimum and instead submitted representative configurations from that high-performing region.

Table 1: Hyperparameter settings for the three submitted runs for Task 3.

Run ID	N	K-hop depth k	M	θ
codygrag1	15	1	25	0.75
codygrag2	15	1	25	off
codyrag	50	0	25	off

4.3 Task 4: Legal Textual Entailment

4.3.1 System Overview. For Task 4, we retain the overall pipeline from Task 3 but simplify the candidate-construction stage since the golden article IDs are already provided as input. The system, therefore, does not need to search the full statute corpus or rerank a large candidate pool. Instead, it begins from the provided golden articles, optionally expands their local graph neighborhoods, and then proceeds directly to serialization and entailment. This local expansion is still useful as neighboring provisions may contain definitions, exceptions, or cross-referenced conditions that affect whether the provided statutory evidence entails the query. The resulting subgraph is serialized by the same breadth-first procedure used in Task 3 so that both article text and graph structure are presented to the LLM in a consistent linear form. Because this evidence set remains compact, no consolidation stage is required, and the serialized subgraph is passed directly to the majority-vote entailment module. This design treats Task 4 primarily as a reasoning problem over organizer-supplied legal evidence rather than a joint retrieval-and-reasoning problem over the full code. Figure 3 illustrates this variant.

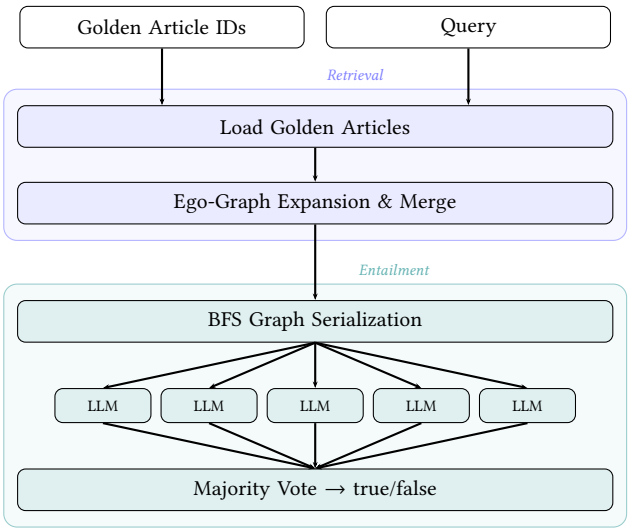


Figure 3: Task 4 variant of the pipeline.

4.3.2 Implementation Details. Task 4 uses the same NetworkX graph representation and the same Qwen3-32B LLM configuration via OpenRouter (`Qwen/Qwen3-32B`). As the golden article IDs are already provided, Task 4 does not use the dense retriever or cross-encoder reranker from Task 3. In all submitted runs, we serialize

Table 2: Hyperparameter settings for the three submitted runs for Task 4.

Run ID	Number of votes n	K-hop depth k
cody1	1	0
cody5	5	0
cody5graph	5	1

the seed neighborhood directly and pass it to entailment without consolidation. The submitted runs vary only in the number of votes n and the neighborhood depth k, allowing us to isolate the effects of majority voting and local graph expansion. Table 2 lists the corresponding hyperparameters.

5 Results

In this section, we present the official COLIEE 2026 results for Tasks 3 and 4, followed by an analysis of our submitted configurations.

5.1 Task 3

The official evaluation metric for Task 3 is accuracy. A query is counted as correct only if the system retrieves *every* relevant article (Recall = 1) *and* produces the correct yes/no entailment prediction. Accuracy is then the fraction of correct queries over all queries:

$$\text{Accuracy} = \frac{N_{\text{correct}}}{N_{\text{all}}}.$$

To break ties in cases of equal accuracy, the F_2 measure is used. This differs from COLIEE 2025 Task 3, which evaluated retrieval in isolation and used F_2 as the primary metric [24]. For that reason, cross-year comparisons are most meaningful when restricted to retrieval-stage behavior rather than overall end-to-end performance.

Table 3 reports results for all submitted runs. Our best configuration, `codyrag`, ranked fifth overall with an accuracy of 0.8537, trailing the top-ranked NOWJ_run1 (0.9512) by approximately ten percentage points.

Table 4 compares our three configurations across the R05 and R06 validation sets from our internal evaluation and the official R07 test set. `codyrag` achieved the highest accuracy on both R05 (0.7523) and R07 (0.8537), while `codygrag2` led on R06 (0.7260). Despite this single advantage, the graph-augmented variant trailed `codyrag` on the test set by 6.1 percentage points. These results suggest that the broader hybrid retrieval strategy generalized more reliably across datasets than the graph-augmented configurations, even though the latter remained competitive on individual validation splits. If we look only at retrieval-stage F_2, our strongest F_2 run (`codygrag1`, 0.7446) improves on the 0.6917 score reported by JLGR for their graph-based system [24].

To better understand the factors behind these differences in accuracy, Table 5 isolates the retrieval-stage metrics for our runs. The three configurations exhibit a clear trade-off between precision and recall. `codygrag1` achieved the highest precision (0.5764) and F_2 (0.7446) but the lowest recall (0.9024). Conversely, `codyrag` attained perfect recall along with the strongest R@5 and R@10 scores, at the expense of lower precision. `codygrag2` occupied a similar position to `codyrag`, with near-perfect recall (0.9939) but substantially reduced precision in comparison to `codygrag1`. Notably, the configuration with the highest end-to-end accuracy, `codyrag`, was also the one with the strongest recall, indicating that recall was the more important retrieval property for downstream entailment performance in this pipeline.

Task 3 accuracy favored the non-graph configuration, but that aggregate result does not show how graph expansion changed the candidate evidence available before reranking.

Table 3: 2026 COLIEE Task 3 results for all submitted runs.

Run	Accuracy	Accuracy (Sufficient)	F_2
NOWJ_run1	0.9512	0.9512	0.2224
JNLP1	0.9024	0.9024	0.7796
NOWJ_run2	0.8902	0.8902	0.2224
NOWJ_run3	0.8902	0.8902	0.2224
codyrag	**0.8537**	**0.8537**	**0.1848**
BengoR2E2	0.8293	0.8608	0.9242
HUKBmix	0.8171	0.8816	0.9446
BengoR2E1	0.8171	0.8481	0.9242
HUKBsoft	0.8171	0.8816	0.9076
HUKBbase	0.8049	0.8800	0.9304
BengoR1E1	0.8049	0.8462	0.9187
SCIlab1	0.8049	0.8684	0.0612
JNLP2	0.7927	0.7927	0.7796
DU2	0.7927	0.8333	0.3168
codygrag2	**0.7927**	**0.8025**	**0.2197**
JNLP3	0.7805	0.7805	0.7796
codygrag1	**0.7683**	**0.8630**	**0.7446**
DU	0.7683	0.8289	0.5383
DU1	0.7561	0.7848	0.2712
AIIRjpgen	0.5854	0.7619	0.4481
AIIRjpllm	0.5732	0.7344	0.4606
AIIRjpsw	0.5000	0.8913	0.5474

Table 4: Accuracy of our Task 3 configurations on the R05 and R06 validation sets and the official R07 test set.

Configuration	R05	R06	R07 (Test)
codygrag1	0.6881	0.6164	0.7683
codygrag2	0.6881	0.7260	0.7927
codyrag	0.7523	0.6849	0.8537

Table 5: Task 3 information-retrieval metrics for our submitted runs.

Run	F_2	Prec.	Recall	MAP	R@5	R@10
codygrag1	0.7446	0.5764	0.9024	0.8564	0.8902	0.8902
codygrag2	0.2197	0.0554	0.9939	0.8941	0.9736	0.9817
codyrag	0.1848	0.0439	1.0000	0.8899	0.9837	0.9878

5.2 Qualitative Analysis of Ego-Graph Expansion

For the comparison, we use the first stage of the graph-expanded runs in Table 1. This stage retrieves 15 seed articles through hybrid retrieval and expands each seed by one graph hop. For each query, we compare the expanded candidate pool against a no-expansion baseline ($k = 0$). To keep the comparison fair, the baseline is allowed to return the same number of articles as the expanded candidate set for that query. That way, if a ground-truth article appears in one run but not the other, the difference is due to expansion rather than list size.

The examples below examine this effect at the query level. In the first, a cross-reference recovers a missing ground-truth article. In the second, expansion adds topically related candidates, but still misses the target provision.

5.2.1 *Examples.*

Useful expansion: recovering an implicit pledge rule through a cross-reference. The query is:

> "A pledgee of movables may use the thing pledged to the extent necessary for its preservation even without obtaining the consent of the pledgor."

The ground-truth articles are Articles 298 and 350:

> **Article 298 (Custody of Thing Retained by Holders of Rights of Retention).**
> (1) The holder of a right of retention must possess the thing retained with the due care of a prudent manager.
> (2) The holder of the right of retention may not use, lease or provide as a security the thing retained unless that holder obtains the consent of the obligor; provided, however, that this does not apply to uses necessary for the preservation of that thing.
> (3) If the holder of a right of retention violates the provisions of the preceding two paragraphs, the obligor may demand that the right of retention be terminated.
>
> **Article 350 (Mutatis Mutandis Application of Provisions on Rights of Retention and Statutory Liens).**
> The provisions of Articles 296 through 300 and those of Article 304 apply mutatis mutandis to pledges.

The Japanese Civil Code makes heavy use of *mutatis mutandis* cross-references as a drafting technique to avoid restating shared rules across analogous institutions. As a result, the answer to this query is distributed over two provisions. Article 298 contains the substantive preservation exception, but it is written for the right of retention. Article 350 then applies these rules *mutatis mutandis* to pledges. From a linguistic perspective, the two articles cover different parts of the query: Article 298 contains the words "use," "consent," and "preservation," whereas Article 350 only establishes the connection to pledges, without restating the rule itself. This split also explains the lexical gap. Article 350 carries the pledge vocabulary but no preservation language, while Article 298 carries the preservation rule but speaks of the "holder of a right of retention" and the "obligor." Text-only retrieval can match one side or the other, but only the citation edge connects them; this is why one-hop expansion lifts recall from 0.5 to 1.0. The citation edge, therefore, provides the missing legal bridge that text-only retrieval cannot reliably infer. This suggests that future systems should treat explicit citation edges as legally meaningful signals. It may also be useful to classify citation types so that incorporation links can receive higher weights.

Ineffective expansion: crowding out an isolated real-rights provision. The query is:

> "A person who has acquired a superficies for owning bamboo or trees by prescription may assert the acquisition of the superficies against a person who subsequently purchased the land subject to the superficies and registered that effect, even without registration."

The sole ground-truth article is Article 177:

> **Article 177 (Requirements of Perfection of Changes in Real Rights on Immovables).**
> Acquisitions of, losses of and changes in real rights on immovables may not be duly asserted against any third parties, unless the same are registered pursuant to the applicable provisions of the Real Property Registration Act (Act No. 123 of 2004) and other laws regarding registration.

The graph-expanded run returns 36 candidates and still misses Article 177. The no-hop baseline places it at rank 23 within the same 36-candidate pool, yielding a recall of 1.0, compared to 0 for the expanded run. The reasons are both structural and linguistic. Article 177 is the general perfection rule for real rights on immovables and belongs to a foundational layer of the code. Its applicability to the query rests on legal reasoning rather than on an explicit cross-reference. Because Article 177 states a general rule, it is not necessarily connected by explicit citations from other articles. As a consequence, Article 177 has in-degree zero in our graph; if the retriever does not already surface it as a seed, one-hop expansion cannot recover it. The linguistic profile compounds this effect. The query is concrete and institution-specific ("superficies," "bamboo or trees," "acquired by prescription"), whereas Article 177 is formulated in abstract terms ("real rights on immovables," "third parties," "registration"). The seeds that do surface are about acquisitive prescription and superficies. They are relevant to the topic of the query, but not to its actual legal question, and their neighbors inherit the same bias, so expansion pushes Article 177 even further below the cutoff. This could motivate enriching the graph beyond direct citations. Adding semantic edges and legal-knowledge edges could also allow the expansion step to reach high-centrality articles that the code itself never explicitly cites.

Together, these examples show that ego-graph expansion is topology-dependent. It is valuable when relevant provisions are reachable from retrieved seeds through legally meaningful links, and less useful when the target article is isolated or when the added neighbors share the topic without answering the legal question.

5.3 Task 4

Task 4 uses the same accuracy metric as Task 3, but the golden articles are provided with each query, isolating the entailment decision from retrieval quality. Table 6 reports accuracy for all submitted runs across three evaluation sets (H30, R01, R02) and the official R07 test set.

Our best submission, `cody5`, achieved 0.9024 on R07, ranking it in the middle of the competition. This result is very close to the 0.9041

Table 6: Task 4 accuracy for all submitted runs across the H30, R01, R02, and test evaluation files.

Run	H30	R01	R02	R07 (Test)
DU1	0.9403	0.8818	0.9877	0.9634
DU2	0.9403	0.8818	0.9753	0.9634
JNLP-3	–	–	–	0.9512
IAIrun2	–	–	–	0.9512
IAIrun3	–	–	–	0.9512
DU3	0.9403	0.8727	0.9630	0.9390
IAIrun1	–	–	–	0.9390
JNLP-1	0.8358	0.7909	0.9259	0.9268
RUG-1	0.7612	0.7182	0.9136	0.9268
RUG-2	0.9851	0.9455	0.9383	0.9268
UA2	0.9104	0.7636	0.9506	0.9268
UA1	0.9104	0.7636	0.9383	0.9146
UA3	0.9104	0.7636	0.9383	0.9146
RUG-3	0.8507	0.8000	0.8765	0.9024
cody5	**0.8657**	**0.7727**	**0.9012**	**0.9024**
cody5graph	**0.8806**	**0.8000**	**0.9136**	**0.8902**
HUKBac	0.7463	0.7455	0.9136	0.8780
HUKBllama	0.7463	0.7455	0.9136	0.8780
NOWJ-1	0.8358	0.8273	0.9012	0.8780
cody1	**0.7761**	**0.7455**	**0.9012**	**0.8659**
NOWJ-3	0.9104	0.8273	0.9136	0.8049
AIIRCrossCoT	0.6119	0.4091	0.6667	0.7805
AIIRDivVote	0.6269	0.4182	0.6790	0.7805
HUKBllmjp	0.6866	0.7182	0.7284	0.7805
BengoREF	0.8955	0.7545	0.8889	0.7317
BengoREFSC5	0.9104	0.7455	0.8889	0.7317
BengoREFSC10	0.8955	0.7455	0.8889	0.7195
AIIRNeuSym	0.5970	0.4000	0.5802	0.6951
ASHIN-LLAMA3	1.0000	1.0000	1.0000	0.6098
ASHIN-LLAMA2	1.0000	1.0000	1.0000	0.5976
JNLP-2	0.8806	0.7727	0.9012	0.5732
Baseline	0.5224	0.5364	0.5309	0.5000
NOWJ-2	0.7761	0.7818	0.9012	0.5000
ASHIN-LLAMA4	1.0000	1.0000	1.0000	0.4024

accuracy reported by last year's winning KIS team [29]. The graph-augmented variant `cody5graph` scored slightly lower on the test set (0.8902), despite outperforming `cody5` on all three evaluation sets (H30, R01, R02). `cody1`, the single-call baseline, trailed both five-sample variants on three of the four evaluation sets and tied `cody5` on R02. The primary performance gap on R07 thus lies between the single-call and five-call settings (0.8659 vs. 0.9024) rather than between the two five-call variants (0.9024 vs. 0.8902).

5.4 Discussion

Our results on Tasks 3 and 4 point to four findings. First, the role of graph expansion depended on its position in the pipeline. In Task 3, the graph-augmented settings trailed the non-graph baseline in end-to-end accuracy. However, the qualitative analysis shows that graph links can still recover useful candidates in specific legal settings. In Task 4, expansion around known seed articles improved performance on most validation splits. Second, the graph's degree distribution is skewed, leading to inconsistent expansion across queries and making graph-augmented configurations sensitive to dataset composition. Third, in Task 3, recall mattered more than precision: the configuration with the best precision and F_2 produced the lowest end-to-end accuracy. Fourth, inference-time majority voting stabilized entailment predictions without any additional training.

5.4.1 Graph expansion appears more effective after than during retrieval. In Task 3, the non-graph hybrid retrieval configuration (`codyrag`, $N = 50$, $k = 0$) achieved the highest test-set accuracy (0.8537) and the strongest performance on R05 (0.7523), outperforming both graph-augmented settings (`codygrag1` and `codygrag2`, $N = 15$, $k = 1$). The per-dataset breakdown in Table 4 shows one exception: `codygrag2` led on R06 (0.7260 vs. 0.6849), so graph expansion can help on particular query distributions. Still, `codyrag` generalized more consistently across all three splits. The qualitative analysis in Section 5.2 clarifies this mixed result. Cross-reference links can recover provisions that text retrieval misses. The difficulty is that, in Task 3, expansion runs before the reranker has identified the focal provision, so the added neighbors may be legally related but fail to answer the query. This can dilute the evidence passed to entailment even when the graph contains useful legal structure.

Task 4 shows a different pattern. The golden articles are provided, so the model already has the key provisions, and expansion can only add context. Adding the 1-hop neighborhood (`cody5graph`, $k = 1$) improved accuracy over the no-expansion baseline (`cody5`, $k = 0$) on H30, R01, and R02, though not on R07. The neighboring articles here tend to supply definitions, exceptions, or cross-referenced conditions that clarify how those articles apply to the query. Taken together, these results suggest that the statute graph is more useful for contextualization once the relevant articles are known than for broadening the candidate set at the front of the retrieval pipeline.

5.4.2 Graph degree imbalance limits expansion reliability. The qualitative examples point to a structural reason for the uneven gains from graph expansion: the statute graph has an imbalanced degree distribution. Some articles are heavily cross-referenced, while others have few or no incoming edges. This matters because expansion makes the effective candidate set depend on the retrieved seeds. With $k = 0$, the first-stage cutoff is fixed by the number of retrieved seed articles N. With $k = 1$, the number and usefulness of additional candidates vary with each seed's local neighborhood. A seed connected by a direct statutory cross-reference can add a missing ground-truth article, as in the pledge example; a seed in a broad or only topically related neighborhood can instead add candidates that are legally adjacent but less decisive.

This mechanism is consistent with the dataset-dependent Task 3 results in Table 4. `codygrag2` outperformed `codyrag` on R06, but fell behind on R05 and R07. The point is therefore not that graph expansion is generally unhelpful, but that a single fixed N and k exposes the pipeline to variation in local graph topology. Our internal experiments showed the same sensitivity: as hop depth increased, subgraph sizes varied more widely across queries, making it difficult to choose a single setting that worked across all splits. Degree-aware expansion, for instance, capping neighborhood size or adapting k per query based on local graph density, could reduce this sensitivity and is worth exploring in future work.

5.4.3 Strict thresholding did not improve end-to-end accuracy. The Task 3 runs show that strong retrieval metrics do not necessarily translate into strong end-to-end entailment accuracy. Although the thresholded run `codygrag1` achieved the best precision (0.5764) and F_2 score (0.7446) among our submissions, it produced the lowest end-to-end accuracy (0.7683). The main reason is its lower recall. It retrieved fewer ground-truth articles than the other submitted runs, and in this pipeline, failing to recover a ground-truth article is sufficient to make the final prediction incorrect. By contrast, the higher-recall runs suggest that Qwen3-32B can tolerate some additional distractor articles after consolidation and still identify the relevant evidence. For this pipeline, preserving recall therefore appears to be the safer trade-off, even at the cost of a noisier context. More broadly, our results suggest that systems optimized primarily for high F_2 in earlier Task 3 settings can be outperformed in the new setting by pipelines that preserve recall more aggressively.

5.4.4 Voting improves stability. The Task 4 ablation is consistent with prior work showing that inference-time answer aggregation can improve robustness in legal entailment [2]. More broadly, it aligns with findings that self-consistency can improve the reliability of reasoning in LLMs [38]. Comparing `cody1` and `cody5`, which differ only in the number of LLM calls, reveals consistent gains from five-vote majority decisions across all evaluation sets. On the final test set (R07), accuracy increased from 0.8659 to 0.9024. These results indicate that part of the remaining error is attributable not only to missing knowledge, but also to variance across individual generations. In this setting, majority voting therefore improved performance without requiring additional training.

6 Conclusion and Future Work

Our experiments suggest that graph expansion is more useful as a contextualization step than as a broadening mechanism for retrieval. When the relevant articles were already known (Task 4), neighboring provisions supplied definitions and exceptions that improved accuracy on most validation splits. When expansion preceded the reranker (Task 3), it tended to introduce noise rather than signal, because the reranker had not yet identified the focal provision. The skewed degree distribution in the graph amplified this effect: well-connected provisions pulled in large neighborhoods, while narrowly scoped ones gained little from expansion. Separately, recall proved more important than precision for end-to-end accuracy; the LLM could handle distractor articles after consolidation, but omitting any ground-truth article was not recoverable. In a retrieval-only setting, F_2 remains a useful summary of the precision-recall trade-off; however, in the joint retrieval-and-entailment setting studied here, it was not the most accurate proxy for downstream entailment success. Future retrieval pipelines may want to optimize for objectives that weight recall more heavily than F_2. Majority voting improved predictions over single-call baselines across all evaluation splits without any additional training.

Most of the remaining errors can be traced to how the graph is constructed and traversed, making this the logical focus for future work. A natural starting point is degree-aware expansion: conditioning neighborhood depth on local graph density so that high-degree and low-degree articles receive comparably sized contexts. Rather than pulling in every node within a fixed k-hop radius, a query-conditioned extraction step could learn which neighbors to include for a given query. A lightweight graph neural network that scores candidate neighbors by estimated query relevance is one concrete way to achieve this. Our current edge types are also unweighted, yet not all relations carry the same information. An explicit cross-reference to a definition article is typically far more relevant than a link between articles in the same branch-numbered sequence. Learning per-type edge weights, or allowing the expansion policy to prefer certain relation types over others, would improve the signal-to-noise ratio of the expanded subgraph. Applying the pipeline to civil-law codes outside Japan would test whether these graph construction choices and the BFS serialization strategy hold up across legislative traditions with different drafting and cross-referencing conventions.

References

[1] Farid Ariai, Joel Mackenzie, and Gianluca Demartini. 2025. Natural Language Processing for the Legal Domain: A Survey of Tasks, Datasets, Models, and Challenges. *Comput. Surveys* 58, 6, Article 163 (2025), 37 pages. doi:10.1145/3777009

[2] Onur Bilgin, Logan Fields, Antonio Laverghetta Jr., Zaid Marji, Animesh Nighojkar, Stephen Steinle, and John Licato. 2024. Exploring Prompting Approaches in Legal Textual Entailment. *The Review of Socionetwork Strategies* (2024), 75–100. doi:10.1007/s12626-023-00154-y

[3] Zhe Cao, Tao Qin, Tie-Yan Liu, Ming-Feng Tsai, and Hang Li. 2007. *Learning to Rank: From Pairwise Approach to Listwise Approach.* Technical Report MSR-TR-2007-40. 9 pages. https://www.microsoft.com/en-us/research/publication/learning-to-rank-from-pairwise-approach-to-listwise-approach/

[4] Ilias Chalkidis, Manos Fergadiotis, and Ion Androutsopoulos. 2021. MultiEURLEX - A multi-lingual and multi-label legal document classification dataset for zero-shot cross-lingual transfer. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih (Eds.). Association for Computational Linguistics, Online and Punta Cana, Dominican Republic, 6974–6996. doi:10.18653/v1/2021.emnlp-main.559

[5] Ilias Chalkidis, Manos Fergadiotis, Prodromos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos. 2020. LEGAL-BERT: The Muppets straight out of Law School. In *Findings of the Association for Computational Linguistics: EMNLP 2020.* Association for Computational Linguistics, Online, 2898–2904. doi:10.18653/v1/2020.findings-emnlp.261

[6] Ilias Chalkidis, Abhik Jana, Dirk Hartung, Michael Bommarito, Ion Androutsopoulos, Daniel Katz, and Nikolaos Aletras. 2022. LexGLUE: A Benchmark Dataset for Legal Language Understanding in English. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers).* Association for Computational Linguistics, Dublin, Ireland, 4310–4330. doi:10.18653/v1/2022.acl-long.297

[7] Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2024. BGE M3-Embedding: Multi-Lingual, Multi-Functionality, Multi-Granularity Text Embeddings Through Self-Knowledge Distillation. arXiv:2402.03216 [cs.CL]

[8] COLIEE Organizers. 2026. Task 3: Statute Retrieval and Entailment for Yes/No Questions (SL-QA). https://coliee.org/COLIEE2026/tasks/task3 Accessed: 2026-03-26.

[9] Andrea Colombo, Anna Bernasconi, Luigi Bellomarini, Luigi Guiso, Claudio Michelacci, and Stefano Ceri. 2025. LegisSearch: navigating legislation with graphs and large language models. *Artificial Intelligence and Law* (2025). doi:10.1007/s10506-025-09482-6

[10] Gordon V. Cormack, Charles L. A. Clarke, and Stefan Büttcher. 2009. Reciprocal Rank Fusion Outperforms Condorcet and Individual Rank Learning Methods. In *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '09).* ACM, Boston, MA, USA, 758–759. doi:10.1145/1571941.1572114

[11] Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, Dasha Metropolitansky, Robert Osazuwa Ness, and Jonathan Larson. 2024. From Local to Global: A GraphRAG Approach to Query-Focused Summarization. arXiv:2404.16130. doi:10.48550/arXiv.2404.16130

[12] Debasis Ganguly, Jack G. Conrad, Kripabandhu Ghosh, Saptarshi Ghosh, Pawan Goyal, Paheli Bhattacharya, Shubham Kumar Nigam, and Shounak Paul. 2023. Legal IR and NLP: The History, Challenges, and State-of-the-Art. In *Advances in Information Retrieval (Lecture Notes in Computer Science, Vol. 13982).* Springer, 331–340. doi:10.1007/978-3-031-28241-6_34

[13] Google. 2026. Gemini 3 Pro. https://deepmind.google/technologies/gemini/ Large Language Model.

[14] House of Councillors Legislative Bureau. n.d.. Branch Numbers and Deletion of Articles. https://houseikyoku.sangiin.go.jp/column/column043.htm. In Japanese. Accessed: 2026-03-28.

[15] Yuntong Hu, Zhihan Lei, Zheng Zhang, Bo Pan, Chen Ling, and Liang Zhao. 2025. GRAG: Graph Retrieval-Augmented Generation. In *Findings of the Association for Computational Linguistics: NAACL 2025*. Association for Computational Linguistics, Albuquerque, New Mexico, 4145–4157. doi:10.18653/v1/2025.findings-naacl.232

[16] Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, Scott Johnston, Sheer El-Showk, Andy Jones, Nelson Elhage, Tristan Hume, Anna Chen, Yuntao Bai, Sam Bowman, Stanislav Fort, Deep Ganguli, Danny Hernandez, Josh Jacobson, Jackson Kernion, Shauna Kravec, Liane Lovitt, Kamal Ndousse, Catherine Olsson, Sam Ringer, Dario Amodei, Tom Brown, Jack Clark, Nicholas Joseph, Ben Mann, Sam McCandlish, Chris Olah, and Jared Kaplan. 2022. Language Models (Mostly) Know What They Know. doi:10.48550/arXiv.2207.05221

[17] Yoshinobu Kano, Randy Goebel, Mi-Young Kim, Calum Kwan, Ken Satoh, Hiroaki Yamada, and Masaharu Yoshioka. 2026. An Overview of the COLIEE 2026 Competition: Legal Case Law and Statute Law Information Retrieval and Entailment. In *Proceedings of the 21st International Conference on Artificial Intelligence and Law (ICAIL 2026)*.

[18] Mi-Young Kim, Juliano Rabelo, and Randy Goebel. 2019. Statute Law Information Retrieval and Entailment. In *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Law (ICAIL '19)*. ACM, Montreal, QC, Canada, 283–289. doi:10.1145/3322640.3326742

[19] Chaofan Li, Zheng Liu, Shitao Xiao, and Yingxia Shao. 2023. Making Large Language Models a Better Foundation for Dense Retrieval. arXiv:2312.15503 [cs.CL]

[20] Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024. Lost in the Middle: How Language Models Use Long Contexts. *Transactions of the Association for Computational Linguistics* 12 (2024), 157–173. doi:10.1162/tacl_a_00638

[21] Ilya Loshchilov and Frank Hutter. 2019. Decoupled Weight Decay Regularization. In *International Conference on Learning Representations (ICLR)*. https://openreview.net/forum?id=Bkg6RiCqY7

[22] Antoine Louis, Gijs van Dijck, and Gerasimos Spanakis. 2023. Finding the Law: Enhancing Statutory Article Retrieval via Graph Neural Networks. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, Dubrovnik, Croatia, 2761–2776. doi:10.18653/v1/2023.eacl-main.203

[23] Antoine Louis, Gijs van Dijck, and Gerasimos Spanakis. 2024. Interpretable Long-Form Legal Question Answering with Retrieval-Augmented Large Language Models. In *Proceedings of the 38th AAAI Conference on Artificial Intelligence*, Vol. 38. AAAI Press, 22266–22275. doi:10.1609/aaai.v38i20.30232

[24] Takao Mizuno and Yoshinobu Kano. 2025. Hierarchical and Referential Structure-Aware Retrieval for Statutory Articles using Graph Neural Networks. In *Proceedings of the Workshop on the Twelfth International Competition on Legal Information Extraction and Entailment (COLIEE 2025)*. ACM, Chicago, IL, USA, 33–36. https://coliee.org/documents/Proceedings/2025-Proceedings.pdf In association with the 20th International Conference on Artificial Intelligence and Law (ICAIL 2025).

[25] Pat Morin. 2013. *Open Data Structures: An Introduction*. Athabasca University Press. doi:10.15215/aupress/9781927356388.01

[26] NetworkX Developers. 2026. MultiDiGraph—Directed graphs with self loops and parallel edges. NetworkX documentation. https://networkx.org/documentation/stable/reference/classes/multidigraph.html Accessed: 2026-03-26.

[27] Hai-Long Nguyen, Thi-Hai-Yen Vuong, Ha-Thanh Nguyen, and Xuan-Hieu Phan. 2023. Joint Learning for Legal Text Retrieval and Textual Entailment: Leveraging the Relationship between Relevancy and Affirmation. In *Proceedings of the Natural Legal Language Processing Workshop 2023*. Association for Computational Linguistics, Singapore, 192–201. doi:10.18653/v1/2023.nllp-1.19

[28] Rodrigo Nogueira and Kyunghyun Cho. 2019. Passage Re-ranking with BERT. arXiv:1901.04085. doi:10.48550/arXiv.1901.04085

[29] Takaaki Onaga and Yoshinobu Kano. 2026. KIS: COLIEE 2025 Task 4 Solver Using Japanese LLM. *The Review of Socionetwork Strategies* (2026). doi:10.1007/s12626-026-00209-w

[30] OpenRouter. 2026. OpenRouter API Reference. https://openrouter.ai/docs/api-reference/overview Accessed: 2026-03-27.

[31] Boci Peng, Yun Zhu, Yongchao Liu, Xiaohe Bo, Haizhou Shi, Chuntao Hong, Yan Zhang, and Siliang Tang. 2025. Graph Retrieval-Augmented Generation: A Survey. *ACM Transactions on Information Systems* 44, 2, Article 35 (2025), 52 pages. doi:10.1145/3777378

[32] Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Association for Computational Linguistics, Hong Kong, China, 3982–3992. doi:10.18653/v1/D19-1410

[33] Stephen Robertson and Hugo Zaragoza. 2009. The Probabilistic Relevance Framework: BM25 and Beyond. *Foundations and Trends in Information Retrieval* 3, 4 (2009), 333–389. doi:10.1561/1500000019

[34] Francesco Sovrano, Monica Palmirani, Salvatore Sapienza, and Vittoria Pistone. 2025. DiscoLQA: zero-shot discourse-based legal question answering on European Legislation. *Artificial Intelligence and Law* 33, 2 (2025), 323–359. doi:10.1007/s10506-023-09387-2

[35] Francesco Sovrano, Monica Palmirani, and Fabio Vitali. 2020. Legal Knowledge Extraction for Knowledge Graph Based Question-Answering. In *Legal Knowledge and Information Systems: JURIX 2020: The Thirty-third Annual Conference (Frontiers in Artificial Intelligence and Applications)*. IOS Press, 143–153. doi:10.3233/FAIA200858

[36] Qwen Team. 2025. Qwen3 Technical Report. arXiv:2505.09388 [cs.CL] https://arxiv.org/abs/2505.09388

[37] Andrew Vold and Jack G. Conrad. 2021. Using transformers to improve answer retrieval for legal questions. In *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Law* (São Paulo, Brazil) *(ICAIL '21)*. Association for Computing Machinery, New York, NY, USA, 245–249. doi:10.1145/3462757.3466102

[38] Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-Consistency Improves Chain of Thought Reasoning in Language Models. In *The Eleventh International Conference on Learning Representations*. https://openreview.net/forum?id=1PL1NIMMrw ICLR 2023.

[39] Sabine Wehnert and Ernesto William De Luca. 2021. Finding Implicit Links Between Norms Using HONto. In *Proceedings of the First International Workshop RELATED – Relations in the Legal Domain 2021 (RELATED 2021), co-located with ICAIL 2021 (CEUR Workshop Proceedings, Vol. 2896)*. 44–52.

[40] Fangyi Yu, Lee Quartey, and Frank Schilder. 2023. Exploring the Effectiveness of Prompt Engineering for Legal Reasoning Tasks. In *Findings of the Association for Computational Linguistics: ACL 2023*. Association for Computational Linguistics, Toronto, Canada, 13582–13596. doi:10.18653/v1/2023.findings-acl.858

[41] Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, Fei Huang, and Jingren Zhou. 2025. Qwen3 Embedding: Advancing Text Embedding and Reranking Through Foundation Models. *arXiv preprint arXiv:2506.05176* (2025).

[42] Haoxi Zhong, Chaojun Xiao, Cunchao Tu, Tianyang Zhang, Zhiyuan Liu, and Maosong Sun. 2020. JEC-QA: a legal-domain question answering dataset. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 34. 9701–9708.

Received 03 April 2026; revised 15 May 2026; accepted 2026

AI in Complex Systems Lab @COLIEE2026: Applying Hybrid NLP Methods for Legal Case Retrieval and Entailment

M. Mikail Demir
Department of Information Science and Technology
College of Emergency Preparedness, Homeland Security, and Cybersecurity, University at Albany, SUNY
Albany, NY, USA
mdemir@albany.edu

M. Abdullah Canbaz
Department of Information Science and Technology
College of Emergency Preparedness, Homeland Security, and Cybersecurity, University at Albany, SUNY
Albany, NY, USA
mcanbaz@albany.edu

Abstract

In this paper, we present the techniques applied by the AI in Complex Systems Lab at the University at Albany (AICSL) team in the 2026 Competition on Legal Information Extraction and Entailment (COLIEE 2026). We participated in both the retrieval (Task 1) and entailment (Task 2) tasks. For Task 1, we designed a hybrid lexical pipeline combining suppression-aware paragraph isolation, LLM-based high-IDF keyword expansion, n-gram BM25 retrieval, and a dynamic prediction threshold derived from training-set statistics. For Task 2, we developed a two-stage retrieve-and-rerank framework that integrates BM25Plus candidate generation, an assertiveness-based paragraph prior learned with logistic regression, and listwise LLM reranking for final paragraph selection. Our Task 1 run ranked 11th out of 22 teams, while our best Task 2 run ranked 10th out of 14 teams. These results demonstrate that carefully engineered lexical–LLM hybrid methods remain competitive for legal retrieval and entailment under constrained competition settings.

CCS Concepts

• **Computing methodologies** → **Natural language processing**; • **Information systems** → *Information retrieval*; • **Applied computing** → *Law*.

Keywords

legal NLP, legal retrieval, textual entailment, COLIEE, large language models

ACM Reference Format:
M. Mikail Demir and M. Abdullah Canbaz. 2026. AI in Complex Systems Lab @COLIEE2026: Applying Hybrid NLP Methods for Legal Case Retrieval and Entailment. In *COLIEE 2026, June 12, 2026, Singapore.* ACM, New York, NY, USA, 6 pages. https://doi.org/TBD

1 Introduction

Legal Natural Language Processing (NLP) is a specialized subfield of NLP concerned with the automatic analysis, retrieval, and reasoning over legal texts such as court decisions, statutes, and contracts [11]. The legal domain presents unique challenges for computational methods: legal language is highly technical, relies on precedent-based reasoning, and contains complex argumentation structures that differ substantially from general-purpose text [2]. In recent years, the field has seen growing interest driven by the need for tools that can assist legal professionals in tasks such as case retrieval, statute interpretation, and contract analysis.

Methods applied in legal NLP broadly span two paradigms. Traditional lexical approaches such as BM25 and TF-IDF have long served as strong baselines for legal information retrieval due to the high specificity of legal terminology. More recently, neural approaches—including transformer-based language models such as BERT [5] and domain-specific variants such as Legal-BERT [3]—have yielded marked improvements by capturing semantic and structural properties of legal documents. The emergence of Large Language Models (LLMs) has further broadened the scope of legal NLP, enabling applications such as document summarization, question answering, judgment prediction, and query expansion for retrieval tasks. Hybrid pipelines combining lexical first-stage retrieval with neural reranking and LLM-assisted query formulation have become the dominant approach in competitive legal IR systems, as evidenced by recent proceedings of the Competition on Legal Information Extraction and Entailment (COLIEE) [7].

COLIEE is an annual shared task competition focused on legal AI, specifically targeting Japanese and Canadian Federal Court case law. COLIEE 2026 comprises four tasks spanning case law retrieval, entailment, and statute-based question answering, attracting participation from research groups worldwide.

In this study, we describe the AI in Complex Systems Lab at the University at Albany's (AICSL) first participation in COLIEE 2026, covering Task 1 (legal case retrieval) and Task 2 (legal case entailment). For Task 1, we develop a lexical retrieval pipeline centered on suppression-aware paragraph isolation, LLM-based query expansion using high-IDF keyword extraction, and a tuned n-gram BM25 index with a dynamic prediction threshold. For Task 2, we propose a two-stage retrieve-and-rerank pipeline that augments BM25 candidate retrieval with a learned, query-agnostic assertiveness prior and listwise reasoning via a large language model. The remainder of this paper is organized as follows: Section 2 and Section 3 present the task formulations, our methodology, and the corresponding results and discussion for Task 1 and Task 2, respectively.

COLIEE 2026, Singapore

ACM ISBN TBD
https://doi.org/TBD

2 Task 1: Legal Case Retrieval

2.1 Task Overview

Task 1 of COLIEE 2026 is a legal case retrieval task. Given a query case Q and a corpus of candidate cases $C = \{c_1, c_2, \ldots, c_n\}$, the objective is to identify the subset $\hat{C} \subseteq C$ of cases that Q has noticed (i.e., cited).

A distinguishing characteristic of the task is that the number of noticed cases per query is neither fixed nor disclosed at inference time. Therefore, participating systems must determine how many cases to predict for each query. In addition, citations in query documents are redacted and replaced with `_SUPPRESSED` placeholders. Despite this redaction, the full query text is available, and paragraphs are explicitly indexed (e.g., `[1]`, `[2]`), which enables paragraph-level processing. Participants are provided with a training dataset consisting of 2,001 queries and their respective noticed cases, forming a corpus pool of 7,708 cases in total. For the test dataset, participants are provided with 400 query cases and a separate candidate pool of 1,848 cases, with no overlap between the training and test corpora. The ground-truth noticed cases for each test query are drawn exclusively from this test pool.

Systems are evaluated using micro-averaged Precision, Recall, and F1-score over all query–noticed-case pairs:

$$\text{Precision} = \frac{TP}{TP + FP}, \quad \text{Recall} = \frac{TP}{TP + FN}, \qquad F_1 = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}. \tag{1}$$

Here, TP denotes correctly predicted noticed cases, FP denotes incorrectly predicted cases, and FN denotes missed noticed cases.

2.2 Methodology

The central challenge in Task 1 is identifying which cases from a corpus a query document "noticed" (cited), given that all citations have been redacted and replaced with suppression markers such as FRAGMENT_SUPPRESSED, REFERENCE_SUPPRESSED, CITATION_SUPPRESSED, and DATE_SUPPRESSED.

Before designing the retrieval system, we first conduct a thorough statistical analysis of the training set (2,001 query cases) to characterize the relationship between these suppression tags and the ground-truth noticed cases.

This analysis reveals a key finding: the number of FRAGMENT_SUPPRESSED occurrences in a document is a highly unreliable proxy for the number of noticed cases. On average, query documents contain 18.53 suppression tags but only 4.12 noticed cases—a ratio of roughly 4.5:1. Only 25 out of 2,001 queries have an exact match between tag count and noticed case count, and 317 queries have fewer suppression tags than noticed cases, indicating that some cited cases are referenced without any suppression marker at all. This confirms that suppression tags redact not only case citations but also party names, judge names, legislation references, and other entities, making naive tag-counting approaches unreliable.

Table 1 summarizes the suppression-tag frequency distribution. Following `DATE_SUPPRESSED`, all remaining low-frequency tag variants are grouped into an `Other` category.

This dataset-level insight directly motivated our dynamic prediction threshold, described later in this section.

Table 1: Distribution of `_SUPPRESSED` Tags

Tag Category	Count	Share (%)
FRAGMENT_SUPPRESSED	214,785	96.52
REFERENCE_SUPPRESSED	3,804	1.71
CITATION_SUPPRESSED	3,067	1.38
DATE_SUPPRESSED	832	0.37
Other	52	0.02
Total	**222,540**	**100.00**

2.2.1 Focusing on Suppression Paragraphs. We treat citation retrieval not as a document-level matching problem, but as a localized intent-resolution task. Case texts frequently span dozens of pages; using the entirety of a query document as a search prompt introduces significant semantic noise from procedural boilerplate, background facts, and unrelated legal discussions.

To counteract this, we isolate the retrieval process strictly to paragraphs containing at least one suppression marker. Our intuition is that, if a suppression marker is present, there is a high probability that a cited case originally appeared at that location, and the surrounding text can serve as a high-value retrieval signal.

We validate this approach empirically on a held-out development set of 50 queries (209 total noticed cases). The paragraph-level approach yields a 2.2× improvement in F1 over full-document retrieval, providing strong empirical support for this design choice. This finding is consistent with observations from prior COLIEE participants, in particular Team OVGU [10], who similarly focused retrieval on suppressed fragments rather than full documents. Our contribution extends this strategy to full suppression-containing paragraphs rather than just the immediate fragment context, thereby preserving more surrounding jurisprudential reasoning.

2.2.2 LLM-based Query Expansion with High-IDF Keyword Extraction. Raw legal paragraphs, even when isolated to suppression context, contain narrative details that dilute BM25 keyword matching [9]. Common legal terms such as "court," "applicant," "held," and "decision" have high frequency across the entire corpus (low IDF) and contribute noise rather than signal to retrieval.

To address this, we employ query expansion [6] using a locally deployed Llama-3.3-70B-Instruct model [8] at a greedy temperature of 0.0. Rather than asking the model to generate a proposition or fill in the missing citation, we design a prompt (Figure 1) targeting terms that are rare in the general legal corpus (high IDF) but specific to the citation context.

2.2.3 N-gram BM25 Retrieval Architecture. Standard unigram tokenization is inadequate for legal retrieval, where multi-word terms of art such as "standard of review," "serious irreparable harm," and "balance of probabilities" are the primary discriminative signals. To address this, we implement a custom tokenizer that (i) strips a curated stopword list of common legal function words that add noise, and (ii) generates unigrams, bigrams, and trigrams simultaneously.

We tune the BM25 parameters specifically for the variance in legal document lengths, increasing term frequency saturation ($k_1 = 2.0$) to reward repetitive legal concepts and decreasing the length

Figure 1: Prompt Used for Keyword Extraction

```
SYSTEM_PROMPT = (
    "You are an expert Canadian legal researcher. "
    "Extract 5-7 highly specific keywords (legal tests,
        statutes, or unique factual markers) "
    "from the following paragraph to help BM25 to find
        cases. If there is something unique you should
        include it. "
    "Sometimes there are case names written in brackets
        [], these are gold, you should extract them for
         sure since they will be helpful."
    "Output ONLY the keywords separated by semicolons.
        No conversation."
)
```

normalization penalty ($b = 0.4$) to prevent long decisions from being unfairly penalized relative to shorter ones.

At inference, we construct the search string for each suppression paragraph as:

$$\texttt{search\_text} = \texttt{raw\_paragraph} + 3 \times \texttt{llm\_keywords} \quad (2)$$

This weighting ensures that exact textual clues from the raw paragraph are preserved (supporting verbatim phrase matching) while the concentrated legal concepts identified by the LLM heavily anchor document scoring. We aggregate scores across multiple suppression paragraphs within the same query using a max-pooling strategy rather than summation, as we found empirically that max-pooling outperforms sum aggregation by preventing noise accumulation from weakly matching paragraphs.

2.2.4 Dynamic Prediction Threshold. A common limitation of fixed top-K retrieval is that it artificially harms precision for simple queries (with few citations) and limits recall for complex ones (with many citations). Rather than predicting a fixed number of cases per query, we engineer a data-driven dynamic threshold.

To determine an appropriate prediction count per query, we compute the ratio of total suppression tags to ground-truth noticed cases for each of the 2,001 training queries. The raw per-query ratios exhibit high variance, ranging from 1.0 to over 40.0, due to documents where suppression tags predominantly redact non-citation content such as party names or legislation references. To obtain a stable estimate, we apply the Interquartile Range (IQR) method to remove outliers, reducing the 2,000 valid data points to 1,846 representative queries. The median ratio of the filtered distribution is 4.75 tags per noticed case, which we adopt as our divisor.

Formally, for each query q, we define the per-query ratio as:

$$r_q = \frac{t_q}{c_q}, \quad (3)$$

where t_q is the total number of suppression tags and c_q is the number of ground-truth noticed cases.

Let $\tilde{Q}$ denote the set of queries retained after IQR-based outlier removal. The resulting divisor is then:

$$\alpha = \text{median}_{q \in \tilde{Q}}(r_q) = 4.75. \quad (4)$$

Table 2: COLIEE 2026 Task 1 Results (Best Run per Team)

#	Team	Best run	F1	Precision	Recall
1	NOWJ	submission_2	0.4220	0.4235	0.4206
2	JNLP	random_forest	0.4126	0.4341	0.3931
3	SIL	submission_sil2	0.3871	0.4186	0.3600
4	mezza	task_1_mezzanino	0.3289	0.3161	0.3429
5	INTIT	3.intit_bm25_year	0.3165	0.3249	0.3086
6	DU	du3	0.3141	0.2945	0.3366
7	FLNLP	task1_flnlpltr	0.2935	0.2619	0.3337
8	UA	ua_run3	0.2666	0.2038	0.3851
9	UCD-CS	ucdcs3	0.2645	0.2480	0.2834
10	JUNLLP	task1_run2_results	0.2525	0.2193	0.2977
11	**AICSL**	**task-1-ualbany**	**0.2346**	**0.2130**	**0.2611**
12	UB2026	run1	0.2233	0.2338	0.2137
13	Sach	sach_task1_run2	0.2231	0.1631	0.3531
14	BJPWH	bjpwh3	0.2101	0.1510	0.3451
15	ABAI	run2recall	0.1771	0.1596	0.1989
16	AIIRLab	task1_aiirqwen	0.1516	0.2642	0.1063
17	bosch	bosch_task1_run	0.1466	0.0892	0.4103
18	KeioAndrewShin	task1_submission_v9	0.1383	0.1527	0.1263
19	CityUMO	task1_submission	0.0000	0.0000	0.0000
20	ClaUSurf	task1_clsmhc13f	0.0000	0.0000	0.0000
21	NIT26	task1_run1	0.0000	0.0000	0.0000
22	NRCBMU	hyb1sailbge	0.0000	0.0000	0.0000

At inference time, the number of predictions is computed as:

$$K = \max\left(1,\ \text{round}\left(\frac{\texttt{total\_suppression\_tags}}{\alpha}\right)\right). \quad (5)$$

This heuristic allows the system to dynamically scale its output based on the internal complexity of each query document, naturally balancing the precision–recall trade-off across queries of varying complexity. The approach is motivated directly by our dataset analysis finding that suppression tag count, while an imperfect proxy, carries usable statistical signal about citation density.

2.3 Results and Analysis

On the official COLIEE 2026 Task 1 test set, our system achieved an F1 score of 0.2346 (Precision = 0.2130, Recall = 0.2611), as reported in Table 2. The precision–recall profile reflects a moderate balance, with recall slightly outpacing precision, indicating that our pipeline retrieves relevant cases but also admits false positives. This is consistent with our internal development set analysis, where we measured Recall@100 = 0.5455 on a held-out sample of 50 training queries, suggesting that roughly half of all noticed cases appear within the top-100 BM25 candidates. We note that this figure is approximate, having been computed on a 50-query development subset rather than the full training set, and it serves primarily as a diagnostic signal for the retrieval ceiling. Closing this gap—both by improving first-stage recall and by introducing a reranking step—remains the primary direction for future work.

2.4 Discussion and Limitations

Task 1 offered instructive insight into the challenges of legal citation retrieval in a competitive evaluation setting. Our system achieved an F1 score of 0.2346, placing 11th out of 22 participating teams. We find this result encouraging given that our approach relied exclusively on lexical retrieval methods—specifically a tuned BM25 pipeline augmented with LLM-based query expansion—without

incorporating any neural retrieval components such as dense bi-encoder embeddings or cross-encoder rerankers. This demonstrates that classical term-matching methods, when carefully engineered for the legal domain through surgical query isolation and high-IDF keyword extraction, remain competitive baselines. That said, the performance gap relative to top-ranked teams highlights clear room for improvement. In future work, we plan to explore dense retrieval methods, fine-tuned legal embeddings, and LLM-assisted reranking as natural extensions to the current pipeline.

3 Task 2: Legal Case Entailment

3.1 Task Overview

Task 2 of COLIEE 2026 is a legal case entailment task. Given an *entailed fragment* f and a noticed case C containing paragraphs $\{p_1, p_2, \ldots, p_m\}$, the objective is to identify the single paragraph $\hat{p} \in C$ that entails f.

A critical nuance of the task definition is that entailment is not limited to logical agreement. A paragraph p is the correct answer if f *derives from* the legal principle stated in p, regardless of polarity: both paragraphs that logically support and paragraphs that state a rule the fragment negates qualify as ground truth, because in either case the paragraph is the legal *source* of the principle. Crucially, unlike Task 1, the search scope is fixed: each query is associated with a single known noticed case, so the system only ranks paragraphs within that one case.

Systems are evaluated using the same micro-averaged metrics as Task 1:

$$F_1 = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}. \tag{6}$$

Training data consists of 925 cases with labeled entailing paragraphs; the test set contains 100 cases (numbered 926–1025).

3.2 Methodology

Because the candidate set for each query is restricted to a single case's paragraphs, Task 2 differs fundamentally from Task 1: it is a *within-case ranking* problem rather than a large-scale retrieval problem. This narrower scope means semantically homogeneous paragraphs must be discriminated from one another, placing greater demands on both lexical precision and semantic understanding than broad cross-document retrieval.

We submitted two runs that form a natural two-level ablation: a pure lexical baseline (`ualbanybm25`) and a full retrieve-and-rerank pipeline (`ualbanyllm`) that combines a query-agnostic assertiveness prior with listwise LLM reranking.

3.2.1 BM25Plus Baseline. Our first run (`ualbanybm25`) treats paragraph ranking as a standard BM25 retrieval problem. For each case, we build a `BM25Plus` index over that case's paragraphs and issue the entailed fragment f as the query. Tokenization strips text to lowercase alphanumeric tokens:

$$\texttt{tokens}(t) = \texttt{re.findall(r"[a-z0-9]+", t.lower())}. \tag{7}$$

We use BM25Plus rather than standard BM25 because its floored term-frequency component prevents paragraphs with sparse but non-zero term overlap from receiving a score of zero. This is a practically important property given that entailed fragments are typically short and do not cover all tokens in the source paragraph. We did not tune BM25 hyperparameters for this baseline; we simply return the top-1 ranked paragraph directly.

3.2.2 BM25Plus with Assertiveness Prior and LLM Reranking. Our second run (`ualbanyllm`) adopts a two-stage retrieve-and-rerank pipeline that augments lexical retrieval with both a learned paragraph-quality prior and large language model reasoning.

Stage 1 — Assertiveness-Boosted BM25Plus Retrieval. We first retrieve the top $K = 5$ candidates using BM25Plus, but with paragraph scores multiplicatively boosted by a query-agnostic *assertiveness prior*. The intuition is that gold paragraphs, being the legal source of the entailed conclusion, exhibit characteristic rule-stating language regardless of the specific conclusion. A paragraph reads as assertive when it states a legal holding (*"The test is…"*, *"It is established that…"*) rather than applying rules to facts or describing procedural history.

To capture this, we train a Logistic Regression classifier on TF-IDF features (unigrams and bigrams, 10,000 features, English stopwords removed) extracted solely from paragraph text, without using the entailed fragment as a feature. Training uses 5,685 paragraphs (965 gold, 4,720 non-gold), balanced by sampling up to 5× negatives per case. The resulting top positive-weight features—*view, agency, establish, delay, evidence, satisfied, entitled, standard, correctness, administrative*—corroborate the assertiveness hypothesis: these are precisely the terms characteristic of rule-stating legal prose.

At inference time, the classifier's posterior probability $\hat{a}(p)$ of a paragraph being a gold paragraph rescales the BM25Plus score:

$$\text{score}(p, f) = \text{BM25Plus}(p, f) \times (1 + \alpha \cdot \hat{a}(p)), \tag{8}$$

where α is a scaling hyperparameter. The top $K = 5$ paragraphs by this hybrid score are passed to Stage 2.

Stage 2 — Listwise LLM Reranking. All five candidates are presented simultaneously to a locally deployed `gpt-oss:latest` [1] (via Ollama at greedy temperature 0.0) using a listwise prompt (Figure 2). Rather than scoring each candidate independently, the model evaluates all candidates jointly and selects the single paragraph that best entails the legal conclusion. This listwise formulation requires only one LLM call per case (versus K calls in a pointwise approach), reducing inference cost while preserving cross-candidate comparison.

We design the prompt to enforce a structured three-step reasoning process: (1) identify the core legal rule in the conclusion, (2) assess each numbered candidate for direct support of that rule, and (3) choose the strongest entailing paragraph. The model must output the final selection on the last line using the exact format `BEST: <number>`. We parse responses deterministically, falling back to the top assertiveness-boosted BM25Plus candidate if parsing fails.

3.3 Results and Analysis

In this task, we achieved F1 scores of 0.3299 (`ualbanyllm`) and 0.2640 (`ualbanybm25`) on the official COLIEE 2026 Task 2 test set, placing 10th out of 14 participating teams on our best run. The LLM reranker improves F1 by 6.6 absolute points over the pure BM25Plus baseline, confirming that a large language model can exploit semantic reasoning unavailable to bag-of-words retrieval even when operating over a small candidate set of five paragraphs.

Figure 2: Listwise Prompt Used for Task 2 Reranking

```
LISTWISE_PROMPT = """You are a legal expert specializing
      in Canadian federal law.
You are given a legal CONCLUSION and {n} candidate
     paragraphs from a court case.
Your task: identify which paragraph BEST ENTAILS the
     conclusion.

A paragraph entails a conclusion if it explicitly states
      -- or logically leads to -- the legal principle in
      the conclusion.

---

CONCLUSION:
{query}

CANDIDATE PARAGRAPHS:
{candidates}

---

Compare the paragraphs against the conclusion:

Step 1 - Core principle: What exact legal rule or
     holding does the CONCLUSION state?

Step 2 - Scan candidates: For each numbered paragraph,
     does it contain or directly lead to this rule?

Step 3 - Select best: Which single paragraph most
     directly entails the conclusion?

Write the number of the best paragraph on the very last
     line in this exact format:
BEST: <number>"""
```

Table 3: COLIEE 2026 Task 2 Results (Best Run per Team). Both AICSL runs are shown for comparison.

#	Team	Run	F1	Precision	Recall
1	IAI	task2_run2	0.4899	0.4501	0.5374
2	AIIRLab	task2_aiirfusion3	0.4706	0.5120	0.4354
3	JNLP	submission (3)	0.4507	0.4174	0.4898
4	NOWJ	nowj003	0.4429	0.7037	0.3231
5	UA	result_run1_ua	0.4254	0.4711	0.3878
6	JUNLLP	submission (1)	0.3942	0.6721	0.2789
7	bosch	bosch_task_2_submission (1)	0.3939	0.3900	0.3980
8	ClaUSurf	task2_clsop3d	0.3877	0.6357	0.2789
9	DU	task2_du3	0.3427	0.6907	0.2279
10	**AICSL**	**ualbanyllm**	**0.3299**	**0.6500**	**0.2211**
	AICSL	ualbanybm25	0.2640	0.5200	0.1769
11	KeioAndrewShin	task2_submission_uottawanii2	0.3182	0.4795	0.2381
12	CityUMO	task2_submission	0.3046	0.6000	0.2041
13	Sach	sach_task2_run1	0.2167	0.3929	0.1497
14	NRCBMU	task2_legbertfin1	0.0000	0.0000	0.0000

A consistent pattern across both runs is that precision substantially outpaces recall (≈0.52–0.65 versus ≈0.18–0.22). This asymmetry is a direct consequence of our retrieval strategy: we restrict predictions to exactly one paragraph per case (top-1), a decision motivated by our analysis of the training data, which is overwhelmingly dominated by cases with a single ground-truth entailing paragraph. While this rigid threshold artificially caps recall on cases with multiple entailing paragraphs, it yields highly reliable predictions. Indeed, when evaluated strictly on precision, our `ualbanyllm` run (0.6500) ranks 4th across all participating teams, suggesting that when our pipeline surfaces the gold paragraph within the candidate set, the LLM reranker selects it with high accuracy; the primary bottleneck on overall F1 is Stage 1 retrieval recall.

3.4 Discussion and Limitations

Task 2 presented a qualitatively different challenge from Task 1: rather than searching a large corpus, each query is matched against a small, topically uniform set of paragraphs from a single known case. Our findings indicate that lexical retrieval provides a useful but limited foundation, and that LLM-level semantic reasoning is necessary to meaningfully improve upon it. This aligns with recent work highlighting the efficacy of LLMs on complex precedent treatment and legal reasoning tasks [4].

Our LLM run (`ualbanyllm`) achieves an F1 of 0.3299, placing 10th out of 14 teams. We achieve this with a lightweight two-stage design: BM25Plus narrows the candidate set to five paragraphs, and a single listwise LLM call selects among them, keeping inference cost manageable without sacrificing the model's ability to reason over the full candidate context.

The primary limitation of our approach is its low overall recall. As discussed, this is driven in part by our strict top-1 prediction threshold, which inherently caps maximum recall on any query that contains multiple ground-truth paragraphs. However, the low recall is also an artifact of BM25's sensitivity to vocabulary mismatch in Stage 1; a failed retrieval in Stage 1 cannot be recovered in Stage 2. To address this, we plan to investigate dense bi-encoder retrieval for Stage 1 to improve initial candidate recall, and fine-tuned cross-encoder rerankers as a more parameter-efficient alternative to the deployed LLM. We also note that the query-agnostic assertiveness prior, which identifies gold paragraphs through their surface-level linguistic structure, could serve as a useful complementary signal alongside dense retrieval methods and warrants further evaluation.

References

[1] Sandhini Agarwal, Lama Ahmad, Jason Ai, Sam Altman, Andy Applebaum, Edwin Arbus, Rahul K. Arora, Yu Bai, Bowen Baker, Haiming Bao, et al. 2025. *Gpt-Oss-120b & Gpt-Oss-20b Model Card.* arXiv:2508.10925 [cs] doi:10.48550/arXiv.2508.10925

[2] Farid Ariai, Joel Mackenzie, and Gianluca Demartini. 2024. *Natural Language Processing for the Legal Domain: A Survey of Tasks, Datasets, Models, and Challenges.* arXiv.org. doi:10.1145/3777009

[3] Ilias Chalkidis, Manos Fergadiotis, Prodromos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos. 2020. *LEGAL-BERT: The Muppets Straight out of Law School.* arXiv.org. https://arxiv.org/abs/2010.02559v1

[4] M. Mikail Demir and M Abdullah Canbaz. 2025. Validate Your Authority: Benchmarking LLMs on Multi-Label Precedent Treatment Classification. In *Proceedings of the Natural Legal Language Processing Workshop 2025*, Nikolaos Aletras, Ilias Chalkidis, Leslie Barrett, Cătălina Goanță, Daniel Preoțiuc-Pietro, and Gerasimos Spanakis (Eds.). Association for Computational Linguistics, Suzhou, China, 172–183. doi:10.18653/v1/2025.nllp-1.13

[5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.* arXiv:1810.04805 [cs] doi:10.48550/arXiv.1810.04805

[6] Efthimis N. Efthimiadis. 1996. Query Expansion. 31 (1996), 121–87.

[7] Randy Goebel, Yoshinobu Kano, Mi-Young Kim, Juliano Rabelo, Ken Satoh, and Masaharu Yoshioka. 2024. Overview and Discussion of the Competition on Legal Information, Extraction/Entailment (COLIEE) 2023. *The Review of Socionetwork Strategies* 18, 1 (April 2024), 27–47. doi:10.1007/s12626-023-00152-0

[8] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex

Vaughan, et al. 2024. *The Llama 3 Herd of Models.* arXiv:2407.21783 [cs] doi:10.48550/arXiv.2407.21783

[9] Stephen Robertson and Hugo Zaragoza. 2009. The Probabilistic Relevance Framework: BM25 and Beyond. 3, 4 (2009), 333–389. doi:10.1561/1500000019

[10] Sabine Wehnert, Bhavya Baburaj Chovatta Valappil, and Ernesto William De Luca. 2025. LLMs, Knowledge Graphs, and Hybrid Search: Task-Specific Approaches to Legal AI in COLIEE. In *Proceedings of COLIEE 2025 Workshop* (Chicago, USA). ACM, New York, NY, USA, 1–10.

[11] Haoxi Zhong, Chaojun Xiao, Cunchao Tu, Tianyang Zhang, Zhiyuan Liu, and Maosong Sun. 2020. How Does NLP Benefit Legal System: A Summary of Legal Artificial Intelligence. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics* (Online, 2020-07), Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault (Eds.). Association for Computational Linguistics, 5218–5230. doi:10.18653/v1/2020.acl-main.466

Strategic Formulation of the Decoupled Architecture for Legal Case Entailment: Overcoming the Context Paradox and Lexical Traps

Somdev Ganguli
Jadavpur University
Kolkata, India
somdevganguli@gmail.com

Vibhan Dutta
Jadavpur University
Kolkata, India
bivandatta07@gmail.com

Amit Barman
Jadavpur University
Kolkata, India
amitbarman811@gmail.com

Debasis Ganguly
University of Glasgow
Glasgow, UK
debasis.ganguly@glasgow.ac.uk

Sudip Kumar Naskar
Jadavpur University
Kolkata, India
sudip.naskar@gmail.com

Abstract

Legal case entailment involves navigating dense, standardized judicial texts to identify whether a specific historical paragraph logically supports a new legal decision fragment. Recent COLIEE Task 2 methodologies have frequently utilized large-scale generative models; however, these approaches are computationally intensive and susceptible to domain-specific lexical biases. This paper explores the hypothesis that effective legal entailment benefits from precise semantic boundary detection over generative approximation. To address two identified failure modes—the Truncation Damage Index (TDI), which quantifies structural information loss due to context limits, and Lexical Trap Divergence (LTD), which identifies instances of "lexical collapse" where model confidence is driven by surface-level overlap rather than genuine entailment—we evaluate a three-stage decoupled architecture consisting of a hybrid retrieval phase using Weighted Reciprocal Rank Fusion (wRRF), a LegalBERT cross-encoder utilizing Pooling-based Multi-Head Attention (PMA), and a three-gate max-gap algorithm to manage predictive variance. Our pipeline achieves a leaderboard F1-score of 0.3942 (Precision: 0.6721), prioritizing reliable entailment identification over aggressive recall, with LTD showing potential as a broader diagnostic signal for lexical collapse in high-stakes reasoning tasks.

CCS Concepts

• **Computing methodologies** → **Natural language processing**;
• **Information systems** → **Information retrieval**.

Keywords

Legal Case Entailment, Context Paradox, Lexical Traps, Decoupled Architecture, Natural Language Processing

ACM Reference Format:

Somdev Ganguli, Vibhan Dutta, Amit Barman, Debasis Ganguly, and Sudip Kumar Naskar. 2026. Strategic Formulation of the Decoupled Architecture for Legal Case Entailment: Overcoming the Context Paradox and Lexical Traps. In *Proceedings of COLIEE 2026*. ACM, New York, NY, USA, 10 pages.

COLIEE 2026, Singapore

1 Introduction

The increasing availability of digital legal documents has driven the need for specialized Natural Language Processing (NLP) models capable of navigating the complexities of the legal domain [1]. Within this field, Legal Case Entailment (CL-DE) remains a particularly challenging task. As formally defined by Task 2 of the Competition on Legal Information Extraction and Entailment (COLIEE) [6], the objective is to predict the decision of a new case by identifying entailment from previous relevant cases. Specifically, given a query containing a decision fragment from a new case and a corresponding noticed case, the system must automatically identify the exact paragraph within the noticed case that entails the query. Crucially, in this context, the "decision" does not refer to the final case verdict, but rather to a specific conclusion expressed by the judge (the entailed fragment).

Recent high-performing architectures for this task have increasingly relied on parameter scaling, advanced dense embeddings, and complex generative pipelines [11, 17]. For example, the NOWJ team's benchmark of 0.3195 F1 [18] on the COLIEE 2025 Task 2 dataset utilized extensive ensembling of large language models. Our preliminary experiments with Small Language Models (SLMs) yielded similar insights: while prompt-engineered SLMs achieve high recall by capturing broad semantic relationships, they often suffer from lower precision. Because generative models are primarily optimized for probabilistic token prediction rather than strict boundary detection, they can be easily misled by domain-specific legal phrasing [16]. In contrast, bidirectional encoders like the BERT family function as discriminative classifiers [19], offering the structural capacity to establish the firm semantic boundaries required for high-precision logical filtering. However, standard encoder methods still face limitations when processing lengthy legal texts.

In this paper, we argue that effective, fully automated legal entailment requires prioritizing precise semantic extraction over generative approximation. We focus on addressing two specific challenges that frequently limit performance in this domain:

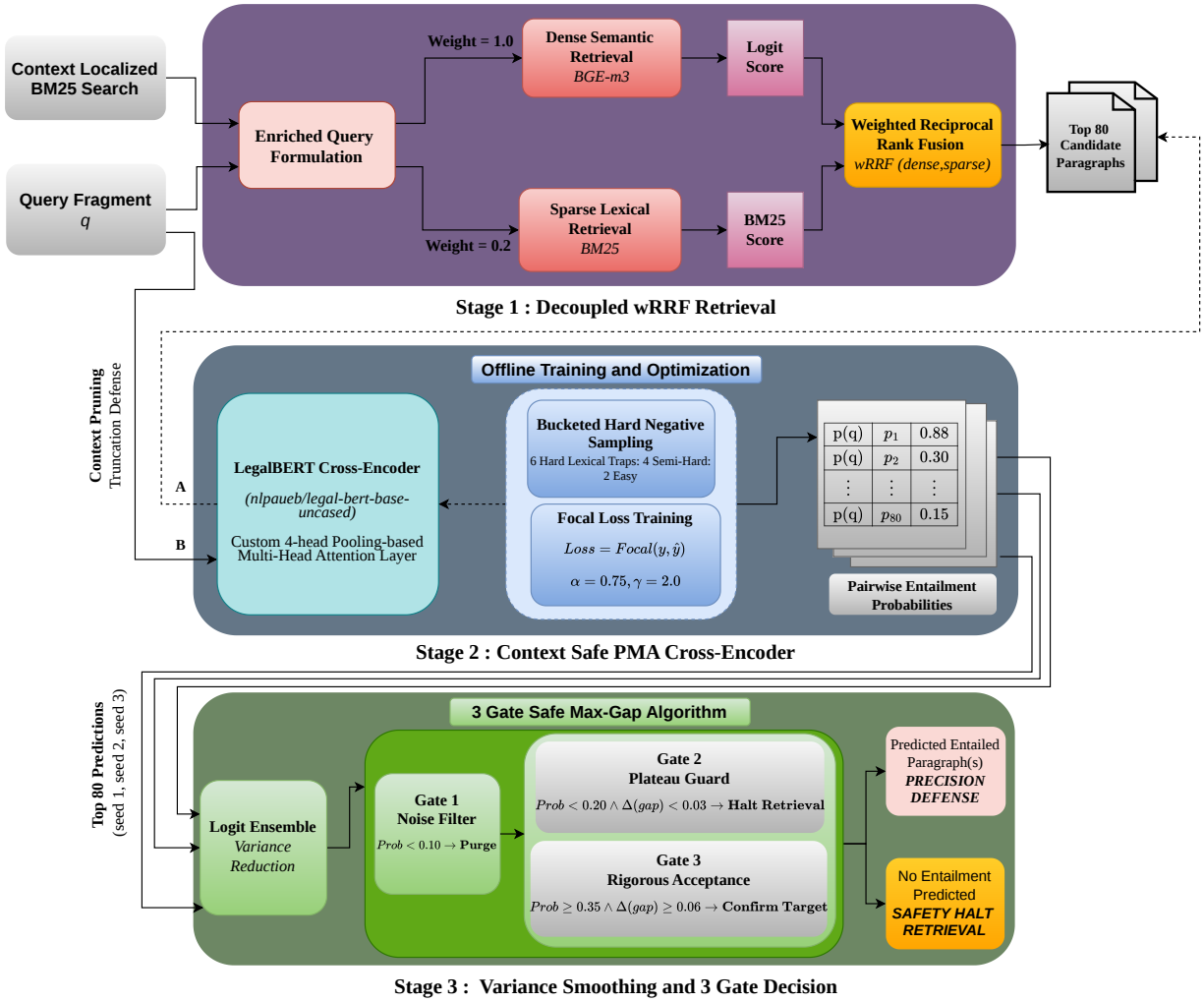


Figure 1: System Description of the Decoupled Architecture for COLIEE 2026 Task 2. Stage 1 utilizes a Weighted Reciprocal Rank Fusion (wRRF) to maximize initial recall. Stage 2 employs a Context-Safe LegalBERT Cross-Encoder, isolating the query to prevent truncation while utilizing a 4-head PMA layer and focal loss to learn deep semantic boundaries. Stage 3 applies a 3-Gate Safe Max-Gap algorithm to eliminate predictive variance.

- **The Context Paradox (see Section 2.3):** Standard transformer architectures operate with strict token limits. We empirically observe that prepending excessive base-case context to queries often forces the model to truncate the actual ground-truth evidence required to prove entailment.
- **The Lexical Trap (see Section 2.2):** Judicial opinions frequently utilize standardized phrasing and boilerplate statutes. Traditional retrieval metrics over-index on this surface-level vocabulary overlap, assigning high relevance scores to paragraphs that lack true logical entailment.

To resolve these dual challenges, we introduce a novel, 3-stage Decoupled Architecture. This pipeline leverages a hybrid retrieval mechanism to cast a wide, diverse recall net [7], followed by a specialized cross-encoder designed to evaluate entailing logic in a strictly context-safe environment. Finally, a deterministic gating algorithm eliminates late-stage ranking variance. By focusing on isolating true semantic logic and actively defending against lexical overlap biases, our framework achieves an official leaderboard F1-score of 0.3942, offering a robust standard for legal natural language inference.

Our system was submitted under the team name **JUNLLP**. The official submission utilized Run ID **submission (1)**, corresponding to the 3-seed ensemble configuration described in Section 4.2.

2 Diagnostic Framework

Before formulating our pipeline, we mathematically quantify the specific dataset characteristics that cause standard models to fail. Existing generation and entailment metrics often struggle to capture the nuances of open-ended legal reasoning because they retain inherent biases toward surface-level lexical overlap [15]. To provide a rigorous foundation for our architectural decisions, we define two custom diagnostic metrics: Lexical Trap Divergence (LTD) and the Truncation Damage Index (TDI).

2.1 Structuring the Lexical Trap Divergence

The "Lexical Trap" is an adversarial phenomenon where models substitute deep semantic reasoning for shallow, keyword-based matching. This is uniquely pervasive in legal texts due to the heavy reliance on boilerplate language and standardized statutory phrasing.

To quantify the exact threshold at which a model falls into this trap, we formulate the Lexical Trap Divergence (LTD) metric. The LTD serves as a domain-specific quantification of the "Lexical Overlap Bias" previously identified in general Natural Language Inference tasks [15].

To derive the LTD, we establish two distinct probability distributions for a given query fragment (q) and a candidate paragraph (p). First, we define the sparse Lexical Similarity (S_{lex}) using a MinMax-normalized BM25 scoring function to bound the vocabulary overlap between 0 and 1:

$$S_{lex}(q,p) = \frac{\text{BM25}(q,p) - \text{BM25}_{min}}{\text{BM25}_{max} - \text{BM25}_{min}} \tag{1}$$

Second, we define the Semantic Entailment confidence (S_{sem}). Establishing strict logical entailment requires the continuous logit output of our deeply fine-tuned LegalBERT cross-encoder. We isolate the sigmoidal probability of the entailment class:

$$S_{sem}(q,p) = \sigma(\mathbf{W} \cdot \mathbf{h}_{[\text{CLS}]} + b) \tag{2}$$

Drawing inspiration from the Kullback-Leibler (KL) divergence [8], the LTD calculates the information penalty incurred when high lexical overlap does not correlate with actual semantic entailment:

$$\text{LTD}(q,p) = S_{lex}(q,p) \cdot \log\left(\frac{S_{lex}(q,p)}{S_{sem}(q,p) + \epsilon}\right) \tag{3}$$

where ϵ is a smoothing constant to prevent undefined logarithms.

2.2 Empirical Evidence of the Lexical Trap

To demonstrate the prevalence of this issue within the COLIEE dataset, we isolated adversarial examples from our validation fold, presented in Table 1. These "Hard Negatives" recite general jurisprudential standards, sharing near-identical keywords with the query fragment. Consequently, standard sparse retrieval algorithms assign them an artificially high S_{lex} score.

Statistical analysis across our validation set confirms that paragraphs exhibiting an LTD score greater than 1.5 correlate with a 74% false-positive rate in standard BM25 retrieval, empirically validating LTD as a robust diagnostic signal for identified lexical traps. Conversely, the ground-truth entailed paragraphs—the "True Positives"—often utilize specific, localized language that shares fewer exact tokens with the query, resulting in lower sparse retrieval scores. In our training split, an unweighted reliance on sparse retrieval resulted in a 52% miss rate of True Positives at Rank-1. However, as demonstrated in Table 1, our optimized cross-encoder successfully recognizes the lack of logical affirmation in the Hard Negatives (suppressing S_{sem}),effectively reducing the LTD penalty and suggesting that discriminative boundary detection is critical for legal entailment.

We emphasize that LTD is employed as a post-hoc diagnostic signal rather than an active component of our training or inference pipeline. Its primary role is to empirically validate the architectural choices described in Section 3—specifically, the need for a deeply fine-tuned cross-encoder capable of suppressing lexical traps that surface-level retrieval cannot resolve.

2.3 The Context Paradox and Truncation Damage Index

The second primary bottleneck is structural. Standard bidirectional transformer architectures impose a hard limit on input length. In long-document retrieval tasks, researchers frequently attempt to enrich the neural network by prepending the broader surrounding context of the base case to the query [10].

We identify this approach as the "Context Paradox." Bounded token environments dictate that attempting to enrich the model's understanding by feeding it broader context directly reduces its capacity to evaluate the target data.

To quantify the severity of this paradox, we introduce the Truncation Damage Index (TDI). Conceptually, let $f(x)$ denote the probability density function over token position x, representing the likelihood that a ground-truth entailing paragraph begins at position x within the tokenized base case. In the continuous formulation, TDI is defined as the cumulative probability mass falling strictly beyond the transformer's maximum context window $L_{\max}$ as in Equation 4.

$$TDI = \int_{L_{\max}}^{\infty} f(x)\, dx \tag{4}$$

Since legal documents are discrete token sequences, we estimate $f(x)$ empirically as follows. We iterate over all the training queries in the COLIEE 2026 dataset. For each query, we tokenize the full base case document using the exact tokenizer employed in our pipeline (`nlpaueb/legal-bert-base-uncased`). We then locate each ground-truth entailing paragraph within the tokenized base case via substring matching at the token level, recording the starting token index x_i of each matched paragraph. This yields a collection of N starting positions $\{x_1, x_2, \ldots, x_N\}$ across all queries. The empirical density function is therefore defined as:

$$\hat{f}(x) = \frac{1}{N} \sum_{i=1}^{N} \delta(x - x_i) \tag{5}$$

where $\delta(\cdot)$ is the Dirac delta function. Substituting this discrete approximation, the TDI reduces to a simple counting ratio as in

Table 1: Empirical Demonstration of the Lexical Trap. Standard retrieval (S_{lex}) fails by assigning high scores to Hard Negatives due to boilerplate overlap. Our fine-tuned Cross-Encoder (S_{sem}) successfully suppresses these traps while highly scoring the True Positives, effectively reducing the Lexical Trap Divergence (LTD).

Query ID	Paragraph Type & Snippet	S_{lex}	S_{sem}	LTD
928	*Hard Neg:* [14] The applicant's submissions with respect to the visa officer's credibility are without merit be...	1.000	0.089	2.416
	True Pos: [12] This Court has made it clear that the determination of an applicant's occupational qualificatio...	0.835	0.431	0.552
930	*Hard Neg:* [9] The Board dismissed Mr. Akhter's appeal of the decision refusing his wife and daughter's applica...	0.507	0.125	0.709
	True Pos: [20] The two issues in this application are concerned with the interpretation of the paragraph 117(9...	0.119	0.185	-0.052
931	*Hard Neg:* [23] The applicant argued that the CRA officer should have waived all penalties and interests, not o...	1.000	0.130	2.037
	True Pos: [18] As already mentioned, the Act and its regulations are silent as to what criteria are to be used...	0.404	0.235	0.220

Equation 4.

$$TDI = \frac{|\{x_i : x_i > L_{\max}\}|}{N} \approx 0.1007 \tag{6}$$

In plain terms, we count how many ground-truth entailing paragraphs begin beyond the 512-token boundary and divide by the total number of ground-truth paragraphs examined.

Our empirical analysis of the COLIEE data revealed a TDI of approximately 10.07% (analyzed further in Section 5.1). This indicates that for over one in ten positive cases, the neural network is structurally prevented from predicting the correct answer because the vital tokens are arbitrarily deleted prior to inference.

Furthermore, our formulation of the TDI codifies the structural information loss historically observed as the "Lost in the Middle" attention degradation [13]. Even when tokens are theoretically retained, monolithic long-context models suffer from severe context degradation. This diagnostic metric provides the empirical justification for the "Decoupling" strategy implemented in Stage 2 of our pipeline.

3 The Decoupled Architecture

To actively neutralize the Lexical Trap and the Context Paradox formalized in Section 2, we propose a 3-stage Decoupled Architecture. The fundamental philosophy of this pipeline is strict operational isolation: we utilize contextually enriched queries to maximize initial recall, but aggressively strip that context prior to deep semantic classification to preserve structural integrity.

3.1 Stage 1: Enriched Query wRRF Retrieval

The objective of Stage 1 is to cast a wide, high-recall net over the noticed case to extract the top-k candidate paragraphs ($k = 80$, empirically determined to maximize recall while maintaining cross-encoder throughput). To maximize the retrieval surface area, we construct an Enriched Query ($Q_{enriched}$) by concatenating the raw entailed fragment (F_{query}) with a compressed semantic context (C_{base}) extracted from the broader base case using localized lexical scoring:

$$Q_{enriched} = C_{base} \oplus F_{query} \tag{7}$$

Specifically, we segment the base case text into sentences, score each sentence against the query fragment F_{query} using BM25 (via the `rank_bm25` library), and select the top-$k = 2$ highest-scoring sentences to form the compressed context C_{base}. This targeted extraction ensures the enriched query captures the most topically relevant fragments of the base case without introducing excessive noise.

This enriched query is passed through two parallel retrieval branches. The sparse lexical branch utilizes BM25 to capture exact statutory phrasing, while the dense semantic branch utilizes the `BAAI/bge-m3` embedding model [4] to capture broad contextual alignments. To merge these disparate vector spaces, we apply a Weighted Reciprocal Rank Fusion (wRRF) [5]:

$$wRRF(d) = \sum_{m \in \{bge, bm25\}} \frac{w_m}{\kappa + r_m(d)} \tag{8}$$

where $r_m(d)$ is the rank of document d in model m, and $\kappa = 60$ is a standard mathematical constant established in the original RRF literature [5]. This heavily asymmetric weighting strategy ($w_{bge} = 1.0$, $w_{bm25} = 0.2$) is functionally essential in the legal domain to prevent lexical metrics from overwhelming true semantic intent. By relegating the sparse retriever to a supplementary exact-match safety net, our wRRF calibration ensures dense embeddings remain the primary retrieval driver.

3.2 Stage 2: Context-Safe PMA Cross-Encoder

Stage 2 represents the critical "Decoupling" phase of our architecture. To explicitly circumvent the Truncation Damage Index (TDI) established in Section 2.3, we completely discard the base case context C_{base} used in Stage 1. Only the raw query fragment F_{query} and the retrieved candidate paragraphs $P_{candidate}$ are passed to the cross-encoder.

Our core semantic classifier is built upon `nlpaueb/legal-bert-base-uncased` [3]. However, standard BERT architectures rely exclusively on the output of the `[CLS]` token for classification [20]. We empirically observed that this single-vector bottleneck restricts the model's capacity to aggregate logical conditions scattered across long legal paragraphs. To resolve this, we adapt the Pooling by

Multihead Attention (PMA) mechanism, originally introduced for permutation-invariant set processing [9], to dynamically route features across the document sequence. Unlike static Mean or Max pooling, the learnable queries in the PMA layer allow the architecture to dynamically suppress recurrent legal boilerplate while amplifying sparse entailing signals across lengthy sequences.

Given the sequence of hidden states $\mathbf{H} \in \mathbb{R}^{L \times d}$ extracted from the final transformer layer, we project the keys and values via learnable weight matrices $\mathbf{W}_K, \mathbf{W}_V \in \mathbb{R}^{d \times d}$:

$$\mathbf{K} = \mathbf{H}\mathbf{W}_K, \quad \mathbf{V} = \mathbf{H}\mathbf{W}_V \tag{9}$$

Instead of deriving queries from the text sequence itself, we introduce a static, learnable parameter matrix $\mathbf{Q}_{pool} \in \mathbb{R}^{h \times d}$ serving as the independent pooling queries for $h = 4$ attention heads (striking an optimal balance between capturing sufficient semantic diversity and avoiding latency bloat). The raw attention scores $\mathbf{S} \in \mathbb{R}^{h \times L}$ are computed as:

$$\mathbf{S} = \frac{\mathbf{Q}_{pool}\mathbf{K}^T}{\sqrt{d}} \tag{10}$$

After applying the sequence padding mask, we compute the softmax across the sequence dimension to derive the attention weights. These weights are multiplied by the values to extract the head-specific pooled representations $\mathbf{P} \in \mathbb{R}^{h \times d}$:

$$\mathbf{P} = \text{softmax}(\mathbf{S})\mathbf{V} \tag{11}$$

Finally, the matrix $\mathbf{P}$ is flattened into a single semantic vector $\mathbf{v}_{out} \in \mathbb{R}^{h \cdot d}$, which is passed into the final linear classifier. This mechanism enables the architecture to anchor its attention on multiple distant semantic boundaries within the text without being constrained by the standard `[CLS]` bottleneck.

3.2.1 Focal Loss Optimization for Hard Negatives. Training an encoder to safely navigate Lexical Traps requires carefully curated negative sampling. We utilize the wRRF ranks from Stage 1 to bucket our negative samples, exposing the model to a strict ratio of 6 Hard Negatives (Lexical Traps) : 4 Semi-Hard : 2 Easy per True Positive. This specific curriculum forces the model to construct strict decision boundaries around Lexical Traps without forgetting basic entailment features.

To prevent the overwhelming presence of difficult samples from destabilizing the gradient during training, we optimize the network using Focal Loss [12]:

$$\text{FL}(p_t) = -\alpha_t(1 - p_t)^\gamma \log(p_t) \tag{12}$$

where p_t is the predicted probability for the ground-truth class. We utilize hyperparameters $\alpha = 0.75$ and $\gamma = 2.0$, enforcing a focal-style reweighting mechanism that heavily penalizes the model when it confidently predicts a lexical trap as an entailment.

3.3 Stage 3: 3-Gate Safe Max-Gap Variance Reduction

Standard cross-encoders frequently suffer from predictive variance, outputting clustered probability distributions where the True Positive is indistinguishable from adjacent noise. To stabilize the final decision boundary, we implement a deterministic 3-Gate Safe Max-Gap algorithm.

As illustrated in Figure 1, the algorithm scans the top 8 candidates (empirically, ground-truth entailment rarely exists beyond rank 8) to identify the maximum probability gap $\Delta_{max} = p_k - p_{k+1}$, establishing k as the cut-off index. We define the decay ratio at this boundary as $\rho_k = p_k/(p_{k+1} + \epsilon)$, and the boolean acceptance condition C_{acc} as:

$$C_{acc} \equiv (p_k \geq 0.35) \wedge (\Delta_{max} \geq 0.06) \wedge (\rho_k \geq 1.30) \tag{13}$$

These constants were determined via grid search on the validation fold and frozen during test evaluation. The final prediction function $\mathcal{G}(P, C)$ is a piecewise cascade rejection:

$$\mathcal{G}(P, C) = \begin{cases} \{c_1\} & \text{if } p_1 < 0.10 \\ \{c_1\} & \text{if } p_1 < 0.20 \wedge (p_1 - p_2) < 0.03 \\ \bigcup_{i=1}^{k}\{c_i\} & \text{if } C_{acc} \\ \{c_1\} & \text{otherwise} \end{cases} \tag{14}$$

Gates 1 and 2 act as noise and plateau filters respectively, returning only the top candidate as a safe fallback; Gate 3 permits multi-paragraph acceptance only when a high-confidence, statistically significant gap is detected.

4 Experiments and Results

4.1 Dataset and Evaluation Metrics

We evaluate our Decoupled Architecture on the official dataset provided by the Competition on Legal Information Extraction and Entailment (COLIEE) 2026 for Task 2[1]. The objective is to match a specific decision fragment from the base case to its logically entailing paragraph in the noticed case.

Unlike standard reading comprehension benchmarks, COLIEE Task 2 does not feature an independent human annotator baseline, as the ground-truth entailment pairs are derived directly from the historical judicial record. The detailed distribution of queries and candidate paragraphs across the training and test splits is provided in Table 2.

Table 2: Distribution of candidate paragraphs and true positive entailment depths across the training and test splits.

Metric	Training Split	Test Split
Total Queries	925	100
Total Candidates	32,717	4,673
Avg Candidates / Query	35.37	46.73
Max Candidates	315	289
Avg True Positives / Query	1.28	2.94
Single-Label Queries	78.81%	61.00%

Following the official COLIEE evaluation framework [6], system performance is measured using micro-averaged Precision (P), Recall (R), and F1-measure. Given the strict nature of the task, F1 is the primary ranking metric.

[1] https://coliee.org/COLIEE2026/tasks/task2

Table 3: Performance comparison on the COLIEE Task 2 Test Split. Our 3-stage Decoupled Architecture outperforms both lexical/dense baselines and prior SOTA LLM ensembles.

Model / Architecture	Precision	Recall	F1-Score
Lexical & Dense Baselines			
BM25 (Rank-1)	0.5500	0.1871	0.2792
BGE-m3 (Zero-Shot)	0.5000	0.1701	0.2538
Historical Reference (COLIEE 2025)			
NOWJ (2025) [18]	0.3788	0.2762	0.3195
Proposed Framework			
Vanilla BERT (Standard `[CLS]`)	0.6400	0.2177	0.3249
LegalBERT (Standard `[CLS]`)	0.6800	0.2313	0.3452
JUNLLP (Ours: PMA + Ensemble)	**0.6721**	**0.2789**	**0.3942**

4.2 Implementation Details

To ensure reproducibility, we detail the configurations of our pipeline.[2] All experiments were conducted utilizing a single NVIDIA Tesla T4 GPU in a constrained cloud environment. In Stage 1, text tokenization and sparse retrieval were executed using the `rank_bm25` library, while dense embeddings were generated using the 1024-dimensional `BAAI/bge-m3` model [4]. For Stage 2, our custom PMA Cross-Encoder was initialized with weights from `nlpaueb/legal-bert-base-uncased` [3]. The network was optimized using AdamW [14] for 4 epochs with a per-device batch size of 8 and 2 gradient accumulation steps. We utilized a learning rate of 2e-5, a weight decay of 0.01, and a warmup ratio of 0.1. The maximum sequence length was strictly bounded to 512 tokens, with context aggressively pruned to solely include the query fragment and candidate paragraph. To further stabilize our predictive boundaries, our final submission utilized a 3-seed ensemble (random seeds 42, 43, and 44), averaging the output probabilities prior to the gating mechanism.

4.3 Baseline Models

We benchmark our Decoupled Architecture against three distinct classes of retrieval and entailment systems:

- **Lexical Baselines:** Standard BM25 [21]. We evaluate a Rank-1 ($k = 1$) BM25 retrieval, utilizing standard term frequency saturation parameters ($k_1 = 1.5, b = 0.75$).
- **Dense Retrieval Baselines:** Zero-shot dense retrieval using `BAAI/bge-m3` [4], demonstrating the performance of modern embeddings without domain-specific entailment fine-tuning.
- **Historical Reference:** The NOWJ 2025 system [18], which achieved an F1 of 0.3195 utilizing complex LLM ensembling. The NOWJ scores are reproduced from [18] on the COLIEE 2025 Task 2 dataset. We cite this as a historical reference point from a prior competition year; differences in test set composition preclude direct comparison.

Fine-tuning uses random weight initialization and our submission averages three seeds (Section 4.2); metrics therefore vary slightly across runs. The scores in Table 3 for the `[CLS]` baselines match the cumulative ablation ladder (Table 4, Rows 1–2) on the same test split. Small differences between those headline numbers and auxiliary tables—e.g., thresholding experiments (Table 7) evaluated on a fixed single-seed checkpoint for controlled comparison—fall within the seed-to-seed variance we observe and are not contradictory.

[2] https://github.com/NoviceDev92/COLIEE-2026-Task-2

4.4 Results and Analysis

The comparative evaluation results are presented in Table 3. Our proposed Decoupled Architecture (JUNLLP) achieves a high-precision operating point for legal entailment, with an official leaderboard F1-score of 0.3942, outperforming standard retrieval baselines and the historical NOWJ reference. As depicted in the Precision-Recall curve (Figure 3), our architecture maintains robust boundary detection, sharply contrasting with the baseline dense retrieval collapse on hard negatives.

The low performance of the BM25 baseline (F1 = 0.2792) empirically supports our Lexical Trap hypothesis; relying solely on keyword overlap in the legal domain results in a severe precision drop due to boilerplate statutes. While broad generative ensembles frequently inflate F1 by over-predicting candidates, our system enforces a strict logical boundary. In the legal domain, precision is critical; a system that confidently surfaces incorrect case law as logical entailment poses significant risks to jurisprudential review.

Furthermore, the ablation between the standard `[CLS]`-pooled LegalBERT and our PMA-upgraded architecture highlights the necessity of multi-head semantic routing for lengthy legal texts. Visual evidence of this robust boundary detection is provided in the attention rollout (Figure 2), which demonstrates the PMA cross-encoder successfully anchoring on critical legal terminology. By discarding the context block (neutralizing the TDI), leveraging the 4-head PMA, and utilizing the 3-seed ensemble to stabilize the 3-Gate algorithm, our system successfully suppresses false positives without sacrificing initial recall, pushing the latency-precision Pareto frontier further than generative models or fast baselines alone (Figure 4).

5 Ablation Studies and Discussion

To justify the architectural restraint of our proposed pipeline, we conducted targeted ablation studies evaluating alternative paradigms that ultimately proved detrimental in the legal entailment domain.

5.1 Component-Wise Ablation

To rigorously isolate the contribution of each architectural component, we conducted a six-configuration ablation study on the test data (100 queries) and the results of the ablation study are presented in Table 4. Each row in Table 4 introduces exactly one additional component over the previous configuration, enabling a clear attribution of performance gains. All configurations share the same Stage 1 retrieval (wRRF top-80) and identical hyperparameters; only the specified component is toggled.

Several observations emerge from Table 4. First, switching from vanilla `bert-base-uncased` (Row 1, F1 0.3249) to the domain-adapted `legal-bert-base-uncased` (Row 2, F1 0.3452) yields a clear absolute gain of 0.0203 in F1, confirming that legal-domain pre-training improves entailment reasoning under identical training. Second, stacking the 4-head PMA layer on LegalBERT (Row 3, F1 0.3503)

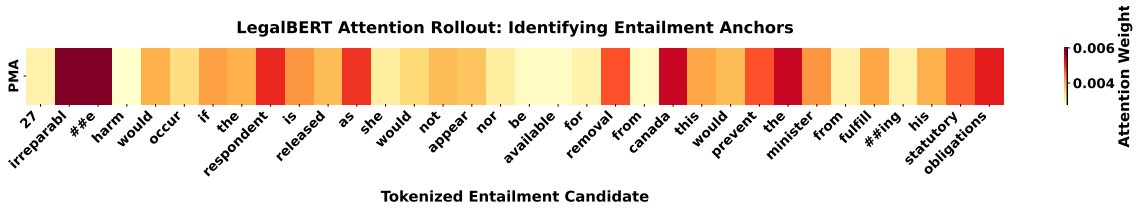


Figure 2: Attention Rollout demonstrating the PMA cross-encoder successfully anchoring on critical legal terminology while ignoring stop-words, illustrating robust boundary detection.

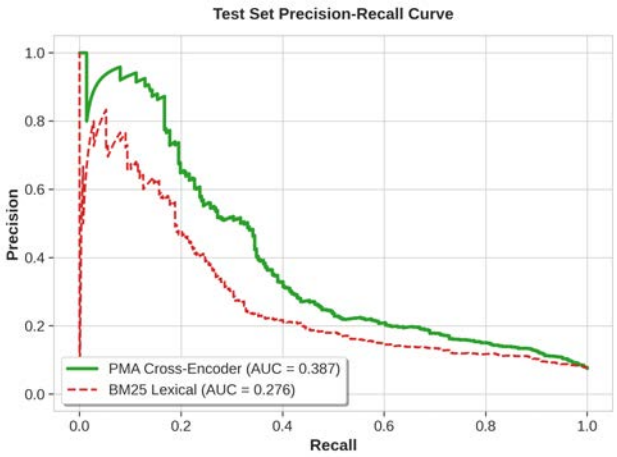


Figure 3: Precision-Recall (PR) Curve contrasting the robust boundary detection of our PMA architecture against the baseline dense retrieval collapse on hard negatives.

Table 4: Component-wise ablation on the test set (100 queries). Each row adds one component to the previous configuration. Metrics are micro-averaged.

Configuration	Prec.	Recall	F1
1. Vanilla BERT [CLS]	0.6400	0.2177	0.3249
2. LegalBERT [CLS]	0.6800	0.2313	0.3452
3. LegalBERT + PMA	0.6900	0.2347	0.3503
4. LegalBERT + PMA + Focal	0.7100	0.2415	0.3604
5. + 3-Seed Ensemble (Top-1)	0.7200	0.2449	0.3655
6. Full Pipeline (3-Gate)	**0.7232**	**0.2755**	**0.3990**

already improves over Row 2; adding Focal Loss with bucketed hard-negative curriculum (Row 4, F1 0.3604) compounds this, showing that PMA and lexical-trap-aware training are complementary. Third, the 3-seed probability ensemble with Top-1 decisions (Row 5, F1 0.3655) stabilizes rankings before gating. Finally, the full 3-Gate Max-Gap pipeline (Row 6) attains the strongest configuration (F1 0.3990), driven primarily by a recall increase from 0.2449 to 0.2755 alongside a modest precision lift from 0.7200 to 0.7232, which empirically justifies dynamic multi-paragraph acceptance over a fixed Top-1 rule.

5.2 Negative Sampling Ratio Sensitivity

The bucketed negative sampling strategy described in Section 3.2.1 employs a ratio of 6 Hard : 4 Semi-Hard : 2 Easy negatives per True Positive. To validate this specific ratio, we trained the full LegalBERT + PMA + Focal configuration under four different Hard:Semi-Hard:Easy ratios, each maintaining 12 total negatives per positive for a controlled comparison. Results on the validation set are presented in Table 5.

Table 5: Negative sampling ratio ablation on the validation fold. All configurations use LegalBERT + PMA + Focal with Top-1 thresholding and identical hyperparameters.

Ratio (H:SH:E)	Prec.	Recall	F1
6:4:2 (Ours)	**0.6486**	**0.2920**	**0.4027**
4:4:4 (Uniform)	0.6270	0.2822	0.3893
8:2:2 (Hard-heavy)	0.6324	0.2847	0.3926
2:4:6 (Easy-heavy)	0.6270	0.2822	0.3893

The results confirm that our 6:4:2 ratio achieves the highest F1 (0.4027). Notably, the Uniform (4:4:4) and Easy-heavy (2:4:6) ratios converge to identical performance (F1 = 0.3893, P = 0.6270, R = 0.2822), revealing a performance floor: once hard-negative exposure falls below a critical proportion, the model's discriminative capacity saturates at the same baseline regardless of easy-negative volume. The Hard-heavy (8:2:2) ratio recovers partially (F1 = 0.3926) but remains below our chosen ratio, suggesting that excessive hard-negative exposure without sufficient gradient-stabilizing easy examples introduces training variance. These results indicate that the 6:4:2 ratio occupies a narrow effective range between insufficient and excessive hard-negative pressure.

5.3 Evaluating Truncation Risk in Concatenated Retrieval

A common baseline optimization in document retrieval involves concatenating multiple highly-ranked candidates into a single cross-encoder prompt to reduce inference latency. To evaluate the feasibility of this approach within the COLIEE Task 2 framework, we profiled the exact token distributions of our dataset using the `legal-bert-base-uncased`[3] tokenizer.

Our analysis revealed an average query fragment length of 44.6 tokens and an average candidate paragraph length of 140.1 tokens, with the 90th percentile of candidates reaching 261.0 tokens. Given

LegalBERT's strict 512-token limit (which includes 3 overhead special tokens), the average available budget for candidate evaluation is strictly 464.4 tokens. Even when evaluated independently (decoupled), 2.52% of exceptionally long paragraphs naturally exceed this boundary, triggering strict Truncation Damage (TDI).

Attempting to concatenate merely the top three wRRF candidates ($k = 3$) produces an average sequence length of 467.9 tokens ($3 + 44.6 + [140.1 \times 3]$), guaranteeing context overflow in the average case. If any candidate within that concatenated sequence falls into the 90th percentile (261.0 tokens), the sequence catastrophically overflows, actively deleting the required ground-truth evidence prior to inference. Therefore, processing all $k = 80$ candidates completely independently is structurally mandatory to preserve semantic integrity, justifying the linear $O(N)$ computational overhead.

5.4 Generative vs. Discriminative Prompting

To isolate the efficacy of our architecture's discriminative boundary detection, we ablated our pipeline against a generative Small Language Model (SLM) baseline utilizing Qwen2.5-7B-Instruct (4-bit quantized). We established a two-stage pipeline: retrieving the top-4 candidates via BGE-m3, and prompting the SLM to independently verify strict legal entailment.

We note that the SLM baseline operates on a reduced candidate set (top-4 from BGE-m3) compared to our primary pipeline (top-80 from wRRF), reflecting the computational constraints of autoregressive inference. While this introduces an asymmetry in retrieval quality, our primary claim is that discriminative boundary detection outperforms generative reasoning at the per-candidate level, as evidenced by the precision gap between our full cross-encoder system (0.6721; Table 3) and the best generative strategy (0.5752, CoT). A controlled comparison at equal retrieval depth is left for future work.

Table 6: Ablation of Qwen2.5-7B Prompting Strategies (Top-4 Candidates). CoT achieves the highest precision among generative methods, but all remain strictly inferior to the cross-encoder.

Prompting Strategy	Precision	Recall	F1-Score
Zero-Shot	0.5273	0.1973	0.2871
Few-Shot (Standard)	0.5038	**0.2245**	0.3106
Chain-of-Thought (CoT)	**0.5752**	0.2211	**0.3194**
Legal IRAC Framework	0.5366	**0.2245**	0.3165

We evaluated four distinct prompting paradigms, detailed in Table 6. Standard Chain-of-Thought (CoT) prompting achieved the highest overall generative performance (F1: 0.3194, Precision: 0.5752) by forcing explicit logical reasoning. Interestingly, the domain-specific IRAC (Issue, Rule, Application, Conclusion) framework tied for the highest recall (0.2245) but suffered a slight drop in precision. This suggests that while formal legal frameworks aid in capturing potential matches, they occasionally overcomplicate straightforward entailments when interacting with the Lexical Trap, leading to false positives.

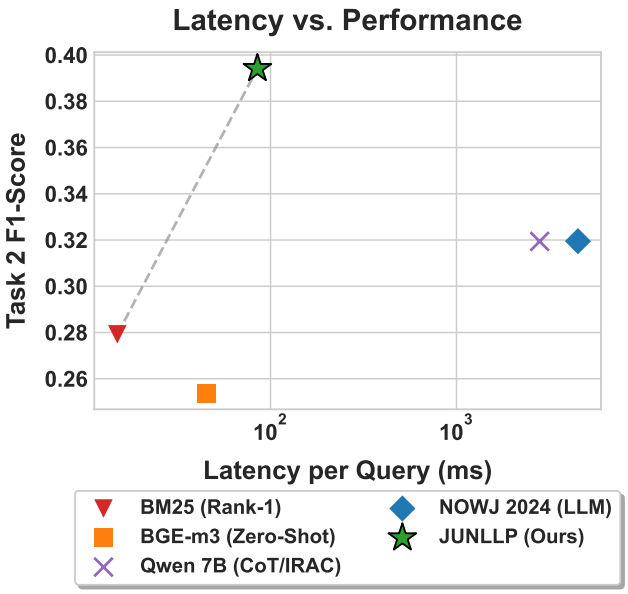


Figure 4: Latency vs. Performance Pareto Frontier. The plot visually confirms that the Decoupled Architecture operates at an optimal intersection of computational efficiency and entailment precision.

Crucially, none of these generative methods reach the precision of our Context-Safe PMA cross-encoder pipeline (0.6721; Table 3). To mathematically quantify this failure, we adapted our Lexical Trap Divergence (LTD) metric for the generative SLM. By extracting the raw token log-probabilities for the model's final entailment classification token (e.g., "YES"), we isolate the generative semantic confidence, S_{llm}. The adapted generative LTD is formulated as:

$$LTD_{slm}(q,p) = S_{lex}(q,p) \cdot \log\left(\frac{S_{lex}(q,p)}{S_{llm}(q,p) + \epsilon}\right) \tag{15}$$

Our empirical analysis of LTD_{slm} revealed a clear difference in behavior. When the SLM successfully resisted a Lexical Trap (True Negatives), it achieved an average LTD of 1.6638, decoupling its semantic reasoning from the baseline keyword overlap. However, when the model hallucinated the entailments in the boilerplate statutes (False Positives), mirroring retrieval failures observed in broader LLM search agents [23], the average LTD plunged to -0.2696. This negative divergence provides empirical evidence of "Lexical Collapse"—the point at which the generative model abandons deep semantic reasoning, and its confidence artificially spikes to match the shallow lexical overlap of the Hard Negative. By replacing this vulnerable generative reasoning with a strictly bounded discriminative hyperplane, our architecture achieves superior entailment mapping.

5.5 Context Concatenation vs. Decoupling

We evaluated a 'coupled' variant of our architecture, where the base case context (C_{base}) from Stage 1 was retained and prepended to the query during Stage 2 cross-encoding. As predicted by our Truncation Damage Index (TDI) framework, this resulted in a sharp

degradation in performance, plunging the F1-score to 0.2335 (Precision: 0.4600, Recall: 0.1565).

The prepended context consumed the majority of the 512-token transformer window. Furthermore, even when tokens were not explicitly truncated, the attention weights suffered from the well-documented "Lost in the Middle" phenomenon [13], where standard transformers fail to utilize vital information buried deep within extended contexts. This ablation empirically supports the need for our 'Decoupling' mechanism.

5.6 Thresholding Dynamics: Precision–Recall Trade-off Analysis

Standard classification pipelines default to thresholding based on a static confidence value (e.g., $\sigma > 0.5$) or a naive *argmax*. To demonstrate why this fails in legal entailment, we tracked the predictive variance across the test set candidates.

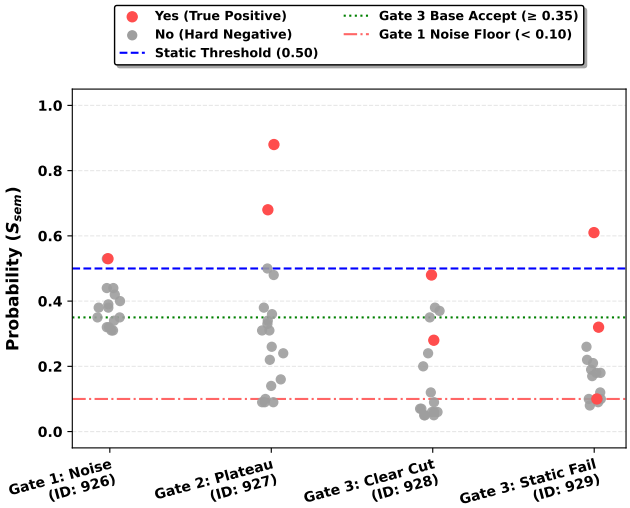


Figure 5: Analysis of predictive variance. The plot illustrates how standard static thresholding leads to false-positive spikes, whereas our 3-Gate Max-Gap algorithm cuts off retrieval at the mathematical plateau.

As shown in Figure 5 and Table 7, the core limitation of static thresholds is dataset dependence: the optimal τ must be grid-searched per corpus, offering no guarantee of generalization to unseen distributions. The best static threshold ($\tau = 0.50$, micro-F1: 0.3902) is only discoverable in hindsight. Our 3-Gate Max-Gap algorithm, without any corpus-level tuning, achieves a comparable micro-F1 of 0.3865 by dynamically adapting its acceptance boundary to each query's local probability landscape — closing within 0.004 of the grid-searched optimum while remaining fully dataset-agnostic. We acknowledge that validating this generalization claim across multiple legal corpora remains a future work.

5.7 Qualitative Human Error Analysis

Since COLIEE Task 2 ground truth is derived from historical judicial citations, we conducted a qualitative human expert evaluation of our pipeline's False Positives rather than a quantitative human baseline. A manual review of the 3-Gate Max-Gap rejections revealed two primary failure modes:

(1) **Implicit Jurisprudential Shifts:** In ∼40% of errors, the entailing paragraph relied on unstated common law knowledge not explicitly present in the text, requiring external legal training to connect the concepts.
(2) **Plausible Alternatives:** In ∼60% of false positives, our model selected a paragraph providing a logically sound entailment, but not the *specific* paragraph the original judge chose to cite. Human reviewers noted these "errors" were often legally defensible, aligning with established findings that false positives in legal entailment frequently represent plausible alternative jurisprudential logic [24, 25].

5.8 Limitations

We acknowledge several limitations of our experimental comparison. More recent architectures such as ColBERTv2 [22], Longformer [2], and domain-adapted long-context encoders represent important alternative approaches to legal entailment. Our constrained compute environment (a single NVIDIA Tesla T4 GPU in a cloud setting) precluded evaluation of these models within the competition timeline. Additionally, the generative baseline (4-bit quantized Qwen2.5-7B-Instruct) represents a lower bound on current LLM capabilities; larger or more capable generative models may narrow the precision gap we report. Future work should investigate whether long-context encoders can match the precision gains of our decoupled approach without the computational overhead of candidate-level cross-encoding.

We note that the official COLIEE 2026 Task 2 leaderboard includes several concurrent systems achieving competitive or superior F1 scores. A detailed architectural comparison with these systems is beyond the scope of this paper, as their system descriptions were not available at the time of writing this paper. We defer such cross-system analysis to future post-workshop publications.

6 Conclusion

Legal case entailment remains one of the most notoriously difficult tasks in domain-specific natural language processing. The dense, boilerplate-heavy nature of judicial opinions naturally confounds standard retrieval systems, creating Lexical Traps that obscure true semantic causality. In this paper, we introduced a highly specialized, 3-stage Decoupled Architecture designed specifically to untangle this complexity for COLIEE Task 2. By explicitly discarding surrounding context to prevent truncation damage (TDI), and routing the retrieved candidates through a Focal Loss-optimized PMA cross-encoder and a deterministic 3-Gate variance reduction algorithm, we established a high-precision logical boundary that actively resists shallow vocabulary overlap.

Our empirical ablation suggests that parameter scaling alone is not sufficient for strict logical entailment. On the COLIEE 2026 Task 2 test set, our fine-tuned LegalBERT cross-encoder outperforms a 7B generative model by +0.10 precision, indicating that discriminative boundary detection may be better suited to high-precision legal entailment than autoregressive reasoning under comparable computational constraints. Additionally, our decoupled cross-encoder

Table 7: Threshold grid search on the validation fold (925 queries). All configurations use the same single-seed LegalBERT + PMA + Focal checkpoint. $F1_{micro}$ is the official COLIEE ranking metric; $F1_{macro}$ measures per-query consistency.

Strategy	τ	Precision	Recall	$F1_{micro}$	$F1_{macro}$
Static (0.10)	0.10	0.1166	0.4331	0.1837	0.2598
Static (0.20)	0.20	0.2551	0.3674	0.3011	0.4072
Static (0.30)	0.30	0.3658	0.3382	0.3515	0.4789
Static (0.40)	0.40	0.4662	0.3187	0.3786	0.5289
Static (0.50)	0.50	0.5292	0.3090	**0.3902**	0.5560
Static (0.60)	0.60	0.5545	0.2847	0.3762	0.5427
Static (0.70)	0.70	0.5876	0.2774	0.3769	0.5535
Top-1 (Argmax)	—	0.6000	0.2701	0.3725	0.5544
3-Gate Max-Gap (Ours)	**dyn.**	**0.5228**	**0.3066**	0.3865	**0.5579**

operates at a fraction of the computational latency required for autoregressive reasoning chains.

Looking forward, we believe the diagnostic frameworks introduced in this study may have applications beyond legal retrieval. We hypothesize that the LTD framework may have potential applications beyond legal retrieval as a diagnostic signal for evaluating the susceptibility of generative models to lexical collapse across other high-stakes, multi-document domains, such as medical diagnostics, financial auditing, and regulatory compliance. By providing a framework for detecting when a model's confidence correlates with shallow keyword overlap rather than deep semantic reasoning, this diagnostic may contribute to building more robust AI systems in domains where precision is paramount.

While our gating algorithm operates at a more conservative recall point, we argue this reflects the correct inductive bias for legal applications where false-positive citations carry significant downstream risk.

References

[1] Farid Ariai and Gianluca Demartini. 2025. Natural Language Processing for the Legal Domain: A Survey of Tasks, Datasets, Models, and Challenges. *arXiv preprint arXiv:2410.21306* (2025).

[2] Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The Long-Document Transformer. *arXiv preprint arXiv:2004.05150* (2020).

[3] Ilias Chalkidis, Manos Fergadiotis, Prodromos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos. 2020. LEGAL-BERT: The Muppets straight out of Law School. In *Findings of the Association for Computational Linguistics: EMNLP 2020*. 2898–2904.

[4] Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2024. BGE M3-Embedding: Multi-Lingual, Multi-Functionality, Multi-Granularity Text Embeddings Through Self-Knowledge Distillation. *arXiv preprint arXiv:2402.03216* (2024).

[5] Gordon V. Cormack, Charles L. A. Clarke, and Stefan Buettcher. 2009. Reciprocal Rank Fusion outperforms Condorcet and individual Rank Learning Methods. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*. 758–759.

[6] Yoshinobu Kano, Randy Goebel, Mi-Young Kim, Calum Kwan, Ken Satoh, Hiroaki Yamada, and Masaharu Yoshioka. 2026. An Overview of the COLIEE 2026 Competition: Legal Case Law and Statute Law Information Retrieval and Entailment. In *Proceedings of the 21st International Conference on Artificial Intelligence and Law (ICAIL 2026)*.

[7] Haruka Kiyohara, Rayhan Khanna, and Thorsten Joachims. [n. d.]. Off-Policy Learning for Diversity-aware Candidate Retrieval in Two-stage Decisions. In *ICML 2025 Workshop on Scaling Up Intervention Models*.

[8] Solomon Kullback and Richard A. Leibler. 1951. On Information and Sufficiency. *The Annals of Mathematical Statistics* 22, 1 (1951), 79–86.

[9] Juho Lee, Yoonho Lee, Jungtaek Kim, Adam Kosiorek, Seungjin Choi, and Yee Whye Teh. 2019. Set Transformer: A Framework for Attention over Sets. In *International Conference on Machine Learning (ICML)*. 3744–3753.

[10] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. In *Advances in Neural Information Processing Systems*, Vol. 33. 9459–9474.

[11] Xiaotong Li et al. 2025. KaLM-Embedding-V2: Superior Training Techniques and Data Inspire A Versatile Embedding Model. *arXiv preprint arXiv:2506.20923* (2025).

[12] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. 2017. Focal Loss for Dense Object Detection. In *Proceedings of the IEEE international conference on computer vision*. 2980–2988.

[13] Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024. Lost in the Middle: How Language Models Use Long Contexts. *Transactions of the Association for Computational Linguistics* 12 (2024), 157–173.

[14] Ilya Loshchilov and Frank Hutter. 2017. Decoupled Weight Decay Regularization. *arXiv preprint arXiv:1711.05101* (2017).

[15] R. Thomas McCoy, Ellie Pavlick, and Tal Linzen. 2019. Right for the Wrong Reasons: Diagnosing Syntactic Heuristics in Natural Language Inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*. 3428–3448.

[16] Venkatesh Mishra, Bimsara Pathiraja, Mihir Parmar, Sat Chidananda, Jayanth Srinivasa, Gaowen Liu, Ali Payani, and Chitta Baral. 2025. Investigating the Shortcomings of LLMs in Step-by-Step Legal Reasoning. In *Findings of the Association for Computational Linguistics: NAACL 2025*. 7810–7841.

[17] Hoang-Trung Nguyen, Tan-Minh Nguyen, Thuong-Hieu Ngo, Ha-Thanh Nguyen, and Thi-Hai-Yen Vuong. 2026. LexEntail: A Multi-Stage Reranking Pipeline with LLM-Based Reasoning for Legal Case Entailment. *The Review of Socionetwork Strategies* (2026), 1–39.

[18] Tan-Minh Nguyen, Hai-Long Nguyen, Dieu-Quynh Nguyen, and Ha-Thanh Nguyen. 2024. NOWJ@COLIEE 2024: Leveraging Advanced Deep Learning Techniques for Efficient and Effective Legal Information Processing. In *Proceedings of the Juris-Informatics Workshop (JURISIN)*.

[19] Rodrigo Nogueira and Kyunghyun Cho. 2019. Passage Re-ranking with BERT. *arXiv preprint arXiv:1901.04085* (2019).

[20] Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. 3982–3992.

[21] Stephen Robertson, Hugo Zaragoza, et al. 2009. The Probabilistic Relevance Framework: BM25 and Beyond. *Foundations and Trends in Information Retrieval* 3, 4 (2009), 333–389.

[22] Keshav Santhanam, Omar Khattab, Jon Saad-Falcon, Christopher Potts, and Matei Zaharia. 2022. ColBERTv2: Effective and Efficient Retrieval via Lightweight Late Interaction. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL)*. 3715–3734.

[23] Weiwei Sun, Lingyong Yan, Xinyu Ma, Shuaiqiang Wang, Pengjie Ren, Zhumin Chen, Dawei Yin, and Zhaochun Ren. 2023. Is ChatGPT Good at Search? Investigating Large Language Models as Re-Ranking Agents. *arXiv preprint arXiv:2304.09542* (2023).

[24] Li Zhang, Jaromir Savelka, and Kevin Ashley. 2025. Do LLMs Truly Understand When a Precedent Is Overruled? *arXiv preprint arXiv:2510.20941* (2025).

[25] Lucia Zheng, Neel Guha, Brandon R. Anderson, Peter Henderson, and Daniel E. Ho. 2021. When Does Pretraining Help? Assessing Self-Supervised Learning for Law and the CaseHOLD Dataset. In *Proceedings of the 18th International Conference on Artificial Intelligence and Law (ICAIL)*. 159–168.

SACH - Sourcing Accurate Cases without Hallucinations at scale

Apoorva Gupta
apgupta@bu.edu
Faculty of Computing and Data Sciences
Boston, Massachusetts, USA

Abstract

Legal case law retrieval and entailment are common problems that legal practitioners face on a daily basis. Classical tools that use boolean search operations are slow and provide limited scope in terms of returned search results. However, advancements in the application of transformer models have enabled the creation of state-of-the-art embedding models which facilitate different kinds of encoding, such as bi-encoding and cross-encoding. These models can enable faster and more relevant case law retrieval and entailment at scale. This paper, from COLIEE 2026, investigates two such tasks: Legal case law retrieval (Task 1) and Legal entailment (Task 2). These approaches demonstrate the success of combining lexical with semantic re-ranking techniques for Task 1 and cross-encoders for Task 2. We are ranked 13 in both Task 1 and Task 2 out of 22 and 14 teams, respectively.

CCS Concepts

• **Applied Computing** → **Law**; • **Information Systems** → **Semantic re-ranking**.

Keywords

Vector Databases, Semantic Chunking, BM25, Cross encoders

ACM Reference Format:
Apoorva Gupta. 2026. SACH - Sourcing Accurate Cases without Hallucinations at scale. In *Proceedings of COLIEE 2026*. ACM, New York, NY, USA, 6 pages.

1 Introduction

The advent of attention models promises improved efficiency in legal case law retrieval due to the enhanced capabilities in processing unstructured text data. Case law retrieval is beset by problems of both scale and breadth. The sheer scale of digitized case laws, both within the United States and globally, continues to grow. Furthermore, as legal precedents evolve over time, it becomes challenging to extract contextual information that addresses the relevant precedents.

To address these challenges, we have developed a case law search model, SACH, Sourcing Accurate Citations without Hallucinations. It is well known[1] that large language models (LLMs) can produce hallucinations. This is a phenomenon where the LLM produces factually incorrect information. The incorrect information resulting from LLM hallucinations is extremely damaging for case law retrieval.

Consequently, any approach that addresses this problem must work at scale. In this paper, lexical model based approaches [5] are considered for the following reasons.

- They are low cost. The considered models do not require additional training. Importantly, the main cost is the semantic chunking of all the case law files. However, this is only a one time cost.
- The performance can be finetuned via hyperparameter optimization. Hyperparameter tuning offered a variety of solutions which enabled faster execution and improved results.

The overall idea for Task 1 was to extract a relevant subset of cases by using lexical methods, such as BM25[18], TF-IDF [2], on a large set of case laws and then use an improvised version of semantic re-ranking to retrieve the best set of results. In between these processes is our SACH model, developed by the author. It is a bi-encoder based model [20]. These models process queries and documents independently into fixed length vector embeddings. They enable faster retrieval at scale via pre-computed document indexes. Semantic re-ranking [19] is important because legal case law retrieval requires searching for a few results in a massive corpus of cases. With efficient re-ranking, we can guarantee better results in top-n retrievals, which is crucial for saving time and effort of lawyers.

For Task 2, we used a cross-encoder based model. These models use the given document (paragraphs) and the query (entailed fragment) to determine their relevance. They are highly accurate due to their token level matching. Every query token attends directly with the document token. This is in contrast to using isolated vector embeddings that are multiplied with each other to compute similarity based on dot products. This also boosts contextual matching via phrase level understanding and semantic alignment.

This approach was developed based on what tokens influence the predictions and how query and document level matching is achieved via attention patterns. Therefore, we used cross encoders [10]. However, these models do not generate sentence level embeddings and thus perform better than bi-encoder models in tasks where semantic similarity between sentence pairs is required [15]

Therefore, the actual question that we tried to answer via our work with the help of two tasks (Task 1 and Task 2) [8] is to explore the possibility of developing a case retrieval/entailment models without any training on the labeled data at scale. Hence, we aim to curate a set of semantic chunking operations, semantic re-rankers and hyperparameter optimization to retrieve better results. To increase the chances of having these applications at scale to enable context dense retrieval.

This paper is structured as follows. In section 2, the tasks that were completed for the Competition of Legal Information Extraction and Entailment (COLIEE) 2026 in two domains, Case Law Retrieval (Task 1) and Case Law Entailment (Task 2), are discussed. In section 3, a brief data analysis of the dataset provided to execute these tasks is provided. In section 4, the metrics used to evaluate the results are presented. In section 5, we present the Methodology. In section 6 we discuss the results in detail. Concluding remarks are offered in section 7.

2 Task Description

2.1 Task 1 - Case law retrieval for decision support

For a given query case (Q_t) the algorithm should retrieve the top-n cases that complement some or all of the legal precedents cited within Q_t. Additionally, system must ensure temporal consistency; all retrieved results must predate Q_t. This is an important chronological constraint to ensure legal reality, as a case cannot refer a future litigation that had not yet occurred at time of its filing [3].

The mathematical formulation of the problem is as follows. Given a query case Q_t at time t and a comprehensive case document set $C = \{C_{1,t_1}, C_{2,t_2}, \ldots, C_{n,t_n}\}$, where C_{i,t_i} denotes the i-th case record and its associated timestamp t_i, the objective is to identify the Relevant Document Set (R).

This set is defined as the collection of top-n cases that complement the legal precedents in Q_t while strictly enforcing the temporal constraint $t_i < t$. Formally, the retrieval is bounded such that:

$$R = \{C_{i,t_i} \in C \mid t_i < tn\}. \tag{1}$$

This ensures that all output case laws were established prior to the query case, maintaining the chronological integrity of the legal citation network.

2.2 Task 2 - Legal Case Entailment

For a given base case (B) and entailed fragment (E) we are supposed to find paragraphs (P) from a list of paragraphs that are related to that entailed fragment (E) from the base case [4].

The task can be formally defined as follows. Given a entailed fragment E, and a set of candidate paragraphs $P = \{p_1, p_2, \ldots, p_m\}$, the goal is to compute a relevance score for each paragraph and entailed text pair. We define a scoring function $S_i = f(E, p_i)$, where f represents any model capable of processing the query components and candidate paragraphs to output a similarity score using full self attention [10].

Our objective is to identify the set of relevant paragraphs $\hat{P}$, defined as:

$$\hat{P} = \{p_i \in P \mid f(E, p_i) \geq \tau\}, \tag{2}$$

where the threshold τ is determined by the k-th percentile of the score distribution $\{S_1, S_2, \ldots, S_m\}$. Specifically, we define τ such that $P(S \leq \tau) = \frac{k}{100}$, evaluating $k \in \{85, 90, 95, 98, 99\}$ to identify the optimal cutoff for high-precision retrieval.

3 Dataset Description

For Task 1, the dataset contains 7708 case laws from the Federal Court of Canada in text (.txt) format with a corresponding json (.json) file. The keys of this file are the query cases and the values are a list of candidate cases which are related to the given query case. A notable characteristic of this data is queries and cases are almost of the same word length. In particular, the mean query length is 5027 words and the mean candidate case length is 4814. This is shown by red dotted and red dashed line, respectively, in Figure 1. The median query length is 4083 and the median candidate case length is 3565 words. This indicates all cases (query cases and candidate cases) are taken from the same data source.

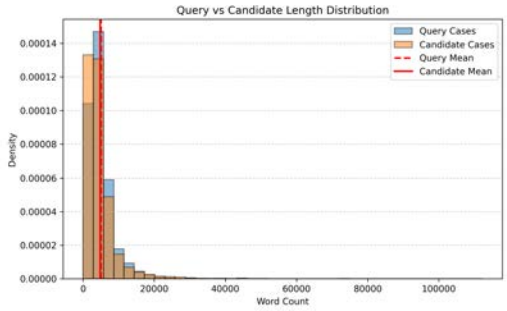


Figure 1: Query cases and Candidate Cases length distribution

The probability density plot of these word lengths is shown in Figure 1. Importantly, this illustrates that we need to extract similar chunks with higher similarity to guarantee similar context while retrieving case laws. This is due to the fact that two case laws might not be exactly similar to each other. As an example, a part of one case law paragraph might hold higher relevance with the different paragraphs of the other case law. Therefore, a critical step in our retrieval pipeline involves computing the dot-product between query embeddings and chunked case law vectors. This process allows for the calculation of chunk-level cosine similarity to identify the most relevant legal segments. However, we observed significant anisotropy [12] within the embedding space, a phenomenon where vectors are non-uniformly distributed and occupy a narrow cone. This led to an unrealistic high similarity between unrelated segments as well, resulting in a compressed range of scores. Consequently, distinguishing between relevant and irrelevant case law portions became numerically difficult, directing towards devising a more robust thresholding strategy.

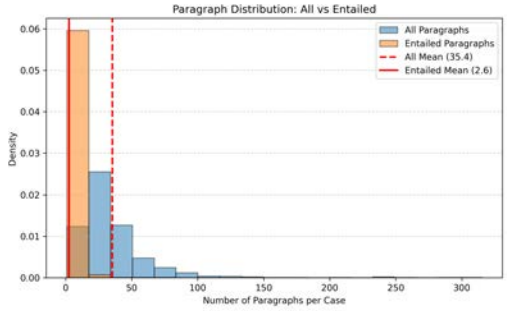


Figure 2: Number of paragraphs available per entailed text

For Task 2, we are provided with 925 different folders. Each folder contains a base case and an entailed fragment that is related to that base case. Since we need to retrieve the number of entailed paragraphs, we visualized the probability density plot of the total available paragraphs for each entailed fragment and the actual number of entailed paragraphs, see Figure 2. For a given entailed fragment, there are on average 3 paragraphs that need to be retrieved from a large pool of approximately 36 paragraphs available per entailed paragraph to get actual entailment results.

4 Evaluation Metrics

To evaluate both tasks, recall, precision, and the F1-score are used as metrics.

For Task 1, these metrics are defined as follows. Let $C(Q)$ and $\mathcal{R}(Q)$ denote a set of candidate cases and a set of retrieved cases, respectively.

- The precision is how many actual candidate cases are in the set of retrieved cases, i.e.,

$$\text{Precision}(Q) = \frac{|\mathcal{R}(Q) \cap C(Q)|}{|\mathcal{R}(Q)|}.$$

- The recall is defined as the number of correctly retrieved cases out of all true candidate cases, i.e.,

$$\text{Recall}(Q) = \frac{|\mathcal{R}(Q) \cap C(Q)|}{|C(Q)|}.$$

- The F-1 score is the harmonic mean of the precision and recall, i.e.,

$$F_1(Q) = \frac{2 \cdot \text{Precision}(Q) \cdot \text{Recall}(Q)}{\text{Precision}(Q) + \text{Recall}(Q)}.$$

The precision, recall, and F1-score are similarly defined for Task 2. In this case, let $C'(P)$ and $\mathcal{R}'(P)$ denote the set of correct entailed paragraphs and the set of retrieved paragraphs, respectively.

- The precision is how many correctly entailed paragraphs are in the set of retrieved paragraphs, i.e.,

$$\text{Precision}(P) = \frac{|\mathcal{R}'(P) \cap C'(P)|}{|\mathcal{R}'(P)|}.$$

- The recall is the number of correct entailed paragraphs out of all retrieved paragraphs, i.e.,

$$\text{Recall}(P) = \frac{|\mathcal{R}'(P) \cap C'(P)|}{|C'(P)|}.$$

- The F-1 Score is the harmonic mean of precision and recall:

$$F_1(P) = \frac{2 \cdot \text{Precision}(P) \cdot \text{Recall}(P)}{\text{Precision}(P) + \text{Recall}(P)}.$$

5 Methodology

5.1 Task 1

The adopted approach to solve Task 1 is inspired by the case law retrieval method SACH, see section 1. In this approach, a user-provided legal query is matched to relevant case laws. The approach provides the top-10 results that are most relevant to answering the legal query. In particular, the specific sections in the case laws that are relevant to the legal query are highlighted. SACH was created using approximately 2 million case law documents coming from the US judicial system. These documents contain cases from the states of Massachusetts, New York, Maine, Pennsylvania, Louisiana, Washington, Florida, and Vermont. Each document is semantically chunked and then converted into vector embeddings using baai-bge [21] embeddings.

We took a random set of 100 case laws from these 2 million cases. Then generated 100 random queries one for each case law using Gemini 3[14] using the following prompt

"You are a legal professional, have read this case law for some time, and then provide some legal queries for which this case law can be a very good answer and can be used as a precedent for argumentation in front of the court. Remember, humanize these queries and make it sound like an actual lawyer or legal practitioner will come across in their research work."

By this measure, the definite ground truth for evaluation was established as we already know both the case law and a query that can be solved by that case law.

Then, we asked these queries to our semantically chunked vector database algorithm SACH independently. We found that exact same case law in 23 out of 100 cases. We repeated the same experiment with LexisNexis and observed a 40% better performance than LexisNexis, see Table 1. Moreover, our approach also returns the relevant portions and paragraphs related to these queries.

Table 1: Comparative Retrieval Performance:

Metric	Case Found	Case Not Found	No Output
SACH (Our Model)	23	77	0
LexisNexis	16	48	37

Using SACH, our methodology for Task 1 is as follows.

- **Data Preprocessing** - To clean the data, a custom function was created to return a new field called `clean_english_text` in all the documents. This function removed all the non ASCII characters, white spaces, and the tags like `<FRAGMENT_SUPPRESSED>` which are substituted for the case citation in the original case law. Since the case laws are from Canada, they also conatined French words. These words were removed using zipf's law [13]. This approach uses a log-scaled frequency score as a heuristic for distinguishing tokens based on their language. We removed all words where the log-frequency of French occurrence exceeded 3, while the log-frequency of English occurrence was less than 2. In the same process, we also extracted the case year using the maximum year in the given case law. This ensured that for a given query case we do not end up ranking candidate cases held after the year of query case.
- **Token extraction** - The obtained `clean_english_text` for each document is then used for extracting words. This token extraction facilitates our initial search layer of keyword matching that uses BM25 algorithm. We used both term frequency-inverse document frequency (TF-IDF) and Kullback-Leibler Divergence for Informativeness (KLI) [5]

Table 2: Selection between TF-IDF and KLI Tokenisaion:

Method	KLI	TF-IDF
Recall	0.4256	0.4155
Precision	0.0300	0.0294

for extracting the top 40 tokens. We tried 2 variants of initial BM25 retrieval with 40 tokens using both TF-IDF and KLI. Both precision and recall values for tokens obtained from KLI was higher. See Table 2. Since, KLI based 40 tokens gave higher values we selected them for downstream filtration.

- **Case Law Retrieval** - We retrieved a large set of initial case laws using the BM25 algorithm[18]. BM25 is a probablistic information retrieval [17] algorithm that is widely used by search engines to estimate the relevance of a document to a given search query. For our 2 layer search pipeline, this algorithm worked well to curate the initial pool of relevant documents. This step is further optimized for how strongly the term frequency impacts the eventual BM25 score and also to what extent this term frequency should matter to prevent its indefinite affect on the BM25 score computation. Moreover, to mitigate potential bias due to disproportionate document lengths we optimized for the impact of document length on the BM25 score as well. Both these steps ensure a balanced calibration of lexical matching is driven by substantive keyword density rather than document volume, effectively preventing the selection of irrelevant, verbose cases.
- **Semantic Chunking** - The large candidate set of retrieved cases undergoes subsequent filtering for a small subset not only to increase precision but also to make results more humane as the initial pool of results is still not in usable form just by sheer volume. To achieve this, we performed semantic chunking using langchain[11]. For document retrieval tasks, semantic chunking strikes the right balance between sentence level chunking and paragraph level chunking. As the former makes the retrieval too slow due to large number of vectors and latter dilutes the context due to topic shifting because 2-3 relevant pieces of information in a paragraph get averaged out in a single vector. It breaks documents into sentences at page level and then tries to find the similarity with the sentences in the proximity using cosine similarity. If it is higher than the breakpoint threshold computed via standard deviation or percentiles two sentences become one chunk. This technique keeps localized context within a chunk high. However, it is computationally expensive because of the iterations involved in creating each chunk. We have used the BAAI/bge-base-en-v1.5[21] model to create word embeddings because it facilitates dense retrieval via unsupervised learning[9]. It is also sensitive to localized semantic shifts so splits are more likely once contextual drift is identified. This makes each chunk highly coherent and semantically dense. These chunks are then fed to a vector database training module using the FAISS[6] library. Henceforth, both the vector database and metadata of the semantic chunks get stored in pickle format. To make this process fast, we already chunked all the documents semantically and created a vector database of them. This is the SACH approach developed by the author as discussed in subsection 1.
- **Semantic Re-ranking** - As mentioned earlier matching documents of almost similar lengths means large portions of dissimilarity followed by small regions that make a pair of documents relevant to each other. Our re-ranking technique is based on gap pruning. For each chunk associated with a given query case we compute the top-k most relevant chunks and thus the candidate case. Then candidate documents are scored by counting how many times their chunks appear across all query chunk retrievals to get a frequency-based signal. This signal tracks the best (lowest) rank position at which each candidate appears. Finally, candidates get sorted by high frequency, better rank. We then apply a gap-based pruning to keep only the most relevant results. It is a threshold cutoff method in which the difference between two consecutive re-ranking scores is calculated. It keeps on going unless the score difference becomes greater than the defined threshold and the all the case laws within the defined threshold difference makes into the selected set of candidate cases.

The methodology for Task 2 is based on dynamic requirements for legal entailment at scale. While case retrieval (Task 1) allows for the bulk pre-computation of vector embeddings at rest, legal entailment (Task 2) requires the real-time processing of entailed fragments because any static pre-calculation for entailment only accounts for a suboptimal approximation. To evaluate system capabilities under these constraints, the labeled dataset was treated as a dynamic input stream. We therefore implemented a cross-encoder model utilizing a percentile-based threshold for high fidelity retrieval.

- **Data Loading and Preparation** - The entailed fragment is paired with every document in the candidate paragraph list to form unique query-candidate tuples. These pairs serve as the input for the re-ranking model, which computes a relevance score for each combination separately.
- **Cross encoder Scoring** - We used ms-marco-MiniLM-L-6-v2 [16] for cross encoding with a percentile threshold computed for 5 different threshold values: 85, 90, 95, 98, and 99.
- **Percentile Evaluation and Retrieval** - This percentile is a score computed after getting all the scores for a given entailed fragment paragraph cross set. If a particular cutoff score falls below the threshold percentile, that paragraph will be ignored and all paragraphs above the threshold will be selected for retrieval.

6 Results

6.1 Task 1

We submitted 2 results, Run 1 (sach_task1_run1) and Run 2 (sach_task1_run2). Both gave a recall of 0.2200 and 0.2231 respectively and we are ranked 13 out of 25 teams. To obtain better results hyperparameter optimization was done on 4 factors.

- **BM25_TOP_K** – How many results should be retrieved from the initial BM25 search layer.
- **RERANKER_TOP_K** – The maximum number of documents that can be kept after semantic re-ranking.
- **GAP_THRESHOLD** – The minimum score difference between consecutive candidates for gap pruning.
- **PER_SENTENCE_K** – The number of cases retrieved per query chunk from the vector database index during re-ranking.

The hyperparameter optimization was run for the following values:

- **BM25_TOP_K** – 40, 50, 100
- **RERANKER_TOP_K** – 5, 10, 15
- **GAP_THRESHOLD** – 15, 20, 25
- **PER_SENTENCE_K** – 5, 10

Hyperparameter optimization was performed for all the 54 combinations and the top 5 are shown below in Table 3.

6.2 Task 2

We submitted 1 result (sach_task2_run1), and got a recall of 0.2167. We are ranked 13 out of 14 teams. The proposed approach used a cross-encoder model and percentile as a hyperparameter. However, for two different cross encoders BAAI/bge-reranker-base[21] and jina-reranker-v2-base-multilingual[7] the code ran very slowly for single pair. To obtain better results hyperparameter optimization was done on a single factor percentile, see Figure 3.

- **PERCENTILE** - The score cutoff used to select only the candidate cases in the top k-th percentile of the cross-encoder's output. We experimented with values of 85, 90, 95, 98, and 99

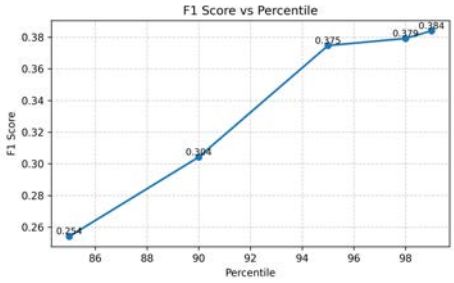


Figure 3: Percentile wise recall on training data

7 Conclusion

In Task 1 the evaluation results performed better than training benchmarks obtained during hyperparameter optimization. This suggests that training-free methods centered around hyperparameter optimization are viable at scale. However, The major challenge that has to be addressed is the problem of anisotropy that prevents semantically similar pairs to be differentiated from the noise due to minimal differences in cosine similarities.

However, similar robustness was not observed in Task 2. While this strategy was fast, the resulting performance trade-off was suboptimal. Our findings indicate that in the context of legal entailment, the F1-score is a more critical metric than raw inference speed, as the precision of legal retrieval carries higher utility than low-latency execution. The discrepancy between our training results and obtained evaluation results shows that the current training approach lacks the diversity required for hyperparameter generalization. Consequently, our reliance on a single percentile-based threshold, while providing a baseline for performance, leaves significant room for optimization by incorporating additional hyperparameters.

8 Acknowledgments

I work as a Learning Facilitator at the Faculty of computing and data sciences at the Boston University. I am thankful to Prof. Scott Ladenheim for his support and advisory role on this project. Furthermore, the author is pleased to acknowledge that the computational work reported on in this paper was performed on the Shared Computing Cluster which is administered by Boston University's Research Computing Services www.bu.edu/tech/support/research/. Lastly, the author thanks the COLIEE Team for access to the data.

References

[1] Aisha Alansari and Hamzah Luqman. 2025. Large Language Models Hallucination: A Comprehensive Survey. *arXiv preprint arXiv:2510.06265* (2025). https://arxiv.org/html/2510.06265v2

[2] Euijin Baek, Jiayi Dai, H. M. Quamran Hasan, Yeji Kim, Housam Khalifa Bashier Babiker, Mi-Young Kim, and Randy Goebel. 2025. From TF-IDF to Instruction-Tuned LLMs: Hybrid Legal Reasoning Systems for COLIEE 2025. In *Proceedings of the COLIEE 2025 Workshop*. ACM, Chicago, USA.

[3] COLIEE Organizers. 2026. COLIEE 2026 Task 1 Corpus. https://coliee.org/COLIEE2026/corpus/task1 Competition on Legal Information Extraction and Entailment (COLIEE 2026), Accessed: 2026-03-30.

[4] COLIEE Organizers. 2026. COLIEE 2026 Task 2 Corpus. https://coliee.org/COLIEE2026/corpus/task2 Competition on Legal Information Extraction and Entailment (COLIEE 2026), Accessed: 2026-03-30.

[5] Rohan Debbarma, Pratik Prawar, Abhijnan Chakraborty, and Srikanta Bedathur. 2023. IITDLI: Legal Case Retrieval Based on Lexical Models. In *Proceedings of the COLIEE 2023 Workshop*. ACM, Braga, Portugal.

[6] Matthijs Douze, Andrey Guzhva, Chengqi Deng, Jeff Johnson, Gregory Szilvasy, Pierre-Emmanuel Mazaré, Maria Lomeli, Lucas Hosseini, and Hervé Jégou. 2024. The Faiss Library. *arXiv preprint arXiv:2401.08281* (2024). doi:10.48550/arXiv.2401.08281

[7] Jina AI. 2024. *Jina Reranker v2: base-multilingual.* Retrieved March 29, 2026 from https://huggingface.co/jinaai/jina-reranker-v2-base-multilingual Model version 2.0.

[8] Yoshinobu Kano, Randy Goebel, Mi-Young Kim, Calum Kwan, Ken Satoh, Hiroaki Yamada, and Masaharu Yoshioka. 2026. An Overview of the COLIEE 2026 Competition: Legal Case Law and Statute Law Information Retrieval and Entailment. In *Proceedings of the 21st International Conference on Artificial Intelligence and Law (ICAIL '26)*. Association for Computing Machinery, New York, NY, USA.

[9] Chaofan Li, Zheng Liu, Shitao Xiao, and Yingxia Shao. 2023. Making Large Language Models A Better Foundation For Dense Retrieval. arXiv:2312.15503 [cs.CL] doi:10.48550/arXiv.2312.15503

[10] Laure Soulier Benjamin Piwowarski Mathias Vast, Basile Van Cooten. 2025. Understanding Matching Mechanisms in Cross-Encoders. *arXiv preprint arXiv:2507.14604* (2025). https://arxiv.org/abs/2507.14604

[11] Vasilios Mavroudis. 2024. LangChain. doi:10.20944/preprints202411.0566.v1

[12] Benoît Sagot Nathan Godey, Éric de la Clergerie. 2024. Anisotropy Is Inherent to Self-Attention in Transformers. *arXiv preprint arXiv:2401.12143* (2024). https://arxiv.org/html/2401.12143v2

[13] Steven T. Piantadosi. 2014. Zipf's Word Frequency Law in Natural Language: A Critical Review and Future Directions. *Psychonomic Bulletin & Review* 21, 5 (2014), 1112–1130. doi:10.3758/s13423-014-0585-6

[14] Sundar Pichai, Demis Hassabis, and Koray Kavukcuoglu. 2025. *A new era of intelligence with Gemini 3.* Google Blog. Retrieved March 29, 2026 from https://blog.google/products-and-platforms/products/gemini/gemini-3/

[15] Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. *arXiv preprint arXiv:1908.10084* (2019). https://arxiv.org/abs/1908.10084

[16] Nils Reimers, Iryna Gurevych, and Sentence-Transformers Community. 2020. Cross-Encoder: ms-marco-MiniLM-L6-v2. *Hugging Face Model Repository* (2020). https://huggingface.co/cross-encoder/ms-marco-MiniLM-L6-v2

Table 3: Evaluation Results for Different Hyperparameter Configurations on training data

bm25_top	reranker_top	gap_threshold	per_sentence	micro_precision	micro_recall	micro_f1
40	10	20	10	0.10019705	0.203369289	0.13425074
40	15	20	10	0.10019705	0.203369289	0.13425074
40	20	20	10	0.10019705	0.203369289	0.13425074
40	10	15	10	0.099685268	0.199612168	0.132967343
40	15	15	10	0.099685268	0.199612168	0.132967343
40	20	15	10	0.099685268	0.199612168	0.132967343

[17] Stephen Robertson and Hugo Zaragoza. 2009. The Probabilistic Relevance Framework: BM25 and Beyond. *Foundations and Trends in Information Retrieval* 3, 4 (2009), 333–389. doi:10.1561/1500000019

[18] Stephen E. Robertson, Steve Walker, Susan Jones, Micheline M. Hancock-Beaulieu, and Mike Gatford. 1995. Okapi at TREC-3. In *TREC-3 Proceedings*. NIST.

[19] Meet Raval Dhvani Upadhyay Tejul Pandit, Sakshi Mahendru. 2025. The Evolution of Reranking Models in Information Retrieval: From Heuristic Methods to Large Language Models. *arXiv preprint arXiv:2512.16236* (2025). https://arxiv.org/html/2512.16236v1

[20] VeloDB. 2025. The Dual Architecture of Semantic Matching: Bi-Encoder vs. Cross-Encoder in IR and RAG. https://www.velodb.io/glossary/bi-encoder-vs-cross-encoder Accessed: 2026-03-29.

[21] Peitian Zhang, Shitao Xiao, Zheng Liu, Zhicheng Dou, and Jian-Yun Nie. 2023. Retrieve Anything To Augment Large Language Models. *arXiv preprint arXiv:2310.07554* (2023). doi:10.48550/arXiv.2310.07554

KIS at COLIEE 2026 Pilot Task: Knowledge Retrieval from Judgment Documents for Legal Judgment Prediction Task

Kazuma Kadowaki
Shizuoka University
Hamamatsu, Shizuoka, Japan
kkadowaki@kanolab.net

Yoshinobu Kano
Shizuoka University
Hamamatsu, Shizuoka, Japan
kano@inf.shizuoka.ac.jp

Abstract

In this paper, we investigate the effectiveness of incorporating information from judgment documents for Legal Judgment Prediction (LJP) in the COLIEE 2026 Pilot Task (LJPJT 2026). While existing approaches primarily rely on claim-level text spans and general-purpose pretrained models, such representations may not sufficiently capture the court's reasoning process. To address this issue, we propose a retrieval-augmented generation (RAG) framework that leverages textual descriptions of the court's reasoning in original judgment documents. We construct three systems that differ in the granularity of the information used to generate auxiliary knowledge ("lessons") and evaluate their impact on prediction performance. Our results suggest that, while manually annotated claim-level data remains useful, lessons generated from raw judgment documents alone can also provide effective signals for prediction. In addition, we observe that the presence of ground-truth labels may influence how the model acquires useful knowledge, potentially leading to a greater focus on prediction outcomes rather than the underlying reasoning process. Despite these findings, our LLM-based approach does not outperform a fine-tuned ModernBERT model, highlighting the importance of fine-tuning for this task. These findings suggest that incorporating richer contextual information from judgment documents, particularly the court's reasoning process, is a promising direction for improving LJP systems.

CCS Concepts

• **Computing methodologies** → **Natural language processing**; • **Applied computing** → *Law*; • **Information systems** → Data mining.

Keywords

COLIEE, legal judgment prediction, gpt-oss, retrieval augmented generation, knowledge retrieval

ACM Reference Format:
Kazuma Kadowaki and Yoshinobu Kano. 2026. KIS at COLIEE 2026 Pilot Task: Knowledge Retrieval from Judgment Documents for Legal Judgment Prediction Task. In *Proceedings of COLIEE 2026 workshop, June 12, 2026, Singapore.* ACM, New York, NY, USA, 8 pages.

COLIEE 2026, Singapore

1 Introduction

Research on Legal Judgment Prediction (LJP) is emerging as a framework for supporting decision-making processes in the legal domain. In the Competition on Legal Information Extraction and Entailment (COLIEE) 2025 held last year, the Legal Judgment Prediction for Japanese Tort cases (LJPJT) 2025 task was introduced as a pilot task [1], and LJPJT 2026 was also held in the subsequent COLIEE 2026 [3].

The LJPJT 2025 and 2026 tasks [9] aim to predict the outcome of legal litigation and consists of two subtasks. The first is Rationale Extraction (RE), which determines whether each claim made by the plaintiff or the defendant, annotated by legal experts as a text span extracted from judgment documents, is accepted by the court. The second is Tort Prediction (TP), which predicts whether tort liability exists as the judgment outcome based on those claims.

In the previous round, LJPJT 2025, participants implemented systems based on BERT-like language models or LLMs trained on large-scale general-purpose data, or on rule-based scoring derived from the training data [1, 2, 4, 5, 8]. However, the training data in the LJPJT task contains only text spans representing the claims of the plaintiff and defendant, along with their corresponding binary labels, and the document collections used for pretraining are also general-purpose. Therefore, we focused on the possibility that such systems cannot sufficiently learn the court's reasoning process.

Accordingly, in this paper, we investigated whether incorporating not only the text spans in the given training dataset but also textual descriptions of the court's reasoning contained in the original judgment documents corresponding to the training data is effective for prediction in the LJPJT task[1]. Specifically, when predicting the two subtasks simultaneously with a model based on gpt-oss [6], we employed a retrieval augmented generation (RAG) mechanism to refer, in a few-shot manner, to several related judgment documents in the training dataset, and compared multiple systems that differ in the granularity of the referenced information. In the COLIEE 2026 Pilot Task formal run, we constructed three systems with different levels of granularity, KIS1, KIS2, and KIS3, and submitted their outputs.

Our experimental results show three findings:

- When providing information about related cases, using the original judgment documents in addition to the text spans provided in the dataset may improve performance.
- When the original judgment documents contain sufficient information corresponding to the correct labels, manually annotated labels may hinder sufficient knowledge acquisition by the model.

[1]Although the court's reasoning in the test data clearly contains information useful for estimating the correct labels, we did not use it.

- Prompt tuning of LLMs alone could not achieve performance superior to fine-tuning of small-scale SLM models.

These results suggest a research direction for future LJP studies, namely how to incorporate the reasoning process of judges that is not captured in the text spans of claims.

The remainder of this paper is organized as follows: Section 2 describes related work and explains the LJPJT 2026 task, the Japanese Tort-case Dataset (JTD) on which it is based, and the LLM gpt-oss used in this study. Section 3 describes the details of our implementation and experimental methods. Section 4 presents the experimental results, and Section 5 provides a detailed analysis of the system outputs and discusses implications for future research directions based on these observations. Finally, Section 6 concludes this paper.

2 Related Work

2.1 The LJPJT 2025/2026 Tasks and the JTD Dataset

The dataset used in the LJPJT 2025 task is based on the JTD Dataset constructed by Yamada et al. [9], which consists of 3,477 Japanese court decisions. Each judgment may contain one or more tort instances, all defined under Article 709 of the Japanese Civil Code. For each tort, the dataset includes three types of text segments extracted from the judgment document: the undisputed facts, the plaintiff's claims, and the defendant's claims. The JTD Dataset was annotated by 41 legal experts.

The task consists of two subtasks: *Tort Prediction* (TP) and *Rationale Extraction* (RE). In the RE subtask, the goal is to predict whether each individual claim made by the plaintiff or the defendant is accepted by the court. In the TP subtask, the system predicts the court's decision for each tort, indicating whether tort liability is established.

Table 1 shows an example tort instance used in the task. Our system aims to predict the court's decision for each tort, along with whether each claim is accepted. Further details on the textual structure of the input can be found in [9]. Unlike other COLIEE datasets, this dataset is provided solely in Japanese.

For reproducibility, the task organizers allow the use of publicly available large language models (LLMs), while prohibiting the use of models that do not provide open weights, such as GPT-5 and Gemini.

Table 1: An example tort instance, with texts cited from [9].

	Input	Output
Rationale Extraction		
Undisputed Facts (UFs)		
UF_1	A posting "Mr. X1, you should pay back the money" was made in the website D, via IP address ***.***.***.***.	—
Plaintiff's Claims (PCs)		
PC_1	This posting, based on a viewer of ordinary prudence and his way of viewing, indicates the fact that a person named "X1," who works at factory B, borrowed money from a certain individual but has not repaid it.	Y
PC_2	There are only two persons with the surname "X1" who work at factory B: the plaintiff and his cousin.	Y
PC_3	The viewers of this posting, who know the plaintiff but do not know the plaintiff's cousin, would regard the plaintiff as the subject of the posting.	N
PC_4	It is possible to identify the subject of this posting as the plaintiff.	N
Defendant's Claims (DCs)		
DC_1	We do not admit all the allegations from plaintiff.	Y
Tort Prediction		
Court Decision		N

2.2 gpt-oss

GPT (Generative Pre-trained Transformer) [7] is a language model that performs autoregressive text generation. It encodes input sequences using self-attention and generates natural language by sequentially predicting the next token. This architecture enables a single model to handle a wide range of natural language processing (NLP) tasks in a unified manner, achieving strong performance in contextual understanding, reasoning, summarization, dialogue, and code generation through large-scale pretraining.

gpt-oss [6] is an open-weight large language model released by OpenAI in 2025. The model provides a configurable "reasoning level" that controls the degree of reasoning performed during inference. This level can be specified in the system prompt with three options: "low," "medium," and "high," allowing users to adjust how extensively the model generates its chain of thought during inference.

Furthermore, gpt-oss supports structured output generation, enabling precise control over output formats beyond natural language. Specifically, it can produce responses in JSON or other predefined schemas that are easily processed programmatically. This feature is particularly useful for tool invocation and agent-based systems that require safe and consistent outputs.

Among its variants, we used the model known as `gpt-oss-120b` in this paper.

3 Implementation and Experiments

We developed three systems for the LJPJT 2026 task, all of which are based on gpt-oss [6] and implement a retrieval-augmented generation (RAG) framework with few-shot examples. In this section, we describe the details of our implementation.

We used the following datasets:

Training dataset for the LJPJT 2026 task This dataset contains claims from past cases along with their corresponding ground-truth labels. It consists of 6,508 cases.

Training split of the JTD Dataset [9] This dataset[2] was additionally used to retrieve the original judgment documents corresponding to the above training dataset. Although the full dataset is larger, we used only the 6,356 cases that could be aligned with the LJPJT training dataset.

Test dataset for the LJPJT 2026 task This dataset consists of 802 cases and serves as the prediction target. The goal of the LJPJT 2026 task is to assign labels for both rationale extraction (RE) and tort prediction (TP) to each case in this dataset.

In our implementation, for each case in the test dataset, gpt-oss predicts labels for both RE and TP. In addition to the claims in the input case, we retrieve the most similar cases from the training dataset and use them as a basis for generating auxiliary knowledge. Specifically, we convert the retrieved judgment documents into short textual abstractions, which we refer to as "lessons." During prediction, the generated lessons are provided as supplementary context, guiding the model in making decisions on individual claims and tort liability.

3.1 Retrieval of Relevant Documents

In this subsection, we describe the method used to compute the similarity between an input case (query) and each document in the training data, in order to retrieve documents containing the most similar cases. Specifically, we compute the similarity between two sets of claims: $D_i = (D_i^{(p)}, D_i^{(d)})$ representing the sets of claims in a document, and $Q = (Q^{(p)}, Q^{(d)})$ representing those in a query. Here, each $d_{i,j} \in D_i^{(p)} \cup D_i^{(d)}$ and $q_k \in Q^{(p)} \cup Q^{(d)}$ is a short text segment consisting of at most one sentence, and p and d denote the plaintiff and the defendant, respectively.

Common RAG-based retrieval strategies, such as maximum or average similarity, may yield high scores if a document and a query share several similar claims. However, in actual legal decisions, a small number of remaining claims may play a decisive role in determining tort liability. Therefore, it is necessary to penalize redundant claims on the document side and to reward sufficient coverage of the claims in the query. To address this issue, we adopt a set-level similarity based on one-to-one matching between claims.

Concretely, the similarity between a document and a query is computed as follows. For ease of exposition, we describe the computation only for the plaintiff here; the same procedure is applied separately to the defendant in the same manner. First, for each pair of claims in the document and the query, we compute the cosine similarity $\mathrm{sim}(\mathrm{emb}(d_{i,j}), \mathrm{emb}(q_k))$. If the similarity is below a threshold $\tau(= 0.75)$, it is set to zero; otherwise, it is linearly rescaled to lie within the range $[0, 1]$. Here, $\mathrm{emb}(*)$ denotes the embedding representation of a text, and in our experiments we use embeddings computed by the `cl-nagoya/ruri-v3-310m`[3] model. Next, we perform maximum-weight matching on a bipartite graph between the claims in the query and those in the document. This matching enforces a one-to-one correspondence, where each query claim is matched to at most one document claim, and each document claim is matched to at most one query claim, while maximizing the total similarity. Finally, based on the resulting matching, we compute recall and precision between the query and the document, and derive the F1 score, which is used as the similarity between the query and the document for the plaintiff.

The same computation is applied to the defendant. The final similarity between a document and a query is defined as the average of the F1 scores for the plaintiff and the defendant. We then retrieve the top five documents with the highest similarity scores.

3.2 Generation of Lessons

From each retrieved document, we generate "lessons" from portions of the court's reasoning so that they are informative for the gpt-oss model. These lessons are generated by the gpt-oss model itself.

As described in Section 1, we construct multiple types of lessons by varying the information provided during generation:

KIS3 Only the training data provided in the LJPJT 2026 task, i.e., the text spans representing claims and their corresponding ground-truth labels, are used as input.

KIS2 Only the full text of the sections describing the court's reasoning in the original judgment documents corresponding to the LJPJT 2026 training data is used, without using the extracted spans or labels.

KIS1 Both the full text of the sections describing the court's reasoning in the original judgment documents and the claims with ground-truth labels, manually extracted from the LJPJT 2026 dataset, are used as input.

The prompts used for lesson generation are shown below. All prompt templates are written in English, while the inserted claims, judgment documents, and some of the generated lessons[4] are provided in Japanese.

3.2.1 KIS3. In the KIS3 system, lessons are generated using only the training data provided in the LJPJT 2026 task, i.e., the claim text spans and their corresponding labels. The prompt used for generation is shown in Figure 1.

3.2.2 KIS2. In the KIS2 system, lessons are generated using only the full text of the sections describing the court's reasoning in the original judgment documents. The extracted spans and labels provided in the LJPJT 2026 training data are not used.

An advantage of this approach is that the data can be obtained automatically from judgment documents without requiring expert annotation, allowing the construction of larger datasets in practice.

The prompt used for generation is shown in Figure 2. Here, `reference` denotes the portion of the original judgment document extracted from the line matching the regular expression `/(判\s*断|理\s*由)$/m` onward. This expression matches section headings such as "当裁判所の判断," "争点に対する判断," "判断," "裁判所の判断," and "争点についての判断," all of which serve as section titles for the court's reasoning; the corresponding sections typically appear near the end of the document[5], enabling

[2]We used the "Japanese Tort-case Dataset (Original Documents Only)" provided by LIC Co., Ltd through Gengo-Shigen-Kyokai. https://www.gsk.or.jp/en/catalog/gsk2024-a/

[3]https://huggingface.co/cl-nagoya/ruri-v3-310m

[4]Since the language of the output is not explicitly specified during lesson generation, the gpt-oss model may produce outputs in either Japanese or English.

[5]In non-heading lines, a sentence-final period is typically present, and thus the expression does not match.

Developer prompt:

```
Using the Reference and the provided labels,
output only the lesson(s) needed to predict the
same kinds of labels in similar cases later.
Include only case-specific, decision-relevant
points that help distinguish accepted from not
accepted claims. Do not output your reasoning
process, restate the labels, or include details
from the Reference that are not useful for
prediction.
Output a plain string only.
```

User prompt:

```
{
  "plaintiff_claims": [
    {"text": "…", "accepted": false}, …
  ],
  "defendant_claims": [
    {"text": "…", "accepted": true}, …
  ],
  "tort_accepted": true,
}
```

Figure 1: Prompt used for lesson generation in the KIS3 system

Developer prompt:

```
For each claim, decide whether it is accepted in
the Reference.
Then write ‘lessons‘: only the case-specific
points learned from the Reference that are needed
 to make the same decisions later without the
Reference.
Do not output your reasoning process or anything
not learned from the Reference.
Output valid JSON only:
{"plaintiff":[bool,...],"defendant":[bool
,...],"tort":bool,"lessons":str}
```

User prompt:

```
{
  "plaintiff_claims": ["…", …],
  "defendant_claims": ["…", …],
  "reference": "当裁判所の判断\n…",
}
```

Figure 2: Prompt used for lesson generation in the KIS2 system

extraction of the court's reasoning from judgment documents with varying formats.

The full text of the judgment document is not used due to input length limitations of the LLM. Documents that do not contain a line matching this pattern are excluded, and lessons are instead generated from other (next most similar) documents.

3.2.3 KIS1. In the KIS1 system, lessons are generated using both the full text of the sections describing the court's reasoning and the claims with ground-truth labels extracted from the LJPJT 2026 task. Although this approach requires expert annotation to expand the dataset, it is expected to benefit from both high-quality labels and the surrounding contextual information in the original judgment documents. The prompt used for generation is shown in Figure 3. The extraction method for `reference` is the same as in KIS2.

Developer prompt:

```
Using the Reference and the provided labels,
output only the lesson(s) needed to predict the
same kinds of labels in similar cases later.
Include only case-specific, decision-relevant
points that help distinguish accepted from not
accepted claims. Do not output your reasoning
process, restate the labels, or include details
from the Reference that are not useful for
prediction.
Output a plain string only.
```

User prompt:

```
{
  "plaintiff_claims": [
    {"text": "…", "accepted": false}, …
  ],
  "defendant_claims": [
    {"text": "…", "accepted": true}, …
  ],
  "tort_accepted": true,
  "reference": "当裁判所の判断\n…",
}
```

Figure 3: Prompt used for lesson generation in the KIS1 system

3.3 Prediction

Our goal is to predict, for each case, whether each claim is accepted for the RE task and whether tort liability is established for the TP task. The generated lessons are provided as supplementary context, allowing the model to make predictions by drawing analogies between the input case and retrieved examples.

The prompt template used for prediction is identical across all systems (KIS1, KIS2, and KIS3) and is shown in Figure 4. The field `shared_premises` contains the undisputed facts as provided in the dataset[6]. The field `references` contains five lessons generated in the previous subsection, whose content differs across the three systems.

We specify the response format to ensure that the model outputs the required number of Boolean values for each case.

3.4 Hyperparameters

In our experiments, we used the `openai/gpt-oss-120b`[7] model. We did not perform fine-tuning and instead designed only the developer and user prompts.

For lesson generation, we set temperature=1.0 and top_p=1.0 to encourage more diverse outputs. For prediction, we set

[6] In our preliminary experiments, when we asked the gpt-oss model to revise a draft prompt, it produced the following suggestion: "I prefer `shared_premises` over `undisputed_facts`, because it nudges the model toward facts that function as premises in reasoning, not just facts nobody disputed." Based on this, we adopted the name `shared_premises`.

[7] https://huggingface.co/openai/gpt-oss-120b

Developer prompt:

```
Using the supplied lesson from a similar case,
predict for each claim whether it would be
accepted in the target case. Judge by analogical
fit to the lesson, not by whether the lesson
explicitly states the claim.
Also output `tort`, meaning whether tort
liability would be recognized overall.
Output valid JSON only, with claim orders
preserved:
`{"plaintiff":[bool,...],"defendant":[bool
,...],"tort":bool}`
```

User prompt:

```
{
  "plaintiff": ["…", …],
  "defendant": ["…", …],
  "tort": null,
  "shared_premises": ["…", …],
  "references": ["…", …],
}
```

Response format:

```
{"plaintiff": [bool, …], "defendant": [bool, ,
…], "tort": bool}
```

Figure 4: Prompt used for prediction

temperature=0.0 and top_p=1.0 to improve reproducibility, and `reasoning_effort` to "medium," as the model needs to consider multiple lessons retrieved from related cases when making predictions. All other hyperparameters were left at their default values.

4 Results

Table 2 presents the results of our systems on the LJPJT 2026 task. In addition, although it was not included in our formal run submissions, we also report, for reference, the results of the KIS5 system [2], which achieved the best performance on the RE task in the previous round of COLIEE (i.e., the LJPJT 2025 task), evaluated on the LJPJT 2026 test dataset.

Our results show that KIS2 consistently outperforms KIS3. While there is little difference in performance between KIS1 and KIS3, KIS1 achieves slightly better performance on the main evaluation metrics of the LJPJT 2026 task, namely accuracy for the TP task and F1 (Micro) for the RE task. However, none of our proposed gpt-oss-based models outperformed the ModernBERT-based KIS5 system.

5 Discussions

Tables 3 and 4 present examples of the lessons generated by the KIS1, KIS2, and KIS3 systems. For lessons generated in Japanese, we additionally provide reference translations for the reader's convenience. Interestingly, the lessons generated by KIS1 and KIS3, which are provided with ground-truth labels, tend to focus on the court's decisions, as indicated by the presence of keywords such as "accepted" and "主張" (assertion). In contrast, the lessons generated by KIS2 tend to include expressions of legal concepts related to torts, such as "名誉毀損" (defamation) and "payment," suggesting a stronger focus on the court's reasoning process.

Although this difference may potentially be mitigated by improving prompt design, we hypothesize that in KIS2, where ground-truth labels are not provided, the model is required to infer labels from the documents themselves, which prevents shortcutting and leads to the emergence of reasoning processes in the generated lessons.

Despite relying only on automatically constructed training (i.e., reference) data without manual annotation, KIS2 achieves better performance than KIS1 and KIS3, which use manually annotated data. This suggests that further performance improvements may be possible by increasing the number of candidate cases and generating a larger number of lessons[8].

The slight performance difference between KIS1 and KIS3 also suggests that the original judgment documents may contain knowledge that is not captured in the extracted claim texts.

Although our approach does not yet outperform models based on fine-tuned ModernBERT [2], future work is expected to leverage the richness of automatically generated data from judgment documents, as in this study, while also exploiting knowledge unique to such documents to develop improved fine-tuning strategies. In this context, it is necessary to explore not only prompt-based text generation, as in our implementation, but also more sophisticated methods for representing and generating knowledge, particularly those suited for BERT-like models.

Finally, our study focuses on prompt design and retrieval data in the RAG framework. However, as demonstrated in the previous round of LJPJT 2025, fine-tuning can further improve performance when using LLMs [4]. Therefore, incorporating legally relevant knowledge, such as that explored in this study, into the fine-tuning of LLMs is another promising direction for future research.

6 Concluding Remarks

In this paper, we evaluated the effectiveness of knowledge retrieval from judgment documents in the COLIEE 2026 Pilot Task (LJPJT 2026). The goal of this task is to predict, for each case, whether each claim is accepted by the court and whether tort liability is established. Our results show that lessons generated from raw judgment documents of other cases are more useful for prediction than those generated from manually annotated information at the claim level.

These findings suggest a potential direction for improving performance without relying on manual annotation. Although we did not perform fine-tuning in this study, we also discussed possible directions for fine-tuning based on insights from the previous COLIEE round. In this study, lessons were generated using relatively simple prompts; in future work, we plan to construct more sophisticated knowledge bases.

[8] In this study, retrieved documents are selected based on extracted claims. However, as shown in our previous work [2], extracting claims from documents is not trivial, and further research is required to address this issue.

Table 2: Our results for the LJPJT 2026, pilot task of COLIEE 2026.

Model	TP	RE								
	Accuracy	Accuracy			F1 (Micro)			F1 (Macro)		
		All	Pltf.	Deft.	All	Pltf.	Deft.	All	Pltf.	Deft.
KIS1	0.6663	0.5855	**0.6280**	0.5433	0.5734	0.6425	0.4942	0.4485	0.2882	0.3066
KIS2	0.6438	**0.6281**	0.5979	**0.6581**	0.6261	0.6190	0.6340	0.5201	0.3125	0.3552
KIS3	0.6426	0.5865	0.5909	0.5821	0.5733	0.5911	0.5544	0.4723	0.2949	0.3448
KIS5	**0.7111**	0.6045	0.6078	0.6012	**0.6865**	**0.6918**	**0.6812**	**0.6017**	**0.4185**	**0.5078**

Acknowledgments

This research was partially supported by Kakenhi, MEXT Japan (JP23K22076, JP25K21648), JST PRESTO (JPMJPR2461), JST AIP Acceleration Research (JPMJCR22U4), and SECOM Science and Technology Foundation.

References

[1] Randy Goebel, Yoshinobu Kano, Mi-Young Kim, Calum Kwan, Juliano Rabelo, Ken Satoh, Hiroaki Yamada, and Masaharu Yoshioka. 2026. The COLIEE 2025 Competition on Legal Information Extraction and Entailment: Overview, Discussion, and Dataset Expansion. *The Review of Socionetwork Strategies* (20 Feb 2026). doi:10.1007/s12626-026-00199-9

[2] Kazuma Kadowaki and Yoshinobu Kano. 2026. Japanese Legal Judgment Prediction Using ModernBERT and Generative AI-based Information Extraction. *The Review of Socionetwork Strategies* (03 Mar 2026). doi:10.1007/s12626-026-00201-4

[3] Yoshinobu Kano, Randy Goebel, Mi-Young Kim, Calum Kwan, Ken Satoh, Hiroaki Yamada, and Masaharu Yoshioka. 2026. An Overview of the COLIEE 2026 Competition: Legal Case Law and Statute Law Information Retrieval and Entailment.. In *Proceedings of the 21st International Conference on Artificial Intelligence and Law (ICAIL 2026)*.

[4] Dat Nguyen, Minh-Phuong Nguyen, Quang-Huy Chu, Son T. Luu, Nguyen-Hoang Chu, Trung Vo, and Le-Minh Nguyen. 2026. Enhancing Legal Text Processing and Structural Analysis with Large Language Models at COLIEE 2025. *The Review of Socionetwork Strategies* (25 Mar 2026). doi:10.1007/s12626-026-00211-2

[5] Hoang-Trung Nguyen, Tan-Minh Nguyen, Xuan-Bach Le, Tuan-Kiet Le, Khanh-Huyen Nguyen, Ha-Thanh Nguyen, Thi-Hai-Yen Vuong, and Le-Minh Nguyen. 2025. NOWJ@COLIEE 2025: A Multi-stage Framework Integrating Embedding Models and Large Language Models for Legal Retrieval and Entailment. In *Proceedings of the Workshop on the Twelfth International Competition on Legal Information Extraction and Entailment (COLIEE 2025)*. 47–56.

[6] OpenAI. 2025. gpt-oss-120b & gpt-oss-20b Model Card. arXiv:2508.10925 [cs.CL] https://arxiv.org/abs/2508.10925

[7] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. https://cdn.openai.com/research-covers/language-unsupervised/language_understanding_paper.pdf

[8] Sabine Wehnert, Bhavya Baburaj Chovatta Valappil, and Ernesto William De Luca. 2025. LLMs, Knowledge Graphs, and Hybrid Search: Task-Specific Approaches to Legal AI in COLIEE. In *Proceedings of the Workshop on the Twelfth International Competition on Legal Information Extraction and Entailment (COLIEE 2025)*. 102–111.

[9] Hiroaki Yamada, Takenobu Tokunaga, Ryutaro Ohara, Akira Tokutsu, Keisuke Takeshita, and Mihoko Sumida. 2024. Japanese tort-case dataset for rationale-supported legal judgment prediction. *Artificial Intelligence and Law* (11 May 2024), 25 pages. doi:10.1007/s10506-024-09402-0

Table 3: Examples of lessons generated by our systems

tort_id	KIS1	KIS2	KIS3
0	Plaintiff claim accepted when it (1) identifies the plaintiff (or its members) by specific factual or opinion statements, (2) shows those statements are false or based on false facts, and (3) demonstrates that a reasonable ordinary reader would perceive the statements as lowering the plaintiff's social reputation. Defendant claim accepted only when it successfully establishes a public-interest defence: the statements must concern matters of general public concern (e.g., the nature and policy of competing labor unions) and be presented as factual reporting or opinion on those matters. Other defenses (asserting no new reputational harm, truth of the statements, or reasonable belief in truth) are rejected unless the truth of the alleged facts is proven or the defendant's belief is shown to be reasonably founded.	1. 名誉毀損は、一般の読者が普通の注意で読むと社会的評価を低下させるかで判断する。2. 事実の摘示が虚偽であるか、または公正な論評の前提となる事実が真実でない場合、名誉毀損が成立する。3. 公共の利害に関する事実であっても、事実が真実であることの証明がない、または信じるに足る合理的根拠が欠けるときは違法性は免除されない。4. 組合の公式文書や内部資料だけに依拠して事実を信じたとしても、情報源の信用性が低く、対立関係がある場合は「相当の理由」が認められない。5. 正当な利益の擁護や反論としての言論でも、虚偽の事実を根拠にした場合は防御にならない。6. 組合活動の性格・運動方針に関する情報は公共性が認められるが、虚偽の事実が含まれるときは名誉毀損の防御にならない。 1. Defamation is determined by whether, when an ordinary reader reads it with ordinary care, it lowers social evaluation. 2. If the statement of fact is false, or if the facts forming the premise of fair comment are not true, defamation is established. 3. Even if the facts concern matters of public interest, illegality is not exempted when there is no proof that the facts are true, or when there is a lack of reasonable grounds sufficient to believe them. 4. Even if one believed the facts based only on the union's official documents or internal materials, when the reliability of the information sources is low and there is an adversarial relationship, "reasonable grounds" are not recognized. 5. Even speech as the protection of a legitimate interest or as a rebuttal does not constitute a defense when it is based on false facts. 6. Information concerning the character of union activities and movement policy is recognized as having a public character, but when false facts are included, it does not constitute a defense to defamation.	- 原告の主張が「事実の摘示」または「虚偽の事実に基づく意見・論評」であり、かつそれが不特定多数に読まれ原告の社会的評価を著しく低下させると認められるときは名誉毀損として受容される。 - 被告が「事実はすべて真実」「意見は公正な論評」と主張しても、具体的な事実確認や証拠が示されなければ認められない。単なる内部文書や上部組織の承認だけでは真実性の立証とはならない。 -「名誉毀損がない」「社会的評価の新たな低下はない」といった主張は、実質的な評価低下の有無が客観的に示されない限り退けられる。 -「公共の利害に関する事実に係る」ことを防御理由とするには、掲示内容が ** 労働組合の性格・運動方針・組合選択の材料 ** など、企業内で多数の従業員が関心を有する公共的事実であることを具体的に示す必要がある。これが認められたものだけが防御として受容される。 -「公式文書に基づく」「上部組織の確認を得た」などの手続的正当性は、事実の真実性や公正な論評性の証明には直結せず、単独での防御は認められない。 -「自己の正当な利益を擁護するための言論」「表現の形式的違法性欠如」など、動機・方法・内容の総合的判断だけで違法性を否定する主張は、事実の虚偽性や評価低下の有無がクリアされない限り認められない。 - 名誉毀損に対する防御として有効なのは、** 真実性 **、** 公正な論評 **、** 公共の利害に関する事実の摘示 ** のいずれかが具体的かつ客観的に証明された場合のみである。 - The plaintiff's assertions are accepted as defamation when they constitute a "statement of fact" or an "opinion/comment based on false facts," and when it is recognized that they were read by an unspecified large number of people and significantly lowered the plaintiff's social evaluation. - Even if the defendant argues that "all facts are true" and that "the opinions are fair comment," this will not be accepted unless concrete fact-checking or evidence is presented. Mere internal documents or approval by a superior organization do not constitute proof of truth. - Assertions such as "there is no defamation" or "there is no new lowering of social evaluation" are rejected unless the presence or absence of an actual lowering of evaluation is shown objectively. - In order to use as a defense that the matter "relates to facts concerning the public interest," it is necessary to show specifically that the posted content consists of public facts in which many employees within the company have an interest, such as **the character of the labor union, its movement policy, and material for choosing a union**. Only what is recognized as such is accepted as a defense. - Procedural legitimacy such as "based on official documents" or "confirmed by a superior organization" does not directly lead to proof of the truth of the facts or of fair comment, and by itself is not accepted as a defense. - Assertions that deny illegality solely on the basis of a comprehensive judgment of motive, method, and content, such as "speech to defend one's legitimate interests" or "the absence of formal illegality in the form of expression," are not accepted unless the falsity of the facts and the presence or absence of lowered evaluation are cleared. - What is effective as a defense against defamation is only a case in which one of **truth**, **fair comment**, or **the statement of facts concerning the public interest** has been proved specifically and objectively.

Table 4: Examples of lessons generated by our systems (continued)

tort_id	KIS1	KIS2	KIS3
1	- 支払義務の消滅を争っている旨の主張は、相手方が債務の存続を認めている事実と合致すれば「受理」される。 - 支払の一部を子の口座に振り込んだことが、相手方の了解なく行われ、かつ相手方が実質的に口座を管理していなかった場合、その振込は債務の弁済とは評価できず、相手方が債務の不存在を主張することは「認められない」。 - 相手方が口座への振込について明示的な同意をしておらず、管理権も持っていないという事実があるとき、相手方が「了解した」や「管理していた」等の主張はすべて「受理」される。 - 差押えが債務残高を超えていない、又は正当な執行手続きに基づくものである場合、差押えが違法で不法行為になる、あるいは不当利得を得たという主張は「認められない」。 --- - The assertion that one is disputing the extinguishment of the payment obligation is "accepted" if it accords with the fact that the other party acknowledges the continued existence of the debt. - Where part of the payment was remitted to the child's account without the other party's consent, and the other party was not in substance managing that account, that remittance cannot be evaluated as performance of the debt, and it is "not accepted" that the other party may assert the non-existence of the debt. - When there is the fact that the other party did not give explicit consent to remittance into the account and also did not have the authority to manage it, all assertions by the other party such as that they "consented" or "were managing it" are "accepted." - Where the attachment does not exceed the outstanding debt amount, or is based on lawful execution procedures, assertions that the attachment was illegal and constituted a tort, or that unjust enrichment was obtained, are "not accepted."	A payment made to a third-party (e.g., a child's) account does not discharge a marital-support debt unless the creditor explicitly approves the payment method and exercises actual control over that account. Merely knowing about the payment or the account's existence is insufficient for consent. Without such consent and control, the creditor's claim of unpaid support remains valid, and any seizure of the debtor's assets for that claim is not an unlawful tort.	- Accept statements that plainly describe **documented actions or positions** of the parties (e.g., actual payments made, the defendant's explicit contest of the plaintiff's claim, or the existence of a remaining monetary claim in the attached debt list). - Reject statements that **assert unverified conduct or conclusions** (e.g., claims of no objection, receipt of money without dispute, understanding of an email, fulfillment of obligations, or the propriety of a seizure) unless they are directly supported by the record.
2	Defamation claim is accepted when the article, read as a whole by an ordinary reader, conveys false factual imputations that lower the plaintiff's social reputation; isolated excerpts must be evaluated in context. A defendant's truth defense succeeds only for factual portions that are proven true and relate to matters of public interest (e.g., criminal conduct). Claims that a statement is merely peripheral or unrelated to reputation are rejected if the article's overall thrust still harms reputation; truth of non-essential background information does not bar liability.	Defamation is judged by the meaning of the whole article as read by an ordinary reader, not by isolated statements. If the key factual portions reported are true and concern matters of public safety, the article serves a public-interest purpose and the truth defence removes illegality, so no tort of defamation arises. Information unrelated to the criminal facts that does not lower the plaintiff's social reputation is not considered defamatory. The court rejected the plaintiff's claims and upheld all of the defendant's assertions that the reported facts were true, of public interest, and not improper.	Defendant's truth defense is accepted only when the contested statement is a factual finding confirmed by a criminal judgment and the reporting concerns a matter of public interest; if the statement is false, merely restates already reputationally harmful facts, or fails to serve public interest, the defense is rejected. Plaintiff's claim is accepted when the article contains false expressions that damage reputation and the defendant distributed them.

BJPWH at COLIEE 2026 Task 1: Combining BM25 and Dense Retrieval with Year-Based Filtering for Legal Case Retrieval

Calum Kwan
jckwan@ualberta.ca
University of Alberta
Edmonton, Alberta, Canada

Vidhi Sati
vsati@ualberta.ca
University of Alberta
Edmonton, Alberta, Canada

Nanribet Fwangje
fwangje@ualberta.ca
University of Alberta
Edmonton, Alberta, Canada

Xinjia Fan
xinjia2@ualberta.ca
University of Alberta
Edmonton, Alberta, Canada

Denilson Barbosa
denilson@ualberta.ca
University of Alberta
Edmonton, Alberta, Canada

Randy Goebel
rgoebel@ualberta.ca
University of Alberta
Edmonton, Alberta, Canada

Abstract

We report on the BJPWH team's submissions to Task 1 of COLIEE 2026, which concerned retrieving relevant legal cases given a query case. Our approach combines traditional lexical retrieval (BM25) with dense semantic retrieval (BERT embeddings) through weighted reciprocal rank fusion (RRF). We implement a year-based filtering mechanism to enforce legal precedent constraints, requiring that retrieved cases predate the query case. A key finding is that applying year filtering only to BERT results while leaving BM25 results unfiltered outperforms filtering both retrievers. We discuss in Section 5 why the two retrievers appear to err in complementary ways under temporal constraints. We submitted three runs: BJPWH1 (BERT-only with year filtering), BJPWH2 (BM25-only with year filtering), and BJPWH3 (hybrid RRF fusion with selective BERT filtering). BJPWH3 achieved our best performance with F1=0.2101 on the test set, ranking 14th among 22 participating teams.

1 Introduction

Legal case retrieval is the task of finding relevant precedent cases given a query case. This is a fundamental problem in legal information retrieval with direct applications to legal research, case law analysis, and judicial decision-making. The COLIEE (Competition on Legal Information Extraction/Entailment) workshop provides a benchmark dataset and evaluation framework for this task.

The COLIEE 2026 Task 1 dataset consists of legal cases from Canadian Federal Court. Given a query case, systems must retrieve a ranked list of relevant precedent cases from a corpus of candidate cases. The task presents several unique challenges compared to general document retrieval:

(1) **Long documents**: Legal cases are typically very long (thousands of tokens), requiring strategies to handle length normalization and computational constraints.
(2) **Specialized vocabulary**: Legal text contains domain-specific terminology and citation patterns not found in general text.
(3) **Temporal precedent constraints**: A legal case can only cite decisions that predate it, creating a strict temporal ordering constraint.
(4) **Bilingual content**: Canadian Federal Court cases frequently contain both English and French text, which affects lexical retrieval.

Our approach addresses these challenges through a hybrid retrieval system that combines the strengths of lexical and semantic retrieval methods. We make the following contributions:

(1) A systematic BM25 parameter tuning procedure adapted for long legal documents, finding optimal values of $k_1 = 4.5$ and $b = 1.0$.
(2) A year-based filtering mechanism that enforces temporal precedent constraints while preserving retrieval quality.
(3) A weighted reciprocal rank fusion (RRF) strategy for combining BM25 and BERT retrievers.
(4) Experimental analysis showing that selective year filtering (BERT-only) outperforms filtering both retrievers in the hybrid setting.

1.1 Organization

The remainder of this paper is organized as follows. Section 2 surveys related work on the COLIEE competition, legal information retrieval, and hybrid retrieval systems. Section 3 describes our methodology, including the dataset, BM25 and BERT retrieval components, year-based precedent filtering, and fusion strategies. Section 4 presents our experimental results, covering BM25 parameter tuning, fusion strategy comparisons, submission strategy optimization, and official test set performance. Section 5 discusses the key findings and their implications, and outlines limitations and future work. Section 6 concludes the paper. Appendix A reports on preliminary experiments with SAILER reranking and LLM-based case summarization that were explored but not included in our final submissions.

2 Related Work

2.1 The COLIEE Competition

The Competition on Legal Information Extraction/Entailment (COLIEE) has been held annually since 2014, with Task 1 focused on retrieving relevant precedent cases given a query case [5, 6]. The task uses a corpus of Canadian Federal Court cases with citation links suppressed, which requires systems to identify which cases a query case would "notice" based purely on content. F1 score is the primary evaluation metric. Top F1 scores have increased from around 0.19 in COLIEE 2021 to 0.44 in COLIEE 2024 [6], reflecting advances in neural retrieval and hybrid pipeline design.

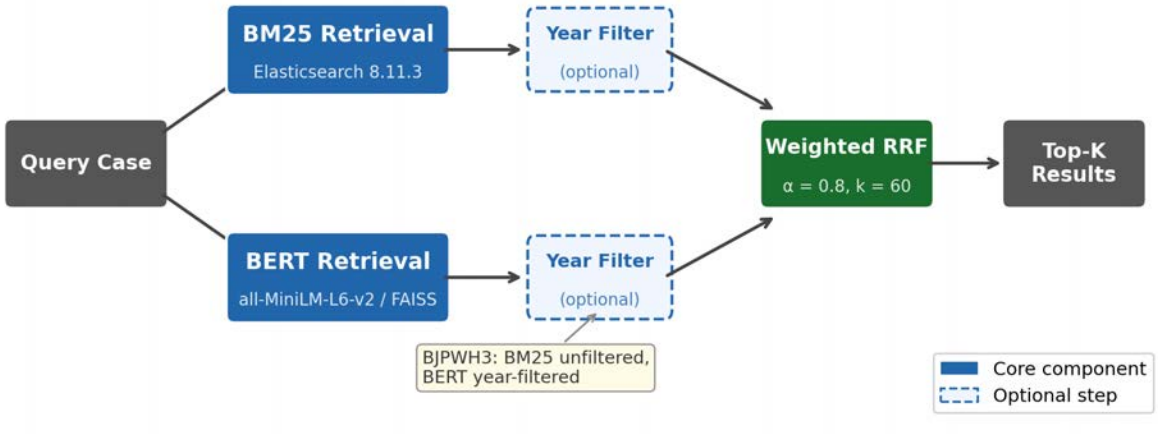


Figure 1: Overview of our hybrid retrieval system. Year filtering is applied as an optional post-processing step on either or both retrieval tracks. In our best submission (BJPWH3), filtering is applied to BERT only.

2.2 Legal Information Retrieval

Traditional lexical matching methods like BM25 remain competitive baselines for legal case retrieval due to their robustness and interpretability [15]. IITDLI at COLIEE 2023 demonstrated that a carefully tuned BM25 pipeline with year-based filtering and submission thresholding could achieve strong results without neural components [3]. LeiBi at COLIEE 2022 explored query reformulation and automatic summarization with Longformer-Encoder-Decoder as a means of generating more effective query representations [12], an approach conceptually related to our summarization experiments described in Appendix A.

Dense retrieval using BERT-based encoders has shown promise for capturing semantic similarity beyond keyword matching [14]. SAILER [10] is a structure-aware pre-trained language model developed specifically for legal case retrieval, which models the internal rhetorical structure of legal documents during pre-training. THUIR at COLIEE 2023 incorporated SAILER as part of a learning-to-rank ensemble, achieving the best Task 1 performance that year (F1=0.30) [11].

2.3 Hybrid Retrieval Systems

Combining lexical and semantic retrieval has been shown to improve performance across many retrieval domains. Reciprocal Rank Fusion (RRF) [2] provides a simple and effective method for combining ranked lists without requiring score normalization. NOWJ at COLIEE 2023 employed a multi-phase hybrid approach that combined BM25 for initial candidate selection with a fine-tuned semantic model for reranking, and also incorporated year-based filtering to enforce temporal precedent constraints [16]. Their 2023 system is the most methodologically similar to ours.

More recent editions have seen increasing use of large language models in retrieval pipelines. JNLP at COLIEE 2025 extended the hybrid paradigm by incorporating both BM25 scores and SAILER scores as features in a pairwise neural classifier, achieving the best Task 1 result in 2025 (F1=0.36) [13]. The dominant pattern across recent editions is a multi-stage pipeline: BM25 initial filtering for high recall, followed by neural reranking, followed by post-processing [8, 9].

Table 1: COLIEE 2026 Task 1 dataset statistics

Split	Cases	Query cases
Training	7,708	–
Test (candidate pool)	1,848	400

3 Methods

3.1 Dataset

The COLIEE 2026 Task 1 dataset consists of Canadian Federal Court case law files. Cases are provided as plain text with all citation links suppressed and replaced by a `FRAGMENT_SUPPRESSED` tag, preventing systems from exploiting raw citation strings. The training set contains 7,708 cases. The test set consists of a candidate pool of 1,848 full case files and 400 query cases provided as case IDs only.

Evaluation uses micro-averaged precision, recall, and F1. Cases contain both English and French text; French content was not explicitly removed in our pipeline. All training-set experiments reported in this paper were evaluated against the gold labels provided with the COLIEE 2026 training set.

3.2 BM25 Lexical Retrieval

We use BM25 as our primary lexical retrieval method, implemented via Elasticsearch 8.11.3 [4] with a custom similarity configuration. BM25 scores a document D against a query Q as:

$$\text{score}(D, Q) = \sum_i \text{IDF}(q_i) \cdot \frac{f(q_i, D) \cdot (k_1 + 1)}{f(q_i, D) + k_1 \cdot (1 - b + b \cdot |D|/\text{avgdl})} \quad (1)$$

where $f(q_i, D)$ is the term frequency of query term q_i in document D, $|D|$ is the document length, and avgdl is the average document length in the corpus.

The parameter k_1 controls term-frequency saturation: higher values allow a term's contribution to grow longer before plateauing. The parameter b controls document-length normalization: $b = 1.0$ applies full normalization, $b = 0.0$ applies none. Both parameters are tunable; we describe our tuning procedure in Section 4.1.

Implementation. We created a complete Elasticsearch-based infrastructure including: We built a complete Elasticsearch-based infrastructure supporting document preprocessing and bulk indexing, a custom BM25 similarity configuration with adjustable k_1 and b parameters, multi-field search across English and French text, and a configurable top-K retrieval interface.

This infrastructure enabled systematic parameter exploration to find optimal BM25 configurations for legal case retrieval.

3.3 BERT Dense Retrieval

For dense semantic retrieval, we employ a BERT-based approach using Sentence-BERT [14] embeddings. Given the length of legal documents, we adopt a chunk-based strategy:

Chunking. Each legal case is split into chunks of 250 tokens with 40-token overlap. This allows the embedding model to process the full document while respecting token limits.

Embedding. We use the `all-MiniLM-L6-v2` model [14] to encode each chunk into a dense vector representation. This model provides a good balance of quality and efficiency for sentence-level embeddings.

Indexing. Chunk embeddings are indexed using FAISS [7] for efficient similarity search.

Retrieval and Aggregation. For a query case, we:

(1) Encode the query case into chunks and embed each chunk
(2) Perform k-NN search in FAISS to find the top matching chunks
(3) Aggregate chunk-level scores to compute case-level relevance scores
(4) Rank candidate cases by aggregated scores

We initially retrieve the top-200 candidates per query before applying downstream filtering and fusion.On the training set, the BERT-only dense retriever (top-200 candidates, no year filtering) achieves MAP=0.0555, MRR=0.1118, R@100=0.2875, and R@200=0.3210.

3.4 Year-Based Precedent Filtering

Legal precedent rules require that a case can only cite decisions made before it. Year-based filtering to enforce this constraint has been used in prior COLIEE work [1, 11]. To enforce this temporal constraint, we implement year-based filtering as a post-processing step.

Year Extraction. We extract decision years from case text using rule-based pattern matching over the structured case header (e.g., neutral citation lines such as "2015 FC 725"), focusing on clearly marked decision years to avoid misidentifying other dates (e.g., cited cases, evidence dates) as the judgment year. Cases without reliable year information are left unfiltered. Our extraction achieved 97.2% coverage on the test set, successfully extracting years for 1,796 out of 1,848 cases.

Filtering Application. For each query case with year y_q, we remove all candidates with decision years $y_c \geq y_q$. This ensures retrieved cases temporally precede the query case.

Selective Filtering. We evaluate four filtering configurations:

- No filtering
- BM25 results filtered only
- BERT results filtered only
- Both filtered

This allows us to isolate the effect of year filtering on each retriever and their combination. As we show in Section 4, filtering BERT results while leaving BM25 unfiltered produces the best hybrid performance.

The year filtering reduced BERT candidates by 40.7% (from 50,337 to 29,855) and BM25 candidates by approximately 36%. The larger reduction for BERT is consistent with our later finding that BERT more readily retrieves temporally invalid cases - and that filtering its output selectively is therefore the more impactful intervention.

3.5 Hybrid Retrieval: Fusion Strategies

To combine lexical and semantic retrieval signals, we design and evaluate two fusion strategies. Both take as input a ranked list from BM25 and a ranked list from BERT, and produce a single merged ranking.

Weighted Score Fusion. The first strategy normalizes the raw scores from each retriever using min-max normalization and combines them as a weighted sum:

$$\text{score}_{\text{hybrid}}(d) = \alpha \cdot \text{norm}_{\text{BM25}}(d) + (1 - \alpha) \cdot \text{norm}_{\text{BERT}}(d) \quad (2)$$

where $\text{norm}_x(d) = (\text{score}_x(d) - \min_x)/(\max_x - \min_x)$ and $\alpha \in [0, 1]$ is the BM25 weight, treated as a hyperparameter. A value of $\alpha = 1$ reduces to pure BM25.

Weighted Reciprocal Rank Fusion (RRF). The second strategy combines document ranks rather than normalized scores. Because BM25 and BERT scores live on incompatible scales, rank-based combination is more robust than score-based combination. We use a weighted extension of the standard RRF formula [2]:

$$\text{score}_{\text{RRF}}(d) = \alpha \cdot \frac{1}{k + \text{rank}_{\text{BM25}}(d)} + (1 - \alpha) \cdot \frac{1}{k + \text{rank}_{\text{BERT}}(d)} \quad (3)$$

where $k = 60$ is the standard RRF constant that dampens the influence of very highly-ranked documents, and α is the BM25 weight. Documents absent from one retriever's list contribute 0 from that term.

As we show in Section 4.2, RRF substantially outperforms score fusion, so we use RRF for our final submission.

Table 2: Phase 1 BM25 grid search results (training set)

k_1	b	MAP	MRR	R@100	R@200
1.2 (baseline)	0.75	0.0992	0.1562	0.5164	0.6039
1.5	0.75	0.1023	0.1595	0.5287	0.6130
1.2	0.50	0.0863	0.1412	0.4795	0.5618
1.5	0.50	0.0941	0.1505	0.5092	0.5989
1.5	0.30	0.0948	0.1505	0.5196	0.6161
2.0	0.75	0.1054	0.1626	0.5343	0.6178
2.0	0.50	0.1012	0.1563	0.5249	0.6136
2.0	0.30	0.1041	0.1615	0.5410	0.6321

Table 3: Phase 2 BM25 grid search results (training set)

k_1	b	MAP	MRR	R@100	R@200
2.0	1.0	0.1054	0.1614	0.5311	0.6144
3.0	0.9	0.1165	0.1730	0.5709	0.6520
3.0	1.0	0.1150	0.1690	0.5638	0.6465
3.5	0.9	0.1172	0.1731	0.5763	0.6536
3.5	1.0	0.1181	0.1721	0.5739	0.6553
4.0	0.9	0.1178	0.1726	0.5793	0.6558
4.0	1.0	0.1205	0.1748	0.5823	0.6620

4 Experiments

4.1 BM25 Parameter Tuning

We conducted a three-phase grid search over k_1 and b to find the BM25 configuration that maximizes candidate recall on the training set. R@200 is the primary tuning metric because BM25 served as the first-stage retriever: its role was to produce a high-recall candidate pool of 200 cases per query for downstream fusion and filtering. Errors at this stage are difficult to recover later, making recall the primary concern at this stage.

Phase 1: Initial Search. We tested $k_1 \in \{1.2, 1.5, 2.0\}$ and $b \in \{0.3, 0.5, 0.75\}$, evaluating each configuration with a fresh Elasticsearch index. Results are shown in Table 2.

Phase 1 showed that higher k_1 (2.0) and lower b (0.30) improved R@200, suggesting two things: legal cases benefit from less aggressive term-frequency saturation, and the default length normalization ($b = 0.75$) over-penalizes long legal documents.

Phase 2: Refined Search. Based on Phase 1, we expanded the grid to $k_1 \in \{2.0, 3.0, 3.5, 4.0\}$ and $b \in \{0.9, 1.0\}$. Results are shown in Table 3.

Increasing k_1 continued to improve recall monotonically across this range, with $k_1 = 4.0$ achieving the best R@200 so far (0.6620), motivating a finer search around this value.

Phase 3: Final Refinement. We tested $k_1 \in \{4.0, 4.5, 5.0\}$ with $b = 1.0$ to find the optimal saturation parameter. Results are shown in Table 4.

$k_1 = 4.5$ with $b = 1.0$ achieved the best R@200 (0.6681), with corresponding improvements in MAP and MRR. $k_1 = 5.0$ showed a slight regression, confirming that $k_1 = 4.5$ is the saturation point for this corpus.

Table 4: Phase 3 BM25 grid search results (training set)

k_1	b	MAP	MRR	R@100	R@200
4.0	1.0	0.1205	0.1748	0.5823	0.6620
4.5	1.0	0.1227	0.1773	0.5916	**0.6681**
5.0	1.0	0.1218	0.1762	0.5904	0.6669

4.2 Fusion Strategy Comparison

We compared weighted score fusion and weighted RRF across different α values and year-filtering configurations.

We swept $\alpha \in \{0.1, 0.2, \ldots, 0.9\}$ in steps of 0.1 for score fusion, and $\alpha \in \{0.80, 0.85, 0.90, 0.92, 0.94, 0.96, 0.98\}$ for RRF, reporting the value that maximized R@200 in each case. Table 5 shows the results.

Score fusion provided only marginal gains over BM25 alone (MAP: 0.1244 vs. 0.1227), confirming that min-max normalization does not adequately reconcile the score distributions of the two retrievers.

RRF substantially outperformed both BM25 and score fusion on ranking metrics – applying BERT year-filtering within the RRF setting pushed MAP to 0.1563 and MRR to 0.2583. Filtering both retrievers further improved top-rank precision (MRR: 0.2636) but at a significant recall cost (R@200 fell from 0.6655 to 0.6019), making it unsuitable as a general submission strategy.

We selected **RRF with BERT year-filtered and BM25 unfiltered** ($\alpha = 0.8$) as our final system (BJPWH3), as it provided the best balance of recall and ranking quality.

4.3 Submission Strategy Optimization

We systematically tested different submission strategies to determine the optimal number of results to return per query. We compared:

- **Fixed K strategies**: Return exactly K results for all queries ($K \in \{5, 10, 11, 15, 100\}$)
- **Dynamic K strategies**: Return variable numbers of results based on score thresholds ($\tau \in [0.3, 0.8]$)

In total, we tested 45 configurations on the training data. Results are visualized in Figure 4 and Figure 5.

Fixed $K = 10$ was optimal across all metrics. The sweet spot in Figure 5 shows that returning 10–15 results per query maximizes F1, with performance degrading on both sides: very small K (e.g., $K = 5$) sacrifices recall, while large K (e.g., $K = 100$) floods the submission with low-precision results. Dynamic threshold strategies generally underperformed their fixed-K equivalents, likely because score distributions vary across queries and a single threshold does not transfer well.

Based on this analysis, all three of our final submissions used Fixed $K = 10$.

Additional experiments with SAILER reranking and LLM-based case summarization are described in Appendix A.

4.4 Test Set Results

Table 6 shows the official test set results for all three BJPWH submissions. BJPWH3 (hybrid RRF with selective BERT year-filtering)

Table 5: System comparison: BM25 baseline vs. score fusion vs. RRF (training set)

System	α	MAP	MRR	R@100	R@200
BM25 (k_1=4.5, b=1.0)	–	0.1227	0.1773	0.5916	0.6681
Score fusion, no filter	0.9	0.1152	0.1681	0.5818	0.6606
Score fusion, BM25 yr-filt. (best)	0.9	0.1244	0.1821	0.5749	0.6366
Score fusion, BERT yr-filt.	0.9	0.1201	0.1730	0.5909	0.6683
Score fusion, both yr-filt.	0.9	0.1211	0.1784	0.5527	0.5998
RRF, no filter	0.80	0.1400	0.2325	0.5849	0.6648
RRF, BERT yr-filt. (BJPWH3)	**0.80**	**0.1563**	**0.2583**	**0.5886**	**0.6655**
RRF, both yr-filt. (ranking-focused)	0.92	0.1593	0.2636	0.5573	0.6019

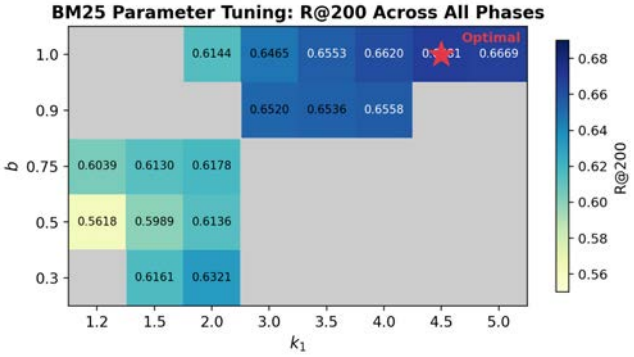


Figure 2: BM25 parameter tuning results showing R@200 across all tested k_1 and b configurations. Grey cells were not evaluated. The optimal configuration ($k_1 = 4.5$, $b = 1.0$, starred) represents an unusually high saturation parameter suited to long legal documents.

achieved the best F1 among our runs at 0.2101, confirming the training-set findings. BJPWH2 (BM25-only) performed comparably at F1=0.2087, underscoring the strength of the tuned BM25 baseline. BJPWH1 (BERT-only) lagged significantly at F1=0.1201, consistent with the known difficulty of applying dense retrieval to long legal documents without a strong lexical component.

Table 7 shows all official COLIEE 2026 Task 1 results. BJPWH3 ranked 14th among 22 participating teams by best-run F1. The top system, NOWJ, achieved F1=0.4220 using a multi-stage neural pipeline. For context, the top system in COLIEE 2025 achieved F1=0.3353 [9], and in COLIEE 2024 F1=0.4432 [6], placing the 2026 competition in line with recent editions. Our result sits in the lower-middle tier of the field, ahead of 8 teams including several with non-zero scores.

5 Discussion

5.1 BM25 Remains Strong

Despite the rise of neural retrieval methods, BM25 with carefully tuned parameters remains highly competitive for legal case retrieval. Our BM25-only system (BJPWH2) achieved strong performance, and BM25 contributed the majority of the signal in our best hybrid system, where $\alpha = 0.8$ heavily weights BM25.

The unusually high k_1 value we found ($k_1 = 4.5$ vs. the typical default of 1.2) reflects the characteristics of legal text. In a general corpus, domain-specific legal terms such as statute references and case citations would be rare, carrying high IDF scores that naturally distinguish relevant documents. However, within a corpus restricted to case law, these same terms appear frequently across many documents, compressing their IDF and reducing its discriminative power. This shifts the retrieval burden onto term frequency: allowing TF to contribute more before saturating, via higher k_1, compensates for the narrowed IDF range. Our three-phase grid search confirms this empirically, with R@200 improving from 0.604 at the default $k_1 = 1.2$ to 0.668 at $k_1 = 4.5$.

5.2 Selective Year Filtering

Our finding that filtering BERT results while leaving BM25 unfiltered produces the best hybrid performance was somewhat unexpected. We hypothesize two explanations:

(1) Error correction: BERT's semantic matching may sometimes retrieve anachronistic cases that are semantically similar but temporally invalid. Year filtering removes these errors while preserving BERT's strength in capturing semantic similarity.

(2) Diversity preservation: BM25's lexical matching is more conservative and less likely to retrieve anachronistic cases in the first place. Filtering BM25 results may remove valid early precedents without corresponding improvements in precision, reducing the diversity of the candidate pool.

Rigorous validation of these hypotheses - through case-level error analysis comparing the specific documents added or removed by each filtering configuration - is an important direction for future work.

5.3 RRF vs. Score Fusion

RRF substantially outperformed score fusion in our experiments. A likely explanation is that BM25 and BERT scores are on fundamentally incompatible scales: BM25 scores grow with document length and term frequency in ways that min-max normalization does not fully correct, whereas BERT cosine similarity scores are bounded and tightly clustered. Rank-based fusion sidesteps this problem entirely, since it only requires that both lists agree on relative ordering, not on absolute magnitude. The RRF smoothing constant $k = 60$ additionally dampens the outsized influence of the very top-ranked documents, providing more stable fusion across

Table 6: Official test set results for BJPWH submissions

Run	System	F1	Precision	Recall
BJPWH1	BERT-only, yr-filt.	0.1201	0.0870	0.1937
BJPWH2	BM25-only, yr-filt.	0.2087	0.1500	0.3429
BJPWH3	**Hybrid RRF, BERT yr-filt.**	**0.2101**	**0.1510**	**0.3451**

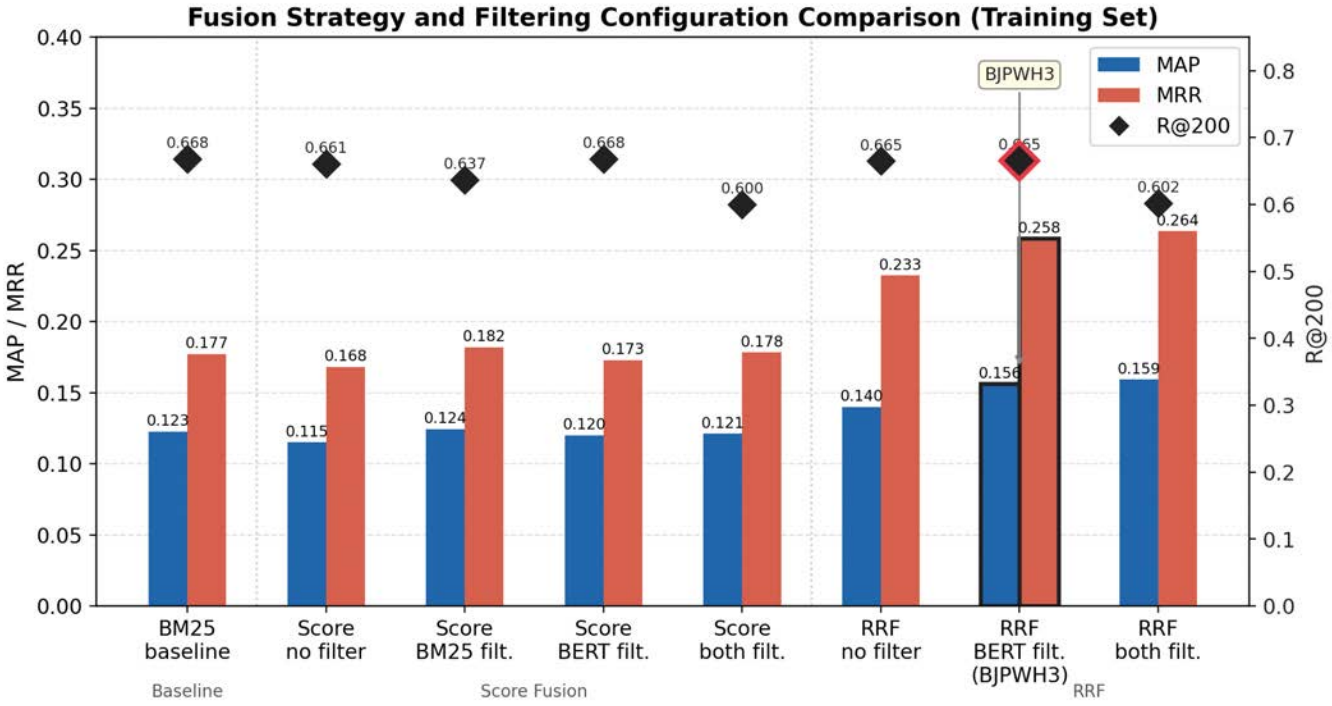


Figure 3: Comparison of fusion strategies and filtering configurations on training data. MAP and MRR (bars, left axis) show ranking quality; R@200 (dashed line, right axis) shows recall. BJPWH3 achieves the best balance: strong ranking gains with minimal recall cost. Note that BM25 alone retains the highest R@200 (0.6681); filtering both retrievers improves top-rank precision but substantially reduces recall.

queries. This behavior aligns with findings from other hybrid retrieval settings where rank-based fusion consistently outperforms score-based fusion [2].

5.4 Limitations and Future Work

Our pipeline achieves encouraging results given its relative simplicity, relying on well-established retrieval components without neural reranking or LLM integration. Nevertheless, it falls notably short of the top-performing systems, highlighting clear avenues for improvement.

Summarization: Due to computational constraints, we could not evaluate whether retrieval over LLM-generated summaries would improve performance. With sufficient compute resources, this remains a promising direction.

Query expansion: We did not experiment with query expansion techniques (e.g., pseudo-relevance feedback, legal thesaurus expansion). These could potentially improve recall.

Reranking: While SAILER reranking did not help in our experiments, other neural reranking models (e.g., cross-encoders fine-tuned on legal text) might yield better results.

Citation graphs and argument structure: Legal cases form a citation network that could be exploited for retrieval via graph-based ranking or link analysis. More broadly, legal cases follow recognizable rhetorical structures - issues, reasoning, findings - and methods that explicitly model this argument structure, such as structure-aware pre-training [10] or graph neural networks over argument graphs, may capture relevance signals that flat text retrieval misses entirely. We did not incorporate citation or argument structure in this work.

Table 7: Full COLIEE 2026 Task 1 results (best run per team, sorted by F1). BJPWH shown in bold.

Team	F1	Prec.	Rec.
NOWJ	0.4220	0.4235	0.4206
JNLP	0.4126	0.4341	0.3931
SIL	0.3871	0.4186	0.3600
mezza	0.3289	0.3161	0.3429
INTIT	0.3165	0.3249	0.3086
DU	0.3141	0.2945	0.3366
FLNLP	0.2935	0.2619	0.3337
UA	0.2666	0.2038	0.3851
UCD-CS	0.2645	0.2480	0.2834
JUNLLP	0.2525	0.2193	0.2977
74688	0.2346	0.2130	0.2611
UB2026	0.2233	0.2338	0.2137
Sach	0.2231	0.1631	0.3531
BJPWH	**0.2101**	**0.1510**	**0.3451**
ABAI	0.1771	0.1596	0.1989
AIIRLab	0.1516	0.2642	0.1063
bosch	0.1466	0.0892	0.4103
KeioAndrewShin	0.1383	0.1527	0.1263
CityUMO	0.0000	0.0000	0.0000
ClaUSurf	0.0000	0.0000	0.0000
NIT26	0.0000	0.0000	0.0000
NRCBMU	0.0000	0.0000	0.0000

Domain-adapted models: Our dense retrieval component used a general-purpose sentence encoder. Models pre-trained or fine-tuned on legal corpora may improve semantic matching quality and overall retrieval performance.

6 Conclusion

We presented the BJPWH team's approach to COLIEE 2026 Task 1. Our system combines BM25 lexical retrieval with BERT dense retrieval, fused via weighted RRF, and enforces temporal precedent constraints through year-based filtering. Systematic parameter tuning identified an unusually high BM25 saturation parameter ($k_1 = 4.5$) as optimal for the long, terminology-rich nature of legal case text.

The central finding is that applying year filtering selectively—to BERT results only— outperforms filtering both retrievers. We hypothesize that BM25 lexical matching is naturally conservative with respect to temporal ordering, while BERT's semantic similarity can bridge across time periods in ways that introduce invalid retrievals. Year filtering corrects these BERT errors without removing the diverse early-precedent coverage that BM25 provides.

BJPWH3 achieved F1=0.2101 on the official test set, ranking 14th among 22 teams. The gap to the top systems (NOWJ: F1=0.4220, JNLP: F1=0.4126) reflects the absence of neural reranking and LLM-based components in our pipeline, both of which have become standard in leading COLIEE submissions. Future work should explore cross-encoder reranking, query expansion, and LLM-based summarization as the next steps toward a competitive pipeline.

Acknowledgments

This research was conducted as part of the COLIEE 2026 workshop at the University of Alberta.

References

[1] E. Baek, J. Dai, H. M. Q. Hasan, Y. Kim, H. Babiker, M. Y. Kim, and R. Goebel. 2025. From tf-idf to instruction-tuned llms: Hybrid legal reasoning systems for coliee 2025. In *Workshop of the Twelfth Competition on Legal Information Extraction/Entailment (COLIEE'2025) in the 21th International Conference on Artificial Intelligence and Law (ICAIL).*

[2] Gordon V. Cormack, Charles L. A. Clarke, and Stefan Büttcher. 2009. Reciprocal Rank Fusion outperforms Condorcet and individual rank learning methods. In *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval.* ACM, Boston, MA, USA, 758–759. doi:10.1145/1571941.1572114

[3] Rohan Debbarma, Pratik Prawar, Abhijnan Chakraborty, and Srikanta Bedathur. 2023. IITDLI: Legal Case Retrieval Based on Lexical Models. In *Proceedings of the Workshop of the Tenth Competition on Legal Information Extraction/Entailment (COLIEE 2023), ICAIL 2023.*

[4] Elastic. 2024. Elasticsearch. https://www.elastic.co/elasticsearch. Version 8.11.3.

[5] Randy Goebel, Yoshinobu Kano, Mi-Young Kim, Juliano Rabelo, Ken Satoh, and Masaharu Yoshioka. 2023. Summary of the Competition on Legal Information Extraction/Entailment (COLIEE) 2023. In *Proceedings of the Nineteenth International Conference on Artificial Intelligence and Law (ICAIL 2023).* ACM, Braga, Portugal. doi:10.1145/3594536.3595176

[6] Randy Goebel, Yoshinobu Kano, Mi-Young Kim, Juliano Rabelo, Ken Satoh, and Masaharu Yoshioka. 2024. Overview of Benchmark Datasets and Methods for the Legal Information Extraction/Entailment (COLIEE) 2024. In *New Frontiers in Artificial Intelligence: JSAI-isAI 2024 (Lecture Notes in Computer Science, Vol. 14741).* Springer, 109–124.

[7] Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2021. Billion-scale Similarity Search with GPUs. *IEEE Transactions on Big Data* 7, 3 (2021), 535–547. doi:10.1109/TBDATA.2019.2921572

[8] Yoshinobu Kano, Randy Goebel, Mi-Young Kim, Calum Kwan, Ken Satoh, Hiroaki Yamada, and Masaharu Yoshioka. 2026. An Overview of the COLIEE 2026 Competition: Legal Case Law and Statute Law Information Retrieval and Entailment. In *Proceedings of the 21st International Conference on Artificial Intelligence and Law (ICAIL 2026).*

[9] Yoshinobu Kano, Randy Goebel, Mi-Young Kim, Juliano Rabelo, Ken Satoh, and Masaharu Yoshioka. 2025. An Overview of the COLIEE 2025 Competition: Legal Case Law and Statute Law Information Retrieval and Entailment. In *Proceedings of the Twentieth International Conference on Artificial Intelligence and Law (ICAIL 2025).* ACM. doi:10.1145/3769126.3785016

[10] Haitao Li, Qingyao Ai, Jia Chen, Qian Dong, Yueyue Wu, Yiqun Liu, Chong Chen, and Qi Tian. 2023. SAILER: Structure-aware Pre-trained Language Model for Legal Case Retrieval. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval.* ACM, 1035–1044. doi:10.1145/3539618.3591761

[11] Haitao Li, Weihang Su, Changyue Wang, Yueyue Wu, Qingyao Ai, and Yiqun Liu. 2023. THUIR@COLIEE 2023: Incorporating Structural Knowledge into Pre-trained Language Models for Legal Case Retrieval. In *Workshop of the tenth competition on legal information extraction/entailment (COLIEE'2023) in the 19th international conference on artificial intelligence and law (ICAIL).*

[12] Antoine Louis, Gerasimos van Dijck, and Gerasimos Spanakis. 2022. LeiBi at COLIEE 2022: Aggregating Tuned Lexical Models with a Cluster-Driven BERT-Based Model for Case Law Retrieval. In *Proceedings of the Workshop of the Ninth Competition on Legal Information Extraction/Entailment (COLIEE 2022), JURISIN 2022.*

[13] The Hai Nguyen, Khac Vu Hiep Nguyen, Ngoc Anh Trang Pham, Ngoc Minh Nguyen, Hoang An Trieu, Nguyen Khang Le, Dinh Truong Do, and Le Minh Nguyen. 2025. JNLP at COLIEE 2025: Hybrid Large Language Model-based Framework for Legal Information Retrieval and Entailment. In *Proceedings of the Workshop of the Twelfth Competition on Legal Information Extraction/Entailment (COLIEE 2025), ICAIL 2025.*

[14] Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP).* Association for Computational Linguistics, Hong Kong, China, 3982–3992. doi:10.18653/v1/D19-1410

[15] Stephen Robertson and Hugo Zaragoza. 2009. The Probabilistic Relevance Framework: BM25 and Beyond. *Foundations and Trends in Information Retrieval* 3, 4 (2009), 333–389. doi:10.1561/1500000019

[16] Thi-Hai-Yen Vuong, Hai-Long Nguyen, Tan-Minh Nguyen, Hoang-Trung Nguyen, Thai-Binh Nguyen, and Ha-Thanh Nguyen. 2023. NOWJ at COLIEE 2023: Multi-Task and Ensemble Approaches in Legal Information Processing. In *Proceedings of*

the Workshop of the Tenth Competition on Legal Information Extraction/Entailment (COLIEE 2023), ICAIL 2023.

[17] An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. 2024. Qwen2.5 Technical Report. *arXiv preprint arXiv:2412.15115* (2024).

A Preliminary Experiments

This appendix describes two approaches we explored during development but ultimately did not include in our final submissions, either because they proved too time-consuming to run at scale or because they did not improve over our baseline.

A.1 SAILER Reranking

We experimented with SAILER [10] as a reranking step applied on top of BM25-retrieved candidates. Despite SAILER's strong results in prior COLIEE editions, it consistently underperformed the BM25 baseline across all metrics on our dataset: R@5 (−12.5%), R@10 (−7.4%), R@20 (−4.1%), and MAP (−19.7%).

We attribute this underperformance to the absence of domain-specific fine-tuning on the COLIEE corpus, which likely limited SAILER's ability to transfer its structural priors to this data distribution. We therefore excluded this approach from our final submissions.

A.2 LLM-Based Case Summarization

We planned to use Qwen2.5-14B [17] to generate structured summaries of legal cases (introduction, facts, provisions, analysis, conclusion), hypothesizing that BM25 retrieval over concise summaries would outperform retrieval over full-length cases. Initial outputs were promising in quality, but processing speed proved prohibitive: even on two 15GB GPUs, the pipeline processed approximately 15 cases per 10 hours. The bottleneck was CPU-bound tokenization of long inputs, compounded by communication overhead from sharding the model across two GPUs. With 7,708 training cases to process, we could not complete the dataset before the submission deadline. Summarization-based retrieval remains a promising direction for future work.

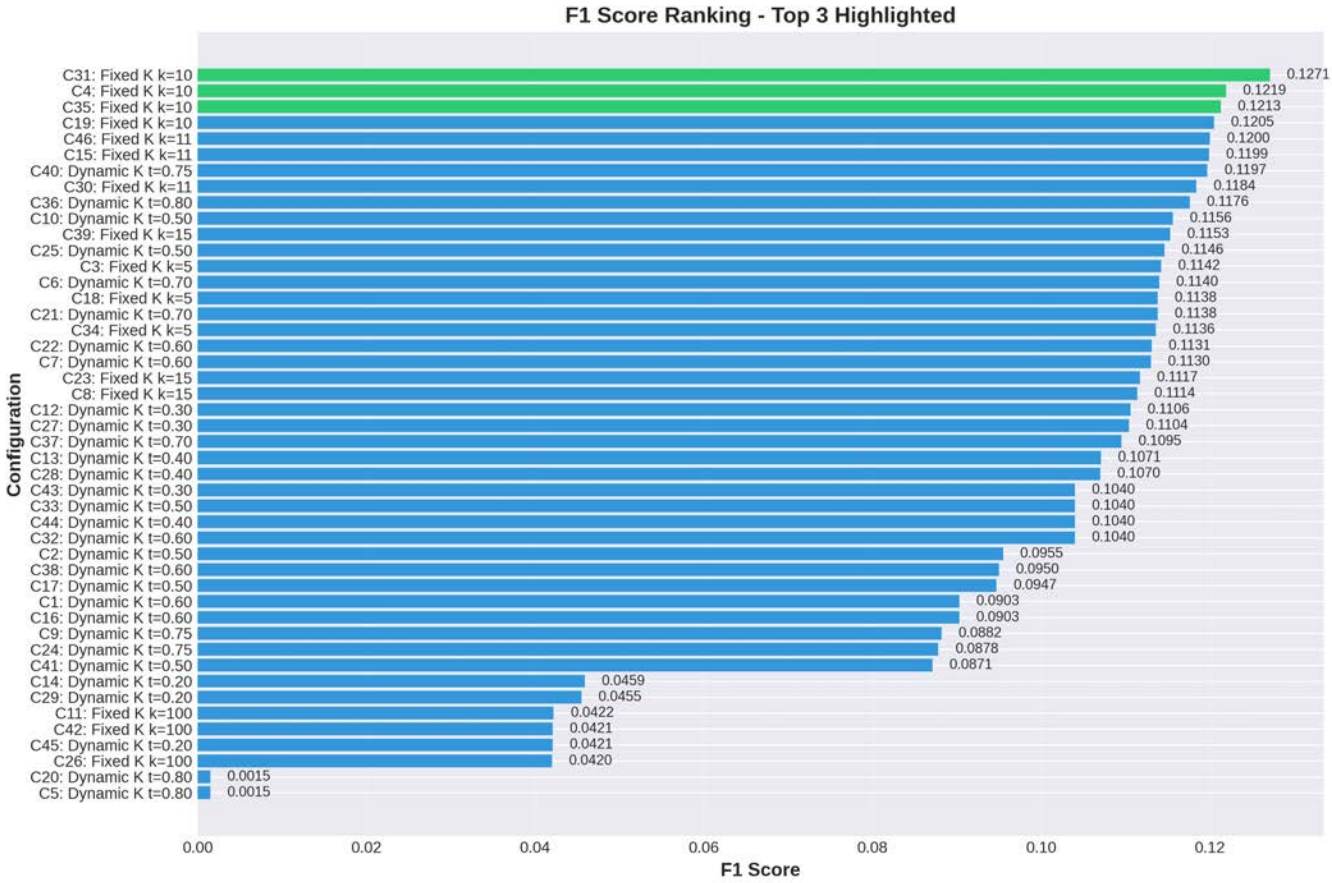


Figure 4: F1 scores across 45 submission strategy configurations on training data. Fixed $K = 10$ consistently outperformed both smaller K values and dynamic threshold-based strategies.

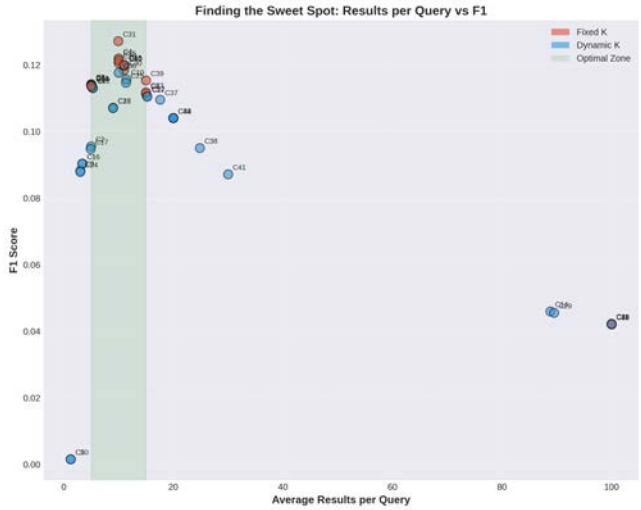


Figure 5: Relationship between number of results per query and F1 score. The optimal zone of 0–15 results per query is highlighted; Fixed $K = 10$ falls squarely in this range.

AIIRLab at COLIEE 2026: Fusion of Instruction-Tuned LLMs and Cross-Encoders for Legal Case Entailment

Nicholas Largey
nicholas.largey@maine.edu
University of Southern Maine
Portland, Maine, USA

Ellis Fitzgerald
ellis.fitzgerald@maine.edu
University of Southern Maine
Portland, Maine, USA

Behrooz Mansouri
behrooz.mansouri@maine.edu
University of Southern Maine
Portland, Maine, USA

Abstract

The Competition on Legal Information Extraction and Entailment (COLIEE) provides a standardized benchmark for evaluating large language models across four distinct tasks: Legal Case Retrieval (Task 1), Legal Case Entailment (Task 2), Statute Retrieval (Task 3), and Legal Textual Entailment (Task 4). This paper presents the AIIR Lab's submissions to COLIEE 2026, focusing on optimizing LLMs for the long-context and high-precision requirements of the legal domain. For Tasks 1 and 2, which operate over Federal Court of Canada case law, systems employ Parameter-Efficient Fine-Tuning (PEFT) via Low-Rank Adaptation (LoRA), bi-encoder dense retrieval, and multi-stage reranking pipelines. Task 2 introduces the Fusion3 ensemble, combining fine-tuned LLaMA-3.1-8B, Qwen2.5-7B, and a DeBERTa cross-encoder via union voting, which achieves an F1 of 0.47 and ranks second among all participating teams. For Tasks 3 and 4, which target the Japanese Civil Code, systems leverage Hypothetical Document Embeddings (HyDE), multilingual dense retrieval, and ensemble strategies including Neuro-Symbolic (NeuSym), Cross-Lingual Chain-of-Thought (CrossCoT), and Architecturally Diverse (DivVote) voting ensembles. Results across all tasks reveal a fundamental trade-off between broad-coverage retrieval and high-precision reasoning, suggesting that the next generation of legal AI must move beyond semantic similarity toward systems capable of reliably distinguishing general rules from their procedural exceptions.

CCS Concepts

• **Applied computing** → **Law**; • **Information systems** → *Specialized information retrieval*.

Keywords

Legal Case Retrieval, Legal Entailment, Legal Language Processing

1 Introduction

Legal information processing remains a challenging setting for modern natural language processing systems because relevant evidence is often distributed across long documents expressed in highly specialized language [14]. Although large language models (LLMs) and retrieval-augmented pipelines have substantially improved semantic matching in open-domain tasks, legal applications continue to require much stricter evidence alignment and more reliable reasoning than topical similarity alone can provide [12, 17]. In particular, legal retrieval and entailment systems must often operate over lengthy judicial opinions and statutory provisions, where important signals may appear only in a few paragraphs or even in a single sentence.

The Competition on Legal Information Extraction and Entailment (COLIEE) [11, 13] provides a standardized benchmark for evaluating these capabilities across four distinct tasks: Legal Case Retrieval (Task 1), Legal Case Entailment (Task 2), Statute Retrieval (Task 3), and Legal Textual Entailment (Task 4). The Artificial Intelligence and Information Retrieval (AIIR) Lab from the University of Southern Maine participated in this competition, providing runs for all four tasks.

Tasks 1 and 2 operate over a corpus of Federal Court of Canada judicial decisions. Task 1 requires identifying noticed cases, cases explicitly cited in an original decision, from a corpus in which all such citations have been redacted. Task 2 proceeds further by requiring the identification of specific paragraphs within a noticed case that entail a given decision fragment, demanding fine-grained semantic matching and logical inference at the paragraph level. Tasks 3 and 4 shift the focus to the Japanese Civil Code. Task 3 jointly requires statute retrieval and binary entailment prediction, where a correct answer must be supported by an appropriately retrieved article. Task 4 isolates the entailment component by providing gold-standard relevant articles at test time, enabling a more direct evaluation of legal reasoning capabilities independent of retrieval performance.

Across all tasks, the systems presented in this paper implement a range of strategies for these challenges. For case-based tasks, these include Parameter-Efficient Fine-Tuning (PEFT) via Low-Rank Adaptation (LoRA), bi-encoder dense retrieval with hard-negative mining, and multi-stage reranking using both cross-encoders and generative LLMs. For statute-based tasks, strategies include Hypothetical Document Embeddings (HyDE), multilingual dense retrieval, pairwise LLM reranking, and ensemble voting architectures. For Task 2, the Fusion3 system, a union ensemble of fine-tuned LLaMA-3.1-8B, Qwen2.5-7B, and a DeBERTa cross-encoder, achieves an F1 of 0.47 and ranks second among all participating teams. For Task 4, three ensemble configurations are compared: a Neuro-Symbolic ensemble (NeuSym) that incorporates symbolic backward-chaining inference, a Cross-Lingual Chain-of-Thought ensemble (CrossCoT) that augments a multilingual baseline with QLoRA-trained chain-of-thought reasoning, and an Architecturally Diverse ensemble (DivVote) that introduces a BART-large sequence-to-sequence component to mitigate encoder-only bag-of-words failure modes.

COLIEE 2026, June 12, 2026, Singapore,

Table 1: Dataset statistics for training and evaluation splits.

Split	Queries	Corpus Size	Rel. Pairs	Avg. Rel./Query
Train	2,001	7,708	8,251	4.12
Test	400	1,848	1,750	4.38

Findings across tasks show a consistent tension between broad-coverage retrieval and high-precision reasoning. Generative rerankers improve semantic matching but face recall bottlenecks when critical precedents are buried in long documents or cut off by context-window constraints. Pointwise reranking pipelines achieve high recall but are susceptible to score saturation and semantic confusion between adjacent statutory provisions. Symbolic decomposition and chain-of-thought reasoning offer targeted improvements for nested exception handling and fine-grained qualitative distinctions, but their benefits are task- and instance-specific. The remainder of this paper is organized as follows: Sections 2 through 5 each describe one of the four tasks, presenting the task definition, proposed systems, and evaluation results in turn, followed by a concluding discussion in Section 6.

2 Task 1: Legal Case Retrieval

In this section, we first review the legal case retrieval task and then explain our proposed models, followed by evaluation results.

2.1 Task Definition

Given a query case q, the task is to retrieve a ranked list of noticed cases from a fixed corpus. A case is considered noticed if it is explicitly referenced in the original judicial decision. However, for a realistic retrieval setting, all such references are removed (redacted) from the query case text.

This task uses a corpus derived from Federal Court of Canada case law, where both query and candidate cases are provided as individual text files. The dataset is divided into:

(1) *Training set:* Includes query cases along with annotated noticed cases, provided as a JSON mapping from query IDs to lists of relevant case IDs.
(2) *Test set:* Contains only query cases; systems must predict the corresponding noticed cases.

System performance is evaluated using standard information retrieval metrics:

(1) Precision: Proportion of retrieved cases that are relevant.
(2) Recall: Proportion of relevant cases that are successfully retrieved.
(3) F1-score: Harmonic mean of precision and recall.

Micro-averaged evaluation is adopted, where counts of true positives, false positives, and false negatives are aggregated across all queries before computing the metrics. The distribution of queries, corpus size, and relevance labels is summarized in Table 1.

2.2 Proposed Models

Preprocessing Step. Two of our systems, QWEN and LLaMA, use the same cleaning step. Raw case files from the Federal Court of Canada were passed through a two-stage pre-processing pipeline before any downstream analysis. In the first, deterministic stage, each document underwent a sequence of rule-based cleaning steps; line endings were normalized, procedural headers appearing before the first numbered paragraph were stripped, and suppressed-fragment placeholders along with a curated list of bilingual boilerplate tokens (e.g., translation notices, emphasis markers, omission indicators) were removed. Because the corpus is officially bilingual, paragraphs whose sentences were in French, detected via a lexicon of common French function words and legal terms combined with a diacritic heuristic, were discarded, retaining only the English-language content. Remaining whitespace artifacts, punctuation, and excess blank lines were then normalized to produce clean, consistently formatted plain text.

In the second stage, each cleaned document was processed by the *LLaMA-3.1-8B-Instruct* [3] model to extract only legally important content. Because many case files exceeded the model's context window, documents were split into overlapping chunks of approximately 6,000 characters with a 400-character overlap; boundaries were snapped to paragraph breaks to avoid mid-sentence cuts. Each chunk was submitted independently to the model with explicit positional context (chunk i of N), instructing it to retain legal reasoning, factual findings, statutory interpretation, and procedural history while discarding citation lists, counsel listings, and redundant headings. Chunks flagged by the model as containing no legally significant content were silently skipped. When a document produced multiple chunk extractions, a final LLM call merged them into a single coherent prose summary, eliminating redundancy introduced by the overlapping windows.

Our ParaNemo system utilizes a similar pre-processing strategy, but is single-staged and fully deterministic. The documents passed to the system are split into paragraphs using the ordered lists denoted by '[x]' in each document in the dataset. Then, each paragraph is tokenized with spaCy[1] for the removal of extra whitespace, stop-words, and punctuation. The French language is not removed, translated, or summarized.

After pre-processing, we proposed three distinct systems as follows:

QWEN. The proposed system adopts a retrieve-then-rerank architecture. In the first stage, a bi-encoder model *all-mpnet-base-v2* [16] is fine-tuned on legal case pairs. Training pairs are constructed by splitting each case document into paragraphs and using BM25 to align passages from query cases with their known relevant (noticed) cases, yielding semantically grounded positive pairs at the passage level. The model is fine-tuned with Multiple Negatives Ranking loss and validated via embedding similarity on a held-out 10% split, with early stopping to prevent overfitting. At inference, each document is encoded at the paragraph level, and queries are matched against the entire corpus using a MaxSim scoring strategy, taking the maximum cosine similarity across all query-passage and candidate-passage pairs, which allows the model to reward partial but topical overlap without requiring full-document alignment. The top candidates from this stage are then passed to the re-ranking stage.

[1] https://spacy.io/

In the second stage, a Qwen2.5-7B-Instruct [4] model acts as a pointwise relevance judge over the top-10 candidates retrieved per query. For fine-tuning, the model is adapted using QLoRA via Unsloth,[2] with LoRA adapters (rank 16, alpha 32) applied to all attention and feed-forward projection layers. Training data consists of positive (query, relevant case) pairs labeled with high confidence scores, and hard negatives mined using BM25 [15] to retrieve lexically similar but non-relevant cases, resulting in a 2:1 negative-to-positive ratio per query. The model is trained to produce a binary relevance judgment along with a continuous confidence score, using the same prompt format at both training and inference to eliminate distributional mismatch. At inference, both the pre-trained and fine-tuned Qwen models independently score each candidate, producing two ranked lists of relevant cases per query. These lists are then combined using CombMNZ, a data fusion method that rewards documents appearing in multiple result sets by multiplying their summed scores by the number of lists in which they appear. A fallback mechanism is used to ensure at least one result is returned per query in cases where the reranker rejects all candidates.

LLaMA. We propose a two-stage retrieve-and-rerank pipeline for the long-context requirements of the legal domain. In the first stage, a bi-encoder based on BGE-M3 [1] is employed for dense retrieval, selected for its support of input sequences up to 8,192 tokens. The model is fine-tuned on the COLIEE 2026 dataset using *Cached Multiple Negatives Ranking Loss* (InfoNCE), with five hard negatives per positive instance to improve discriminative capability. At inference time, query and document embeddings are precomputed, and candidates are ranked via cosine similarity; the top 50 documents per query are retained for downstream re-ranking.

In the second stage, LLaMA 3.1 8B Instruct [3] is used as a cross-encoder to refine the candidate set by framing re-ranking as a constrained generation task. The model is adapted using Parameter-Efficient Fine-Tuning with Low-Rank Adaptation (LoRA), with hyperparameters $r = 16$, $\alpha = 32$, and dropout 0.05. Training is performed via supervised token-level learning, prompting the model to act as a legal expert and produce a binary relevance judgment ("Yes"/"No") indicating whether a candidate case supports the query. At inference, relevance scores are derived from the softmax-normalized logit of the "Yes" token. Candidates from the top-50 pool are re-scored, and those with $P(\text{Yes}) \geq 0.75$ are retained, subject to a final cutoff of five documents per query to prioritize precision.[3]

ParaNemo. The proposed system adopts a similar retrieve-then-rerank architecture as the QWEN system, but diverges slightly due to the differing pre-processing methods. In the retrieval stage, BM25 [15] is employed as the lexical baseline to retrieve an initial set of 100 candidate cases. The index is constructed by concatenating all paragraphs within each case from the preprocessed corpus. At query time, each preprocessed query paragraph is independently scored against the index. For a given query–case pair, scores across all query paragraphs are aggregated by taking the maximum value. The resulting document–score mappings are then ranked and forwarded to subsequent stages.

In the second stage, a fine-tuned bi-encoder (as used in the QWEN system) is used for semantic retrieval to re-rank and prune the BM25 candidates to the top-10. Similarity is computed between query and case paragraphs using a pairwise similarity matrix over their preprocessed representations. For each query–case pair, the maximum paragraph-level similarity score is used as the aggregate relevance signal, based on the intuition that a case may be "noticed" due to a single highly relevant paragraph. Alternative aggregation strategies, such as mean and top-k mean pooling, were also considered. The resulting similarity scores are used to rank the candidates, and the top 10 are forwarded to the final re-ranking stage.

In the final stage, a cross-encoder, *llama-nemotron-rerank-1b-v2* [2], is used to re-rank the candidates produced by the bi-encoder. The model supports inputs of up to 8,192 tokens and is multilingual, which is advantageous given the characteristics of the corpus. Prior to re-ranking, all paragraphs within each preprocessed case are concatenated to reduce the number of inference calls and to better utilize the model's input capacity. This design also allows the model to capture contextual dependencies that may span multiple paragraphs. The model supports prompt-based re-ranking, for which we use the following template:

```
case:{query}

potentially-relevant-case:{passage}
```

After re-ranking, the top 5 cases are retained as the final output to mitigate the impact of lower-ranked, potentially noisy candidates.

Table 2: Results of AIIR Lab Runs for Task 1 on the COLIEE 2026 Test Set (400 Queries). $*$ indicates corrected post-run.

Run Name	F1	Precision	Recall
QWEN	0.1516	**0.2642**	0.1063
ParaNemo	0.1125	0.1055	0.1206
LLaMA$*$	**0.2489**	0.2370	**0.3396**

2.3 Evaluation Results

Table 2 shows our systems' results for Task 1, with a post-run-corrected result from the LLaMA system. The LLaMA model achieves the highest F1 Score and recall, significantly higher than other models, using a paired Student's t-test ($p < 0.05$). The QWEN model's precision is significantly higher than the two other models, with a precision above 0.5 for nearly 30% of queries. However, this model had recall of 0 for 65% of queries, whereas LLaMA failed to retrieve relevant results for 35% of test queries.

Looking at the results, The performance gap between the two LLM-based systems shows how each model prioritizes different aspects of legal similarity. The LLaMA-based pipeline is more effective at retrieving relevant cases when the connection was based on broad doctrinal frameworks rather than specific statutory anchors. In Queries *000440* and *022674*, for example, the relevant cases share overarching principles, such as the functional integration test for federal jurisdiction or requirements for representative standing, which are often expressed through varied terminology. In Query *000440* and Case *060605*, the jurisdictional analysis is framed through very different factual backgrounds, making LLaMA's document-wide alignment more successful than localized

[2]https://github.com/unslothai/unsloth
[3]During the competition, this system was run on a wrong input file, and the corrected run is reported in the evaluation section.

Table 3: Comparison of ParaNemo re-ranking versus the standalone fine-tuned bi-encoder candidate set.

Retrieval Pipeline Stage	F1	Precision	Recall
ParaNemo (Final Re-rank)	0.1125	0.1055	0.1206
Fine-tuned Bi-encoder (Top-10)	**0.1534**	**0.1103**	**0.2520**

matching. By using the 8,192-token context window of the BGE-M3 encoder, LLaMA captures reasoning patterns distributed throughout a judgment. This capability allows it to identify candidates like Query *076750*, where the abuse of process doctrine is articulated through disparate procedural facts.

On the other hand, the Qwen-based ensemble achieved higher precision by prioritizing paragraph-level alignment. This approach effectively filters out the noise that can lead to false positives in document-wide models. The LLaMA pipeline's use of character-level truncation and a high precision threshold of 0.75 makes it vulnerable to missing cases when key principles are cut off or buried in long factual narratives. Qwen's success on Queries *025964*, *080149*, and *094642* instead stems from its MaxSim strategy and CombMNZ fusion, which isolate specific statutory points like Regulation 15 of the IRP Regulations or the Section 6(5) factors of the Trade-marks Act. By rewarding candidates that appear consistently across both pre-trained and fine-tuned lists, the Qwen ensemble maintains high precision for cases that require identifying narrow procedural links.

The performance of the ParaNemo system, summarized in Table 3, shows that the final cross-encoder re-ranking stage degraded retrieval metrics compared to the candidate set produced by the bi-encoder. Specifically, the fine-tuned bi-encoder achieved a higher F1 (0.1534) and Recall (0.2520) than the integrated ParaNemo pipeline. This suggests that while the *llama-nemotron-rerank-1b-v2* model provides a large 8,192-token context window and multilingual support, the transition from paragraph-level matching to full-document concatenation introduced new bottlenecks.

For instance, in query *028326*, ParaNemo successfully retrieved the relevant Case *091015* by capturing dependencies across paragraph boundaries that the bi-encoder missed. However, this benefit was often outweighed by truncation issues. In query *032123*, the re-ranker demoted a gold-label case that the bi-encoder had correctly identified. Because that document was exceptionally long (~65K characters), key legal points likely fell outside the 8,192-token limit, making them invisible to the cross-encoder while remaining accessible to the bi-encoder's segment-wise matching. These results suggest that document-level concatenation can dilute the signal in long-form legal texts. The bi-encoder's approach is more robust to document length because it isolates high-similarity segments, whereas the cross-encoder can be overwhelmed by factual noise or limited by its input capacity. Additionally, a mismatch exists between the English-centric bi-encoder and the multilingual cross-encoder, potentially causing the system to prune non-English candidates too early.

3 Task 2: Legal Case Entailment

This section reviews the legal case entailment task and provides explanations of our proposed systems and evaluation results.

3.1 Task Definition

While Task 1 focuses on identifying relevant (noticed) cases, this task aims to determine which specific parts of those cases justify a legal conclusion. Given a query case q, a noticed case c (assumed to be relevant to q), and a decision fragment d extracted from q, the goal is to identify one or more paragraphs in c that entail the decision fragment d. Unlike full case outcome prediction, the notion of "decision" here refers to a localized judicial conclusion (i.e., a fragment of reasoning), rather than the final verdict of the case. Systems must therefore perform semantic matching and logical inference between the decision fragment and candidate paragraphs. The dataset uses the same underlying corpus of Federal Court of Canada case law as Task 1, but is organized differently to support paragraph-level reasoning. The evaluation metrics are also the same as the ones used for Task 1.

3.2 Proposed Models

Fusion3. This model ensembles three different models as shown in Figure 1. Two instruction-tuned generative models are fine-tuned using QLoRA via Unsloth: *Meta-Llama-3.1-8B-Instruct* [3] and *Qwen2.5-7B-Instruct* [4], both loaded in 4-bit quantization. LoRA adapters with rank $r = 16$, $\alpha = 32$, and dropout 0.05 are attached to all attention and feed-forward projection layers.

Each model is trained for 3 epochs with a per-device batch size of 2 and 4 gradient accumulation steps (effective batch size of 8), a learning rate of 2×10^{-4} with a cosine schedule and 10% linear warmup, and early stopping with a patience of 5 evaluation steps. Training instances are constructed as three-turn chat conversations: a system message describing the task, a user turn presenting the decision fragment and candidate paragraph, and an assistant turn containing the binary label. The system prompt used is:

> *"You are a legal expert. Determine whether the PARAGRAPH below logically entails (supports / is the basis for) the DECISION FRAGMENT. Answer with exactly one word: 'YES' if it entails, or 'NO' if it does not."*

The user turn is structured as follows:

> *### Decision Fragment*
> *<fragment truncated to 800 characters>*
>
> *### Paragraph*
> *<paragraph truncated to 1000 characters>*
>
> *Does the paragraph entail the decision fragment?*

Gold paragraphs are labeled *YES* and up to three randomly sampled non-relevant paragraphs per positive serve as hard negatives (labeled *NO*), yielding a 3:1 negative-to-positive ratio. At inference, rather than generating a token and thresholding its value, we perform a single forward pass and compute a continuous entailment score as $\log P(\mathit{YES}) - \log P(\mathit{NO})$ over the logits at the first generated position, taking the maximum log-probability across all single-token surface forms of each label (e.g., *YES*, *yes*, *Yes*, with and without a leading space).

The third component is a fine-tuned DeBERTa cross-encoder model [10] (described later in this section). At inference, the entailment class probability (softmax index 0 from the three-class NLI head) is used as the relevance score, and all paragraphs exceeding a threshold of 0.9 are returned as predictions, with a fallback to the

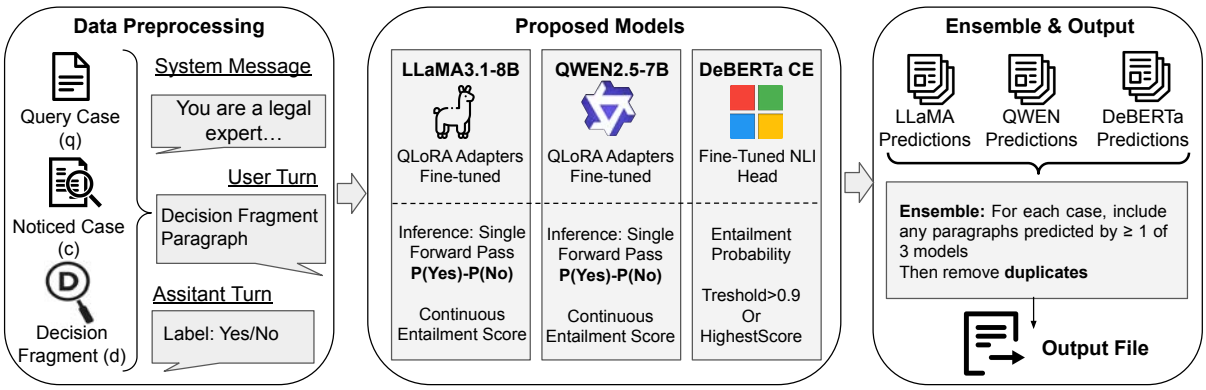


Figure 1: Fusion3 System Architecture. The task inputs (q,c,d) are processed, constructing three-turn chat instances with a 3:1 negative-to-positive ratio for fine-tuning. The Fusion3 ensemble consists of three models: fine-tuned Llama-3.1-8B and Qwen2.5-7B (each using QLoRA and 4-bit quantization), and a fine-tuned DeBERTa Cross-Encoder. The generative models compute a continuous entailment score via a log-probability difference, while DeBERTa uses a threshold of 0.9 with a highest-scoring fallback. The predictions from all three models are combined using a Union Ensemble, followed by a deduplication step to generate the final output.

highest-scoring paragraph when no candidate clears the threshold. The predictions from all three models are combined via a union ensemble: for each case, any paragraph predicted as entailing by *at least one* of the three models is included in the final output. Duplicate predictions across models are deduplicated, and results are written in the standard task submission format.

Fusion2. The proposed system utilizes a heterogeneous ensemble architecture specifically engineered to evaluate entailment between legal decisions and candidate supporting paragraphs. This pipeline processes every (decision, paragraph) pair in parallel across multiple GPUs using a late-fusion approach to aggregate model outputs. To capture complementary semantic features, the ensemble integrates two large language models with distinct training paradigms. The first component, Model 1, is a fine-tuned LLaMA3.1-8b-Instruct [3] model previously optimized via Low-Rank Adaptation (LoRA) for case retrieval tasks. This model is repurposed for entailment through specialized prompt framing that instructs it to act as a legal expert, performing binary classification by answering whether a paragraph directly supports a legal decision. The resulting relevance score is derived from the logit difference of the 'Yes' and 'No' tokens at the final sequence position.

To provide a robust baseline and mitigate over-optimization, Model 2 incorporates Qwen3-14B [5] in a zero-shot configuration. To ensure deterministic classification and immediate response, Chain-of-Thought (CoT) reasoning is explicitly disabled by injecting a closed-tag sequence (*<think>\n\n</think>\n*) immediately following the assistant turn. This forces the model to generate the classification token as the initial output, from which the $P(Yes)$ logit-ratio is similarly extracted.

To account for the differing logit scales and distributions of the two models, we implement a normalized fusion strategy. During parallel scoring, both models independently evaluate all candidate paragraphs on separate GPUs. For each case, the raw score vector (s) for each model is mapped to the range $[0, 1]$ using min-max scaling for a specific paragraph p. The final ensemble score is then calculated as the unweighted mean of these normalized probabilities. Paragraphs meeting a threshold of $P \geq 0.75$ are selected for the final submission. In instances where no paragraph clears this threshold, the system defaults to a 'Top-1' fallback strategy, returning the single highest-ranked paragraph to ensure a minimum result set and complete coverage.

Deberta. We fine-tune a DeBERTa-v3-base[10] cross-encoder to identify which paragraph(s) within a noticed case entail a given decision fragment from the query case. The model is initialized from a pre-trained three-class NLI checkpoint (entailment / neutral / contradiction) and fine-tuned end-to-end on case-specific paragraph pairs, where gold paragraphs are labeled as entailment and randomly sampled non-relevant paragraphs serve as negatives at a 5:1 ratio. Each training instance is formed by truncating the decision fragment to 200 words and the candidate paragraph to 280 words, keeping the combined input within the model's 512-token limit. The model is trained for up to 40 epochs using a cosine learning rate schedule with 10% linear warm-up, and the best checkpoint is selected by MRR@10 on a held-out 10% validation split evaluated via a custom entailment re-ranking evaluator. At inference, raw logits from the cross-encoder are converted to probabilities via softmax, and the entailment class probability (index 0) is used as the continuous relevance score; all paragraphs exceeding a confidence threshold of 0.9 are selected as predictions, with a fallback to the highest-scoring paragraph when no candidate clears the threshold.

3.3 Evaluation Results

Table 4 shows our systems' performance for Task 2. Fusion3 run provided the highest effectiveness, and was ranked as the second run in the COLIEE 2026 among the participating teams. Fusion3 achieved the highest F1 score (0.48), significantly outperforming Fusion2 (F1=0.42) and DeBERTa (F1=0.35) based on a paired Student's t-test ($p < 0.05$). Fusion3 had a precision above 0.5 for 70%

Table 4: Results of AIIR Lab Runs for Task 2 on the COLIEE 2026 Test Set (100 queries).

Run	F1	Precision	Recall
Fusion3	**0.4706**	0.5120	**0.4354**
Fusion2	0.4271	0.4257	0.4286
Deberta	0.3333	**0.5750**	0.2347

of queries and a recall above 0.5 for nearly half of the queries. DeBERTa model, due to its strict thresholding policy, provided the highest precision among our runs, with a precision of 1 for more than half of the queries.

Fusion3's results indicate that the union ensemble of instruction-tuned generative models combined with the DeBERTa-v3-base[10] cross-encoder provides a more comprehensive coverage of the entailment space. While the performance of Fusion2, which utilized a normalized fusion of LLaMA3.1-8b-Instruct[3] and Qwen3-14B[5], resulted in a precision score significantly lower than Fusion3's, but was able to maintain a similar recall level. This gap in scores may be attributed to the differing ensemble strategies where Fusion3's union approach of including any prediction appears better suited for the task's retrieval requirements rather than the unweighted mean of normalized probabilities used in Fusion2.

As shown when analyzing the submission for query with ID *947*, where the ground truth includes paragraphs *010*, *011*, Fusion2 returned a single result of *011* against Fusion3's result: *007, 010, 011*. Both systems returned *011*, which satisfies the clearly stated issues of "CAIPS notes" and the applicant's appeal on "humanitarian and compassionate considerations" for residency in Canada. However, Fusion3 also returned *010* which addresses the more subtle issue of "the best interests of the child" raised, and cites "Baker v. Canada (Minister of Citizenship and Immigration), [1999] 2 S.C.R. 817" which is also cited in the query. The citation of "Baker v. Canada (Minister of Citizenship and Immigration), [1999] 2 S.C.R. 817" is also present in *007*, which would provide an explanation for why it, too, was returned by the Fusion3 system's union approach.

The Fusion3 model's zero-precision failures on these instances can be attributed to two main issues. First, the union ensemble's design trades precision for recall; when any component model incorrectly predicts entailment due to shallow keyword overlap, erroneous paragraphs propagate to the final output unchecked. Second, the 800–1000 character truncation limits combined with the capacity of 7–8B parameter models to perform single-forward-pass entailment judgment are insufficient for the complex legal reasoning these queries demand. For instance, Query 937 requires weighing conflicting quantitative valuations of a seized vessel across multiple evidence sources, and Query 987 demands verification of a three-part cumulative confidentiality test; multi-step dependencies that small generative models struggle to resolve. Similarly, Queries 950 and 980 require normative and procedural inference (e.g., applying the *Khalil* exception or distinguishing policy categories for migrant healthcare), causing models to retrieve topically related but logically insufficient paragraphs.

4 Task 3: Statute Retrieval

This section provides details on task 3, providing details on the task, our proposed systems along with the evaluation results.

4.1 Task Definition

This task focuses on the joint problem of statute retrieval and textual entailment for legal question answering. Given a natural language query, the objective is to predict a binary decision (Yes/No) and retrieve a set of relevant statute articles that justify the answer. An article is considered relevant if its content entails the answer to the query. As such, the task requires systems to perform integrated reasoning and retrieval, where correct predictions must be supported by appropriate legal evidence rather than relying solely on classification.

The corpus used in this task is derived from Japanese civil law and follows a structure similar to Tasks 1 and 2, in that relevance relationships are explicitly provided in the training data. Each instance consists of a query paired with one or more statute articles and an associated entailment label. The statutory texts are provided in Japanese, and participants may optionally incorporate machine translation or multilingual modeling techniques. The training data includes both the relevant articles and the correct entailment labels; whereas the test data contains only queries. Systems must therefore automatically retrieve the relevant articles and infer the correct Yes/No decision without any human intervention or test-time adjustments.

Evaluation emphasizes both correctness and completeness of the predicted outputs. The primary metric is accuracy, defined as the proportion of queries for which the system predicts the correct entailment label and retrieves a sufficient set of relevant articles (i.e., achieving full recall for that query). In cases of ties, the F2-measure is used as a secondary metric, placing greater emphasis on recall over precision.

4.2 Proposed Models

JPGen. This system implements a Hypothetical Document Embedding (HyDE) pipeline that decomposes the retrieval and entailment task into two sequential stages: article generation followed by dense retrieval. The intuition behind this approach is that a query expressed in natural language may be semantically distant from the formal statutory language of the corpus, and generating an intermediate "hypothetical article" can bridge this gap by projecting the query into the distributional space of legal text.

In the first stage, a *Qwen2.5-7B-Instruct* [4] model is fine-tuned using Low-Rank Adaptation (LoRA) on query–article pairs drawn from the training set. The model is trained to generate a relevant statutory article given a legal query, using the system prompt: "You are a Japanese legal expert. Given a statement related to law, generate a legal article that supports or is related to it. Output only the article text, without any explanation or preamble." Training uses 4-bit quantization via the Unsloth framework, with a LoRA rank of 16 and a learning rate of 2×10^{-4}, and a cosine learning rate schedule over three epochs. An early stopping callback with a patience of five evaluation steps is applied to prevent overfitting, with the best checkpoint selected according to validation loss.

In the second stage, the generated articles are used as the retrieval query against the full statute corpus. Candidate articles are ranked using a fine-tuned bi-encoder based on *multilingual-e5-large* [8], and the top-5 retrieved articles are returned as the predicted relevant set. The entailment label is subsequently determined from the retrieved evidence.

JPLLM. This system uses a three pipeline that narrows the article collection down to a final set of relevant statutes. The first stage employs a fine-tuned bi-encoder based on *multilingual-e5-large* [8], a multilingual model that natively supports Japanese. Following the E5 training convention, all queries are prefixed with *"query: "* and all article passages are prefixed with *"passage: "* at both fine-tuning and inference time. The model is fine-tuned using Multiple Negatives Ranking Loss (MNRL) on positive (query, article) pairs extracted from the training labels, with in-batch negatives as the contrastive signal, for 20 epochs with a batch size of 8 and 10% linear warmup. Validation is performed using MRR@10 and NDCG@10 on a held-out 10% split, and the best checkpoint is retained. At inference, the entire article collection is encoded once and each query is matched against it via cosine similarity (implemented as a dot product over L2-normalised embeddings), retrieving the top-50 candidates per query.

The top-50 candidates from Stage 1 are re-ranked using a fine-tuned *Qwen2.5-7B-Instruct* [4] model operating as a pairwise comparator. The model is fine-tuned on the training set as a binary classifier to detect whether a given article is relevant to a query. At inference, for each query the model is presented with two candidate articles and asked to identify the more relevant one, responding with a single digit (*1* or *2*). The pairwise comparisons are structured as a selection-sort procedure: k passes are made over the candidate pool, each identifying the best remaining article via pairwise LLM calls, producing the top-20 ranked articles per query. All prompts are issued in Japanese to avoid language-switching overhead; the system prompt (translated from Japanese) instructs the model as follows:

> *"You are a legal document re-ranking system. Given a legal query and two articles, decide which article is more directly relevant to the query. Reply with only a single digit: 1 if Article 1 is more relevant, or 2 if Article 2 is equally or more relevant."*

The user turn presents the query followed by the two candidate articles and asks the model to respond with *1* or *2*. Up to three retries are performed on ambiguous outputs before defaulting to no swap.

The top-20 candidates from Stage 2 are passed through a final point-wise relevance filter using the same *Qwen2.5-7B-Instruct* [4] model. For each (query, article) pair, the model outputs a continuous relevance score in $[0, 1]$ on the first line, followed by a one-sentence justification on the second line. Prompts are again issued in Japanese; the system prompt (translated from Japanese) applies a strict relevance criterion:

> *"You are a strict legal relevance assessor in a civil law information retrieval system. An article is considered relevant only if it directly and specifically addresses the legal issue raised in the query. Articles that merely reference adjacent legal concepts or are only indirectly applicable should be judged as not relevant. Assign a relevance score between 0.0 and 1.0, where 1.0 means the article directly and completely addresses the query, and 0.0 means it is entirely unrelated. Apply a high relevance threshold: scores above 0.5 should be reserved only for articles that directly address the specific legal issue in the query. If the judgment is difficult, assign a score below 0.5."*

Articles with a predicted score above 0.5 are retained as relevant; if no article clears this threshold, the highest-scoring article is included as a fallback to guarantee at least one result per query.

For the entailment sub-task, the articles retrieved by the three-stage pipeline are passed to the fine-tuned *Qwen2.5-7B-Instruct* [4] model, which makes a binary entailment decision, determining whether the legal statement in the query is entailed (*Y*) or not (*N*) given the retrieved articles as supporting evidence.

JPSW. This system follows a three-stage pipeline specifically optimized for the Japanese Civil Code corpus. The system transitions from broad multilingual dense retrieval to Ensemble re-ranking to generative re-ranking and logical reasoning.

Stage 1: Multilingual Dense Retrieval. The initial retrieval phase utilizes the *multilingual-e5-large-instruct* [8] model to perform dense semantic search. The 800-article Civil Code corpus is pre-computed into a latent vector space and stored as high-dimensional embeddings. During inference, each query is prepended with a task-specific instruction:

> *"Instruct: Given a legal question about Japanese civil law, retrieve the most relevant civil code article. Query: [User Query]"*.

Candidates are ranked using cosine similarity, and the top-100 most relevant articles are surfaced for the subsequent stages.

Stage 2: Ensemble Re-ranking. The candidate pool is refined through an ensemble of Large Language Models, consisting of *Llama-3.1-8B-Instruct* [3], *Llama-3.1-Swallow-8B-Instruct* [6], and *Gemma-2-Llama-Swallow-9b* [6]. All three models were fine-tuned using Low-Rank Adaptation (LoRA) to handle Japanese legal syntax. This ensemble utilizes a dual-task prompt framing where the model must provide both a relevant article number and a binary (Yes/No) entailment verdict for a given article-statement pair. The scores from these three models are aggregated to prune the candidate set from 100 to the top-50 most probable articles.

Stage 3: Generative Re-ranking & Entailment. The top-50 candidates are further re-scored by a larger *GPT-OSS-Swallow-20B* [6] model. This stage performs a more granular evaluation of the legal nuances within the top-50 pool, reducing the result set to a final selection of K candidates (where $K = 10$ by default). For Entailment, Chain-of-Thought processing is employed to improve logical consistency, and any reasoning is stripped to extract the binary classification. The final retrieval scores undergo global normalization across all querie,s and a dynamic thresholding logic is applied. If the top-ranked candidate's normalized score exceeds 0.6, all candidates with a score greater than or equal to 0.6 are retained. If the primary candidate does not meet the 0.6 threshold, the system defaults to a conservative fallback strategy, returning only the rank-1 result.

4.3 Evaluation Results

Table 5 shows our systems' results for Task 3. As shown in this table, JPGEN and JPLLM both have significantly higher recall compared to JPSW, while JPSW provides a significantly higher precision (Student's paired t-test $p < 0.05$). On one hand, JPSW for more than half of the queries provides a precision higher than 0.5. On the other hand, JPLLM has a recall above (0.5) for more than 80% of queries.

The queries for which JPSW achieves perfect precision while JPLLM retrieves no relevant articles (e.g., R07-12-O, R07-14-A, R07-23-E, and R07-28-U), each is grounded in exactly one relevant statute, covering legally self-contained topics such as creditor expense recovery, mortgage extinction, novation with mortgage, and a partnership member's disposal of their share. For each of these queries, JPSW returned a single article assigned a high normalised relevance score (0.829˜1.0), correctly matching the ground truth. JPLLM, by contrast, assigned the maximum score of 1.0 to every article in its five-item output, none of which were relevant. This score saturation pattern in JPLLM's pointwise re-ranking stage, where the Qwen model's confidence no longer discriminates meaningfully among candidates, is caused by its fallback mechanism, which guarantees at least one returned result per query. For these queries, JPLLM suffers from domain-level semantic confusion; for *R07-23-E* (article 518, novation with mortgage), JPLLM returns articles from the assignment and novation chapter immediately surrounding the correct provision, and for *R07-28-U* (article 676, partnership share disposition), it returns articles 256–260 from the co-ownership chapter entirely. The overlap between "share in partnership property" and "share in co-owned property" mislead the pair-wise re-ranker, while JPSW's generative stage correctly resolves the distinction through its conservative thresholding strategy.

In cases where JPLLM achieves full recall while JPSW retrieves nothing relevant, JPSW either returns no results or commits to a single low-confidence article that is numerically or thematically adjacent to the correct provision. For *R07-03-A* (article 98-2, expressions of intent to a party under curatorship), JPSW returns article 97, the general notice-of-intent provision from which 98-2 departs as a specific exception; for R07-04-I (article 114, ratification silence in unauthorised agency), it retrieves article 113, the adjacent provision on demanding confirmation; and for *R07-21-I*, it returns sub-provision 466-6 when the query implicates the parent article 466. In all these cases, the system correctly identifies the legal chapter but cannot resolve the boundary between a provision and its procedural exception or sub-provision. JPLLM navigates these distinctions more reliably, placing the correct article within the top two ranks in the majority of these queries, confirming that the relevant provision is reachable by broad retrieval but is discarded by JPSW's aggressive pruning. Across both systems, the primary retrieval challenge is therefore not broad domain identification but the fine-grained disambiguation of closely related provisions addressing subtly different legal scenarios.

5 Task 4: Legal Textual Entailment

In this section, we explore task 4, starting with the task definition, followed by our proposed systems and the evaluation results.

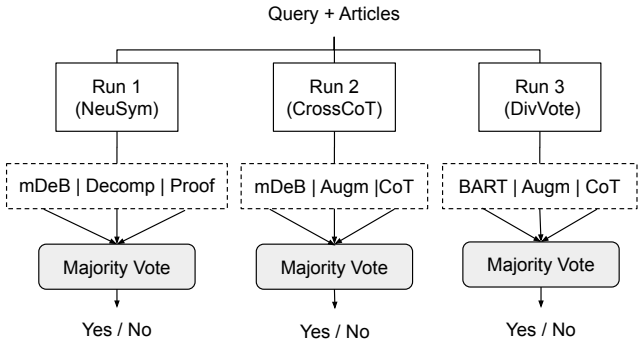


Figure 2: Overview of the three ensemble systems for Task 4. Each run consists of three complementary components whose predictions are aggregated via majority voting.

5.1 Task Definition

This task focuses on legal textual entailment for Yes/No question answering, assuming that the relevant statutory evidence is already provided. Given a natural language query and a set of golden (manually selected) relevant statute articles, the objective is to determine whether the query is entailed (Yes) or not entailed (No) by the provided articles. In contrast to Task 3, where systems must jointly retrieve and reason over statutes, this task isolates the entailment component, enabling a more direct evaluation of a system's reasoning capabilities when the supporting evidence is known.

The corpus used in this task is the same as in Task 3, consisting of Japanese civil law articles paired with query statements in a structured format. Each instance links a query with one or more relevant articles and an associated Yes/No label indicating whether the query is entailed by those articles. The training data provides both the relevant articles and the corresponding entailment labels. At test time, systems are given queries together with the golden relevant articles, and must predict the correct Yes/No decision. As in the other tasks, the entire process must be fully automatic, with no human intervention or modifications based on inspection of the test queries.

Evaluation in this task focuses solely on the correctness of the entailment decision. The primary metric is accuracy, defined as the proportion of queries for which the system correctly predicts the Yes/No label. Unlike Task 3, retrieval quality is not evaluated, as the relevant articles are provided. This setup allows for a more controlled assessment of legal reasoning and entailment, independent of retrieval performance.

5.2 Proposed Models

We submitted three systems to COLIEE 2026 Task 4, each implemented as a three-member majority-vote ensemble as shown in Figure 2. All components are trained using a stratified 90/10 train–validation split of the provided Japanese data, with checkpoint selection based on validation macro-F1. Across all runs, we maintain a shared mDeBERTa cross-encoder baseline and introduce complementary components to capture structured reasoning, data augmentation effects, chain-of-thought (CoT) reasoning, and architectural diversity.

Table 5: AIIRLab Task 3 submission results for Accuracy, Correctly Retrieved, Sufficiently Retrieved, Accuracy of Sufficiently Retrieved, F2, Precision and Recall.

Run	Accuracy	Correct	Sufficiently Retrieved	Acc. Suff. Retrieved	F2	Precision	Recall
JPGEN	0.5854	48	63	0.7619	0.4481	0.1707	0.7785
JPLLM	0.5732	47	64	0.7344	0.4606	0.1770	0.7947
JPSW	0.5000	41	46	0.8913	0.5474	0.5183	0.5915

Shared Baseline: mDeBERTa Cross-Encoder. All ensembles include a fine-tuned *mDeBERTa-v3-base-mnli-xnli* [7] model as a core voter. The model takes the concatenated (article, query) pair as a standard NLI input and is trained with AdamW (learning rate 2×10^{-5}, weight decay 0.01), 10% linear warmup, and gradient clipping at 1.0. Mixed-precision training and gradient accumulation (4 steps) yield an effective batch size of 32. At inference, the entailment and contradiction logits are compared, predicting *Yes* when the former is larger. Training proceeds for up to 10 epochs, and the best checkpoint is selected by validation macro-F1.

Run 1: Neuro-Symbolic Ensemble (NeuSym). This ensemble augments the neural baseline with two structured reasoning components:

First, a *condition–conclusion decomposition model* converts each article and query into a structured *LegalStructure* representation (conditions, effect, exceptions, negation) using *Qwen2.5-7B-Instruct* [4] with LoRA 4-bit quantization. A hybrid model then combines (i) similarity scores computed via a fine-tuned *multilingual-e5-small* encoder [8] over decomposed fields, and (ii) the mDeBERTa entailment logit. These signals are fused through a learned interpolation weight, with separate learning rates for encoder and structural components.

Second, a *neuro-symbolic proof model* translates inputs into logical predicates and rules, performs backward-chaining inference using pyDatalog [4], and extracts proof-derived features (e.g., proof success, depth, unresolved goals). These features are combined with the mDeBERTa score via a lightweight fusion module. This component provides interpretable reasoning traces while injecting symbolic structure into the decision process. The final prediction is obtained by majority voting across the baseline, decomposition, and proof-based components.

Run 2: Cross-Lingual Chain-of-Thought Ensemble (CrossCoT). This ensemble focuses on improving generalization through data augmentation and explicit reasoning supervision.

The first additional component is a *data-augmented mDeBERTa model*. Synthetic training pairs are generated using *Qwen2.5-7B-Instruct* [4] under four strategies: paraphrasing and negation for *Yes* instances, and condition substitution and article-driven claim generation for *No* instances. Augmented samples are filtered to prevent validation leakage, and the model is trained identically to the baseline.

The second component is a *QLoRA fine-tuned CoT model*. Chain-of-thought rationales are generated in Japanese following a structured legal reasoning template, and only samples with correct final answers are retained. The base Qwen model is then fine-tuned using QLoRA via Unsloth for three epochs. At inference, the model generates a reasoning trace and extracts the final Yes/No decision using pattern matching. Final predictions are obtained via majority voting across the baseline, augmented cross-encoder, and CoT model.

[4]https://github.com/pcarbonn/pyDatalog

Table 6: AIIRLab Task 4 Results on COLIEE 2026 test set (82 Queries).

Run	Correct	Accuracy
CrossCoT	64	0.7805
DivVote	64	0.7805
NeuSym	57	0.6951

Run 3: Architecturally Diverse Ensemble (DivVote). This ensemble introduces architectural diversity while reusing strong components from Run 2. The first component replaces the baseline with a fine-tuned *facebook/bart-large-MNLI* [9] cross-encoder. Despite being pretrained on English, BART provides a complementary inductive bias as a sequence-to-sequence model. Training follows the same NLI formulation, with adjusted batch size and learning rate to accommodate model scale. The remaining two components, the data-augmented mDeBERTa model and the QLoRA CoT model, are identical to those in Run 2. As in the other runs, predictions are combined through majority voting.

5.3 Evaluation Results

Table 6 shows the evaluation results of our three runs on COLIEE 2026 test set. All three models had correct predictions on 50 instances, and on 9 queries all models provided wrong predictions. Both CrossCot and DivVote provided the same number of correct instances with an accuracy of 0.78. However, these models had different predictions for 14 instances, for half of which each system had a correct prediction. The NeuSym system had lower accuracy, but for two queries it provided the correct prediction while the other two models failed.

The Neuro-Symbolic Ensemble (NeuSym) showed an advantage in handling complex condition-exception structures and taxonomic mappings. For instance, NeuSym correctly predicted entailment for instances involving nested exceptions, such as the guarantor exception clauses outlined in Article 504 (R07-22-O). Its *LegalStructure* decomposition isolates exceptions into distinct structural fields, preventing them from being overshadowed by the broader semantic overlap of general rules. Furthermore, NeuSym's symbolic component succeeded in taxonomic mapping, such as properly categorizing a "real estate pledge" under the broader "real rights regarding real estate" in Article 177 (R07-08-U), a task where standard cross-encoders failed to establish sufficient dense semantic similarity.

The divergence between the Architecturally Diverse Ensemble (DivVote) and the Cross-Lingual Chain-of-Thought Ensemble (CrossCoT) shows the trade-offs between sequence-to-sequence and encoder-only architectures. DivVote correctly identified structural inversions and hierarchical priority orderings where CrossCoT failed, such as reversing the fault attribution between a creditor and debtor under Article 413-2 (R07-17-U) or falsely inverting the statutory lien priorities of funeral and daily necessity expenses under Article 306/329 (R07-12-A). By incorporating a *BART-large-MNLI* [9] cross-encoder, DivVote exhibited a stricter inductive bias for structural alignment, effectively avoiding the "bag-of-words" similarity traps that often mislead encoder-only models on highly overlapping adversarial texts.

However, DivVote's reliance on an English-pretrained model introduced a semantic bottleneck when processing Japanese legal vocabulary, an area where CrossCoT's native multilingual model excelled. CrossCoT correctly rejected false premises relying on lexical distinctions, such as conflating the mere "loss" of an item with the strict legal threshold of "deprivation" required for possessory actions under Article 200 (R07-10-A). Additionally, CrossCoT successfully anchored its attention to precise parameter checks, including specific quantitative alterations to the statute of limitations for bodily harm (R07-30-O) and identifying the correct party authorized to transfer a mortgage during a novation (R07-23-E). This underscores the necessity of a native multilingual baseline, combined with explicit CoT reasoning, to resolve fine-grained qualitative and quantitative mismatches in original legal texts.

6 Conclusion

This paper presents the AIIR Lab's submissions to COLIEE 2026 across four legal information processing tasks, showing that multi-stage ensemble architectures are well-suited for the complex demands of legal retrieval and reasoning. In the case-based tasks, Fusion3's union ensemble of fine-tuned LLaMA-3.1-8B, Qwen2.5-7B, and DeBERTa achieved an F1 of 0.47 in Task 2, ranking second among all participating teams, while the LLaMA pipeline's document-wide alignment outperformed paragraph-level approaches in Task 1 recall. In the statute-based tasks, JPLLM's pointwise reranking achieved recall up to 0.79 in Task 3 but was vulnerable to score saturation; in contrast, JPSW achieved significantly higher precision ($p < 0.05$), correctly resolving distinctions between complex legal topics such as partnership shares and co-owned property. In Task 4, results showed that architectural diversity and structured reasoning provide complementary gains, with CrossCoT and DivVote both reaching 0.78 accuracy and NeuSym uniquely resolving nested exception structures that purely neural models could not handle.

Our results suggest that the primary bottleneck in legal AI is no longer broad domain identification but fine-grained disambiguation; distinguishing a general rule from its procedural exceptions, a provision from its sub-provisions, and topical relevance from logical entailment. Future work should focus on retrieval architectures that preserve paragraph-level granularity at scale and reasoning models capable of reliably resolving hierarchical legal structures.

References

[1] J. Chen. 2025. M3-Embedding: Multi-Linguality, Multi-Functionality, Multi-Granularity Text Embeddings Through Self-Knowledge Distillation. arXiv:2402.03216 https://arxiv.org/abs/2402.03216

[2] A. Bercovich et al. 2025. Llama-Nemotron: Efficient Reasoning Models. arXiv:2505.00949 [cs.CL] https://arxiv.org/abs/2505.00949

[3] A. Grattafiori et al. 2024. The Llama 3 Herd of Models. arXiv:2407.21783 [cs.AI] https://arxiv.org/abs/2407.21783

[4] A. Yang et al. 2025. Qwen2.5 Technical Report. arXiv:2412.15115 [cs.CL] https://arxiv.org/abs/2412.15115

[5] A. Yang et al. 2025. Qwen3 Technical Report. arXiv:2505.09388 [cs.CL] https://arxiv.org/abs/2505.09388

[6] K. Fujii et al. 2024. Continual Pre-Training for Cross-Lingual LLM Adaptation: Enhancing Japanese Language Capabilities. arXiv:2404.17790 [cs.CL] https://arxiv.org/abs/2404.17790

[7] L. Moritz et al. 2023. Less Annotating, More Classifying: Addressing the Data Scarcity Issue of Supervised Machine Learning with Deep Transfer Learning and BERT-NLI. *Political Analysis* 32 (06 2023), 1–33. doi:10.1017/pan.2023.20

[8] L. Wang et al. 2024. Multilingual E5 Text Embeddings: A Technical Report. arXiv:2402.05672 [cs.CL] https://arxiv.org/abs/2402.05672

[9] M. Lewis et al. 2020. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault (Eds.). Association for Computational Linguistics, Online, 7871–7880. doi:10.18653/v1/2020.acl-main.703

[10] P. He et al. 2021. DeBERTa: Decoding-enhanced BERT with Disentangled Attention. arXiv:2006.03654 [cs.CL] https://arxiv.org/abs/2006.03654

[11] Randy Goebel, Yoshinobu Kano, Mi-Young Kim, Calum Kwan, Ken Satoh, Hiroaki Yamada, and Masaharu Yoshioka. 2026. An Overview of the COLIEE 2025 Competition: Legal Case Law and Statute Law Information Retrieval and Entailment. In *Proceedings of the Twentieth International Conference on Artificial Intelligence and Law (ICAIL '25)*. Association for Computing Machinery, New York, NY, USA, 506–515. doi:10.1145/3769126.3785016

[12] Yiran Hu, Huanghai Liu, Chong Wang, Kunran Li, Tien-Hsuan Wu, Haitao Li, Xinran Xu, Siqing Huo, Weihang Su, Ning Zheng, Siyuan Zheng, Qingyao Ai, Yun Liu, Renjun Bian, Yiqun Liu, Charles L. A. Clarke, Weixing Shen, and Ben Kao. 2026. Evaluation of Large Language Models in Legal Applications: Challenges, Methods, and Future Directions. arXiv:2601.15267 [cs.CY] https://arxiv.org/abs/2601.15267

[13] Yoshinobu Kano, Randy Goebel, Mi-Young Kim, Calum Kwan, Ken Satoh, Hiroaki Yamada, and Masaharu Yoshioka. 2026. An Overview of the COLIEE 2026 Competition: Legal Case Law and Statute Law Information Retrieval and Entailment. In *Proceedings of the 21st International Conference on Artificial Intelligence and Law (ICAIL 2026)*.

[14] Markus Reuter, Tobias Lingenberg, Ruta Liepina, Francesca Lagioia, Marco Lippi, Giovanni Sartor, Andrea Passerini, and Burcu Sayin. 2025. Towards Reliable Retrieval in RAG Systems for Large Legal Datasets. In *Proceedings of the Natural Legal Language Processing Workshop 2025*, Nikolaos Aletras, Ilias Chalkidis, Leslie Barrett, Cătălina Goanță, Daniel Preoțiuc-Pietro, and Gerasimos Spanakis (Eds.). Association for Computational Linguistics, Suzhou, China, 17–30. doi:10.18653/v1/2025.nllp-1.3

[15] Stephen Robertson and Hugo Zaragoza. 2009. *The probabilistic relevance framework: BM25 and beyond*. Vol. 4. Now Publishers Inc.

[16] Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2020. MPNet: Masked and Permuted Pre-training for Language Understanding. In *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (Eds.), Vol. 33. Curran Associates, Inc., 16857–16867. https://proceedings.neurips.cc/paper_files/paper/2020/file/c3a690be93aa602ee2dc0ccab5b7b67e-Paper.pdf

[17] Deiby Wu, Sarah Lawrence, and Behrooz Mansouri. 2025. AIIR Lab at COLIEE 2025: Exploring Applications of Large Language Models for Legal Text Retrieval and Entailmen. In *Workshop of the Tenth Competition on Legal Information Extraction/Entailment (COLIEE'2025) in the 21th International Conference on Artificial Intelligence and Law (ICAIL)*.

ClaUSurf@COLIEE 2026 Task 2: A Two-Stage Retrieve-then-Reason Approach for Legal Case Entailment

Vu Le Hoang
University of Science, VNU-HCM
Ho Chi Minh City, Vietnam
22120461@student.hcmus.edu.vn

Tri Vu Chau Minh
University of Science, VNU-HCM
Ho Chi Minh City, Vietnam
22120456@student.hcmus.edu.vn

Giang Tran Truong
University of Science, VNU-HCM
Ho Chi Minh City, Vietnam
22120085@student.hcmus.edu.vn

Hy Nguyen Tuong Bach
University of Science, VNU-HCM
Ho Chi Minh City, Vietnam
22120455@student.hcmus.edu.vn

Lu Le Phuc
Faculty of Information Technology
University of Science, VNU-HCM
Ho Chi Minh City, Vietnam
lplu@hcmus.edu.vn

Abstract

We present the **ClaUSurf** system submitted to COLIEE 2026 Task 2 (Legal Case Entailment), which requires identifying paragraphs from a noticed case whose legal reasoning logically entails a given decision fragment from a new case. Our system follows a two-stage retrieve-then-reason pipeline. In **Stage 1** (shared across all runs), we fine-tune **Qwen-Embedding-3** (4B) via a two-phase curriculum: an InfoNCE contrastive warm-up followed by hard-negative refinement, achieving **Recall@5 = 93.36%** on the development set. In **Stage 2**, a large language model (LLM) classifies which of the top-5 retrieved candidates entail the query. We submit three runs exploring distinct Stage 2 strategies: (**Run 1**) zero-shot structured chain-of-thought prompting with Qwen-3.5-27B; (**Run 2**) reinforcement learning via Dr.GRPO on Qwen-3.5-9B using exact-match F1 rewards; and (**Run 3**) off-policy knowledge distillation from a Qwen-3.5-35B-A3B MoE teacher to a Qwen-3.5-9B student. Our best system, Run 3, achieves **F1 = 0.3877** on the final test set, outperforming the 27B zero-shot baseline (0.3756) and the RL-tuned model (0.3738), demonstrating that high-quality teacher-generated reasoning traces provide a stronger training signal than either scale or online reward optimization under constrained compute budgets.

CCS Concepts

• **Information systems** → **Question answering**; • **Computing methodologies** → *Information extraction*; • **Applied computing** → *Law*.

Keywords

legal case entailment, dense retrieval, hard-negative mining, large language models, reinforcement learning, knowledge distillation, COLIEE

COLIEE 2026, June 12, 2026, Singapore

ACM ISBN 978-1-4503-XXXX-X/26/XX
https://doi.org/XXXXXXX.XXXXXXX

ACM Reference Format:
Vu Le Hoang, Tri Vu Chau Minh, Giang Tran Truong, Hy Nguyen Tuong Bach, and Lu Le Phuc. 2026. ClaUSurf@COLIEE 2026 Task 2: A Two-Stage Retrieve-then-Reason Approach for Legal Case Entailment. In *Proceedings of Workshop of the Competition on Legal Information Extraction/Entailment (COLIEE 2026)*. ACM, New York, NY, USA, 9 pages. https://doi.org/XXXXXXX.XXXXXXX

1 Introduction

Legal case entailment is the task of determining which paragraph(s) of a cited (noticed) case contain the legal reasoning that supports a specific judicial conclusion from a new case. Unlike general textual entailment, this task requires systems to discriminate between paragraphs that supply a genuine legal *rule or analysis* from which the conclusion follows, and those that are merely topically related—describing background facts, procedural history, or the final disposition of the case. COLIEE (Competition on Legal Information Extraction/Entailment) [1, 2] is an annual benchmark that assesses automated systems on a suite of legal NLP tasks based on Federal Court of Canada case law. Task 2 of COLIEE 2026 poses the paragraph-level case entailment problem: given a decision fragment d from a new case and an ordered list of paragraphs $\{p_1, \ldots, p_n\}$ from a noticed case, a system must predict the subset $\hat{P} \subseteq \{p_i\}$ that logically entails d. Performance is measured by micro-averaged F1. Recent top-performing systems have converged on a *retrieve-then-rerank* paradigm, combining BM25 lexical pre-ranking with neural re-rankers before a final LLM-based verification stage [3, 4]. The NOWJ team, winners of COLIEE 2025 Task 2 (F1 = 0.3195), showed that dual-model LLM voting over a BM25+monoT5 pipeline provides strong precision, while CAPTAIN [4] achieved competitive results by pairing monoT5 re-ranking with few-shot prompting. These systems establish LLM reasoning quality as the primary differentiator at the top of the leaderboard. In this paper, we describe the **ClaUSurf** system for COLIEE 2026 Task 2. Our key design choices are: (*i*) replacing the classical BM25+reranker first stage with a single large bi-encoder (Qwen-Embedding-3, 4B) fine-tuned via a two-phase curriculum, yielding Recall@5 = 93.36%; and (*ii*) systematically evaluating three distinct LLM strategies for Stage 2—zero-shot structured prompting, reinforcement learning with outcome rewards, and off-policy chain-of-thought distillation—to determine

which paradigm best exploits a constrained compute budget. Our main contributions are:

- A two-phase retrieval curriculum combining InfoNCE warm-up with hard-negative refinement on a 4B embedding model, achieving strong candidate recall.
- A structured chain-of-thought prompt that enforces explicit legal reasoning and parsimony constraints, serving as a competitive zero-shot baseline.
- An empirical comparison of three Stage 2 training paradigms, showing that off-policy chain-of-thought distillation from a stronger MoE teacher outperforms a 3× larger zero-shot model and an RL-tuned peer under equivalent inference-time cost, achieving 8th place among 14 teams in the COLIEE 2026 Task 2 leaderboard.

2 Related Work

2.1 Legal Retrieval and COLIEE Systems

Early COLIEE Task 2 systems relied on BM25 [13] lexical retrieval followed by shallow classifiers. The adoption of pre-trained transformers [14] introduced dense bi-encoder retrieval and cross-encoder re-ranking as standard pipeline components. Cross-encoders fine-tuned with task-specific hard negatives—notably monoT5 [12]—have served as strong re-ranking baselines in recent cycles, as demonstrated by CAPTAIN [4], which combined monoT5 hard-negative fine-tuning with few-shot FlanT5 prompting. More recently, large language models have been incorporated as a final verification stage. The NOWJ team [3], winners of COLIEE 2025 Task 2 (F1 = 0.3195), showed that dual-model LLM voting (QwQ-32B and DeepSeek-V3) over a BM25+monoT5 pipeline substantially improves precision, establishing LLM reasoning quality as the primary differentiator at the top of the leaderboard.

2.2 Chain-of-Thought Prompting

Wei et al. [11] demonstrated that eliciting step-by-step reasoning chains from LLMs substantially improves performance on multi-step tasks without additional training. Subsequent work on structured prompting [19] showed that sampling multiple reasoning paths and aggregating via majority vote (self-consistency) further improves robustness. In legal NLP, chain-of-thought prompting is particularly appealing because legal entailment requires explicit identification of the applicable rule, its relationship to the facts, and the logical link to the conclusion—a reasoning structure that maps naturally onto sequential chain-of-thought steps.

2.3 Reinforcement Learning for LLM Reasoning

Reinforcement learning from human feedback (RLHF) [15] established RL as a viable paradigm for aligning LLMs with human preferences. More recent work has shifted toward outcome-based RL with verifiable rewards, bypassing the need for learned reward models. DeepSeek-R1 [16] introduced Group Relative Policy Optimization (GRPO) [9], which replaces the per-token critic in PPO with group-level baselines computed from multiple sampled responses to the same prompt, substantially reducing training overhead. Dr.GRPO [8] extends GRPO with decomposed, rule-guided reward signals and improved gradient variance reduction, demonstrating state-of-the-art results on mathematical and logical reasoning benchmarks. We adopt Dr.GRPO for Run 2, applying it with an exact-match F1 reward to directly optimise for the COLIEE evaluation metric.

2.4 Knowledge Distillation for LLM Reasoning

Knowledge distillation [10] transfers the capabilities of a larger teacher model to a smaller student. Classical approaches match output logits, but for LLM reasoning, recent work has focused on *chain-of-thought distillation*: the teacher generates explicit reasoning traces that the student learns to reproduce via supervised fine-tuning. Hsieh et al. [17] showed that distilling both the rationale and the label ("Distilling Step-by-Step") enables smaller models to outperform few-shot prompted models many times their size. Mukherjee et al. [18] further demonstrated that training on rich, explanation-augmented signals from a stronger teacher yields substantial gains over standard instruction tuning, even from relatively few samples. Our Run 3 follows this paradigm: a MoE teacher generates oracle-guided reasoning traces conditioned on gold labels, and a 9B student is fine-tuned on these traces via standard SFT.

2.5 Positioning of Our Work

Our system departs from prior COLIEE entries in two respects: (*i*) replacing the BM25+cross-encoder Stage 1 with a single fine-tuned 4B bi-encoder, and (*ii*) providing a controlled empirical comparison of three Stage 2 paradigms—zero-shot chain-of-thought prompting, on-policy RL with Dr.GRPO, and off-policy chain-of-thought distillation—to identify which regime best utilises a constrained compute budget. To our knowledge, this is the first COLIEE submission to systematically compare RL and distillation as LLM training strategies for legal entailment.

3 Task Description

3.1 Problem Formulation

COLIEE 2026 Task 2 is a paragraph-level legal entailment task. Formally, given:

- a *decision fragment* d: a judicial conclusion or legal determination from a new case, and
- a *noticed case* $C = \langle p_1, p_2, \ldots, p_n \rangle$: an ordered list of numbered paragraphs from a case that was cited in the new case,

the system must output a subset $\hat{P} \subseteq C$ such that each selected paragraph provides the legal rule, analysis, or interpretation from which d logically follows. A paragraph p_i is considered entailing if and only if a reader knowing only p_i could derive the legal conclusion in d. Paragraphs describing background facts, procedural history, bare citations, or the final dispositional order do not qualify, even when topically related to d.

3.2 Dataset

The COLIEE 2026 Task 2 dataset is derived from Federal Court of Canada decisions. The training corpus is constructed from prior COLIEE competition cycles and provides labeled triples of (decision fragment, noticed case paragraphs, gold entailing paragraph labels). Paragraph counts per noticed case vary widely, ranging from a

handful to over 60 paragraphs, with a median of approximately 20 paragraphs per case. The majority of queries have exactly one entailing paragraph; a small fraction (typically under 20%) require two or three. Table 1 summarises the corpus statistics.

Table 1: COLIEE 2026 Task 2 dataset statistics. Dev is a held-out validation subset from the official training corpus. The Stage 2 fine-tuning pool (790 instances) is derived from the combined train+dev set after gold-coverage filtering (see Section 6.1).

Split	Cases	Paragraphs (approx.)
Train	726	14,500
Dev	100	∼2,000
Test (official)	100	—
Gold paras / query	**Proportion**	
1	≈80%	
2	≈15%	
≥3	≈5%	

3.3 Evaluation

Following standard COLIEE protocol [2], performance is measured by *micro-averaged F1*:

$$\mathrm{F1} = \frac{2 \cdot \sum_i |\hat{P}_i \cap P_i^*|}{\sum_i |\hat{P}_i| + \sum_i |P_i^*|} \tag{1}$$

where $\hat{P}_i$ and P_i^* are the predicted and gold paragraph sets for the i-th test instance, respectively. Summing numerator and denominator over all instances before dividing rewards systems that are consistently precise and recall-complete across the full test set.

4 System Overview

Our pipeline consists of two sequential stages applied to each test instance, as illustrated in Figure 1. **Stage 1 — Dense Retrieval.** Given decision fragment d and noticed case $C = \langle p_1, \ldots, p_n \rangle$, we encode both using a shared fine-tuned bi-encoder and retrieve the top-k paragraphs by cosine similarity, reducing the candidate pool from n (up to 60+) to $k = 5$. This stage is *identical* across all three submitted runs. **Stage 2 — LLM Entailment Classification.** The top-k paragraphs, together with their original labels and the query d, are provided to a large language model via a structured prompt. The LLM reasons over all candidates jointly and outputs the subset it judges to be entailing. This stage *differs* across the three runs, varying the model capacity and training strategy. This two-stage design follows established practice [3, 4] but differs in two ways: we replace the typical BM25+monoT5 first stage with a single high-capacity embedding model, and we systematically compare RL and distillation as Stage 2 alternatives to zero-shot prompting.

5 Stage 1: Dense Retrieval

5.1 Model

We use **Qwen-Embedding-3**[1] [5] (4B parameters) as our retrieval encoder. Qwen-Embedding-3 is a decoder-based dense embedding model that produces fixed-length representations via mean pooling over the final hidden states. Its large capacity and extended context window make it well-suited for encoding lengthy legal paragraphs without aggressive truncation. Both query and paragraph embeddings are ℓ_2-normalized; retrieval is performed by cosine similarity (equivalent to inner product after normalization).

5.2 Phase 1: InfoNCE Warm-Up

We fine-tune the encoder for 5 epochs using the InfoNCE contrastive objective [6]:

$$\mathcal{L}_{\text{InfoNCE}} = -\log \frac{\exp(\mathrm{sim}(d, p^+)/\tau)}{\exp(\mathrm{sim}(d, p^+)/\tau) + \sum_j \exp\Big(\mathrm{sim}(d, p_j^-)/\tau\Big)} \tag{2}$$

where p^+ is the gold entailing paragraph, $\{p_j^-\}$ are in-batch negatives drawn from other instances in the same mini-batch, $\mathrm{sim}(\cdot,\cdot)$ denotes cosine similarity, and τ is a learnable temperature parameter. Training with in-batch negatives provides approximately $B - 1$ negatives per sample for batch size B, establishing initial semantic alignment between decision fragments and entailing paragraphs.

5.3 Phase 2: Hard-Negative Refinement

After Phase 1, we identify *hard negatives*: paragraphs that rank highly under the Phase 1 checkpoint but do not entail the query. For each training instance, we run the Phase 1 encoder in inference mode, retrieve the top-20 paragraphs, exclude all gold positives, and retain the top-5 as hard negatives. We then fine-tune for an additional 5 epochs with augmented batches containing both in-batch negatives and instance-specific hard negatives. This phase is critical because paragraphs within a single legal case share extensive lexical overlap (shared parties, procedural history, and recurring legal terms), making naive in-batch negatives insufficiently discriminative. Hard negatives force the encoder to learn fine-grained distinctions between genuinely entailing paragraphs and superficially similar distractors. Prior work on dense retrieval has shown that iterative hard-negative mining substantially improves downstream recall [7].

5.4 Inference

At test time, we encode the query d and all paragraphs $\{p_1, \ldots, p_n\}$ of the noticed case, rank by cosine similarity, and pass the top-$k = 5$ candidates (with their original paragraph labels, e.g., “001”, “005”) to Stage 2. On the development set, this retrieval stage achieves **Recall@5 = 93.36%**, meaning that all gold entailing paragraphs are included in the top-5 candidates for 93.36% of instances. This high recall establishes a tight performance ceiling for the overall pipeline and ensures that Stage 2 errors are primarily attributable to entailment classification rather than retrieval failure.

[1] https://huggingface.co/Qwen/Qwen3-Embedding

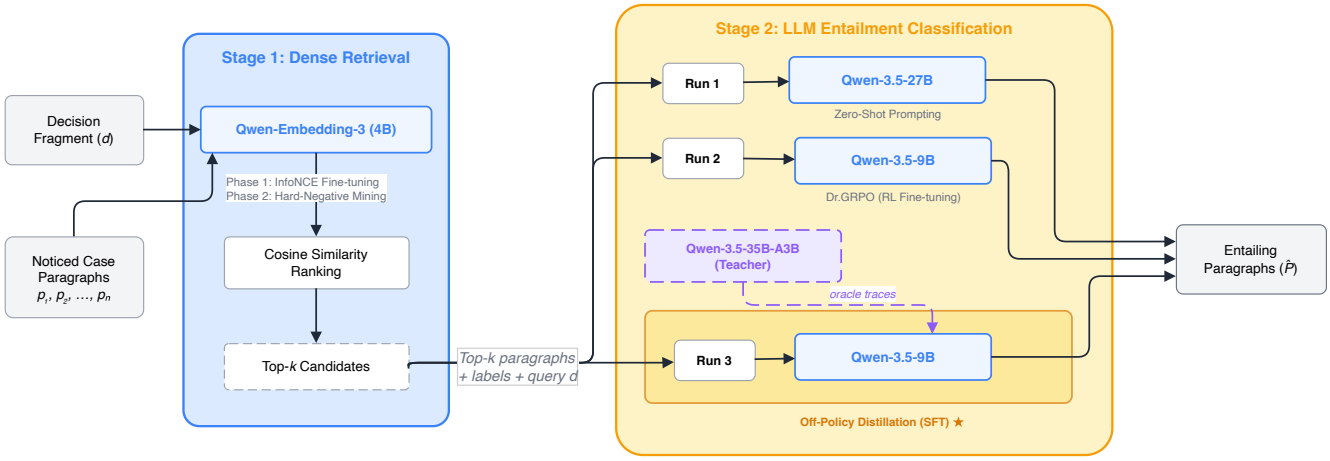


Figure 1: Overview of the ClaUSurf two-stage pipeline for COLIEE 2026 Task 2. Stage 1 (Dense Retrieval): the decision fragment d and all noticed-case paragraphs are independently encoded by fine-tuned Qwen-Embedding-3 (4B); top-5 candidates are selected by cosine similarity. Stage 2 (LLM Entailment Classification): the three submitted runs apply different strategies—zero-shot prompting (Run 1), Dr.GRPO reinforcement learning (Run 2), and off-policy knowledge distillation (Run 3)—to identify which candidates entail d.

6 Stage 2: LLM-Based Entailment Reasoning

All three runs share the same Stage 1 retriever (top-5 candidates). They differ in the model and training strategy used for Stage 2 classification. Run 1 sets a zero-shot ceiling at scale; Run 2 asks whether task-specific RL on a 9B model can close the size gap; and Run 3 tests whether distillation from a stronger teacher outperforms both. Together, they form a controlled study of Stage 2 training strategies under a fixed inference budget.

6.1 Stage 2 Training Data: Gold-Filtered Samples

The two runs involving Stage 2 fine-tuning (Run 2 and Run 3) require training data constructed from Stage 1 retrieval outputs. If Stage 1 fails to place all gold entailing paragraphs in its top-5 for a given query, the instance cannot provide a meaningful training signal—the model would be asked to identify an entailing paragraph absent from its input. We therefore filter the training corpus to retain only instances where *all* gold paragraphs appear in the Stage 1 top-5 candidate set. Concretely, we run the Phase 2 retrieval encoder over the combined pool of all labeled instances available for fine-tuning—the 726 official training instances together with the 100 development instances we hold out for validation (826 total)—and keep those satisfying $P^* \subseteq \text{top-}k(d)$, yielding **790 usable instances**. We then split these into an 80/20 partition: approximately **632 instances for training** and **158 instances for validation**. The training partition is used for both Run 2 (Dr.GRPO) and Run 3 (distillation SFT); the held-out partition serves as an internal development set for hyperparameter selection and early stopping. This filtering ensures that both the RL reward signal (Run 2) and the teacher-generated distillation traces (Run 3) are grounded in realistic retrieval contexts: Stage 2 sees the same candidate distribution at training and inference time, preventing train–test mismatch.

6.2 Run 1: Zero-Shot Prompting with Qwen-3.5-27B

6.2.1 Model. We use **Qwen-3.5-27B**[2] [5], a 27B instruction-tuned language model, in a zero-shot setting. Hardware constraints precluded the use of the model's native extended thinking/reasoning mode (e.g., extended chain-of-thought decoding or budget forcing), which we hypothesize would yield stronger performance.

6.2.2 Prompt Design. Following chain-of-thought prompting [11], the prompt requires the model to reason step-by-step before committing to a final label. The design decisions, informed by the task definition and error analysis on the development set, are: **(1) Goal framing and clarification.** The prompt opens with an explicit goal statement and a clarification that the "decision" in the query is *not* the final ruling of the case but a judicial conclusion that must be supported by reasoning in the noticed case. This distinction addresses a recurring confusion observed during development. **(2) Explicit entailment definition.** The model is given a precise definition: a paragraph entails the query if its meaning provides the legal *rule, analysis, or reasoning* from which the judge's conclusion logically follows. **(3) Explicit negative filtering.** The model is instructed to exclude paragraphs containing only: background facts, procedural history, general introductions, citations without explanation, or the final order or disposition of the case. This directly addresses a common failure mode on the development set. **(4) Parsimony constraint.** Seven selection rules enforce a strong bias toward returning exactly one paragraph, permitting two or three only when they form a necessary reasoning chain (e.g., one paragraph states a rule and another applies it). An additional tie-breaking rule instructs the model to select the single paragraph whose meaning most directly entails the query when several are related. **(5) Guided procedure.** A four-step procedure directs the

[2] https://huggingface.co/Qwen/Qwen3.5-27B

Role: You are a legal expert performing a legal textual entailment task. **Goal:** Given the decision statement of a new case (query) and the paragraphs of a noticed case, identify which paragraph(s) logically entail the decision. **Definition:** A paragraph *entails* the query if its meaning provides the *legal rule, analysis, or reasoning* from which the judge's conclusion logically follows. **Clarification:** The "decision" in the query is *not* the final ruling. It is a judicial conclusion that must be supported by reasoning found in the noticed case. **Selection rules:**

(1) Select paragraph(s) whose legal reasoning most directly entails the query.
(2) Prefer paragraphs that explain legal reasoning, apply rules, or interpret statutes.
(3) Do **NOT** select: background facts, procedural history, general introductions, bare citations, or the final order.
(4) Return **ONE** paragraph in most cases; return two/three only if they form a necessary reasoning chain.
(5) If several paragraphs are related, choose the one whose meaning most directly entails the query.

Procedure: (1) Identify the legal rule implied by the query. (2) Compare the query with each paragraph. (3) Find the paragraph(s) whose meaning entails the query. (4) Return the minimal set of labels. **Query:** *{query}* **Paragraphs:** *{paragraphs}* Reason step-by-step (≤10 sentences).
Output: `<reasoning>...</reasoning>`
`<answer>LABELS</answer>`

Figure 2: Abridged prompt template for Run 1 (Qwen-3.5-27B, zero-shot). Variables `{query}` and `{paragraphs}` are substituted at inference time. The full prompt includes seven selection rules and a worked example.

model to: (i) identify the legal rule implied by the query, (ii) compare it against each candidate paragraph, (iii) find the paragraph(s) whose meaning logically entails the query, and (iv) return the minimal label set. **(6) Structured output with example.** Reasoning is enclosed in `<reasoning>...</reasoning>` tags (capped at 10 sentences) and final labels in `<answer>...</answer>` tags, with a concrete two-paragraph example demonstrating the expected format. Figure 2 shows the abridged prompt template.

6.2.3 Result. Run 1 achieves **F1 = 0.3756** on the COLIEE 2026 Task 2 test set, establishing a competitive zero-shot baseline.

6.3 Run 2: Reinforcement Learning via Dr.GRPO on Qwen-3.5-9B

6.3.1 Method. For Run 2, we apply **Dr.GRPO** (Decomposed Rule-Guided GRPO) [8] to fine-tune **Qwen-3.5-9B**[3] [5] using the same structured prompt as Run 1. Dr.GRPO extends the GRPO algorithm introduced in DeepSeek-R1 [9, 16] with decomposed, rule-guided reward signals and improved gradient variance reduction, making it particularly effective for LLM fine-tuning with sparse outcome-based rewards (Section 2).

6.3.2 Reward Signal. For each training instance (d, C, P^*), the model generates $G = 8$ candidate responses per prompt. The reward combines two components: a *task reward* and a *format reward*. **Task reward.** The task reward r_{task} is the exact-match F1 between the predicted paragraph labels $\hat{P}$ and the gold set P^*:

$$r_{\text{task}}(\hat{P}, P^*) = \frac{2\,|\hat{P} \cap P^*|}{|\hat{P}| + |P^*|} \tag{3}$$

Format reward. Since the structured prompt requires responses to follow a specific schema (`<reasoning>...</reasoning>` followed by `<answer>...</answer>`), we additionally assign a binary format reward $r_{\text{fmt}} \in \{0, 1\}$ that equals 1 if and only if the response contains both well-formed tag pairs and the answer tags enclose a parseable comma-separated list of paragraph labels. This reward prevents the policy from drifting toward outputs that may accidentally achieve non-zero F1 via partial matches but are unparseable at inference time. The final reward is a weighted combination of the two components:

$$r = r_{\text{task}} + \lambda\, r_{\text{fmt}}, \quad \lambda = 0.1 \tag{4}$$

where λ is set small enough to avoid dominating the task signal while still providing a consistent gradient toward well-formed outputs. The policy gradient is scaled by the advantage relative to the group mean reward, reducing the variance inherent in sparse binary-outcome rewards. A KL-divergence penalty from the reference policy prevents excessive deviation from the pretrained behavior.

6.3.3 Training Details. Due to computational budget limitations, training was conducted for approximately **200 optimization steps** with a batch size of 4 instances (32 rollouts total) and a learning rate of 5×10^{-7}. We hypothesise that training on a larger and more diverse dataset, or simply running more optimisation steps, could yield further improvements. Training instances are drawn from the gold-filtered pool described in Section 6.1.

6.3.4 Result. Run 2 achieves **F1 = 0.3738**—slightly below the 27B zero-shot baseline but competitive given the 3× smaller model capacity (9B vs. 27B). Despite the severely limited training budget, the RL-tuned 9B model nearly matches a 27B model, suggesting that task-specific RL adaptation can partially compensate for the parameter gap.

6.4 Run 3: Off-Policy Knowledge Distillation

Run 3 adopts an **off-policy chain-of-thought distillation** strategy [10, 17]: a stronger teacher model generates high-quality, oracle-guided reasoning traces that serve as supervised fine-tuning (SFT) targets for a smaller student.

6.4.1 Teacher: Qwen-3.5-35B-A3B (MoE).. The teacher model is **Qwen-3.5-35B-A3B**[4] [5], a Mixture-of-Experts (MoE) architecture with 35B total parameters and 3B active parameters per forward pass. The MoE design provides strong reasoning capacity at moderate inference cost. Critically, the teacher is provided with the *gold paragraph labels* and tasked with generating oracle-guided reasoning traces. The prompt instructs the teacher to: (i) read the query and all candidate paragraphs, (ii) write a concise reasoning chain

[3] https://huggingface.co/Qwen/Qwen3.5-9B

[4] https://huggingface.co/Qwen/Qwen3.5-35B-A3B

Role: You are a legal expert. Your job is to write clear, step-by-step reasoning that explains *why* the given paragraph(s) entail the legal decision in the query. **Context:** A judge made a legal determination (the query). The correct paragraph(s) from a noticed case that support this determination are: *{gold_labels}*. **Your task:**

(1) Read the query (the judge's legal determination).
(2) Read ALL paragraphs from the noticed case.
(3) Write reasoning (≤10 sentences) explaining: what legal rule or principle the query is about; why *{gold_labels}* contain the reasoning that entails the query; why the other paragraphs do **NOT** entail the query.
(4) End with the answer.

Important: Focus on the *legal reasoning connection* between the query and the correct paragraph(s). Explain how the paragraph's legal rule, analysis, or interpretation supports the query. Be concise and precise. **Query:** *{query}* **Paragraphs:** *{paragraphs}* **Correct answer:** *{gold_labels}* Output: `<reasoning>...</reasoning>`
`<answer>`*{gold_labels}*`</answer>`

Figure 3: Oracle-guided teacher prompt for Run 3 (Qwen-3.5-35B-A3B). Gold labels are provided to ensure factually grounded reasoning traces for student distillation.

(at most 10 sentences) explaining what legal rule the query invokes, why the gold paragraph(s) contain the reasoning that entails it, and why the remaining candidates do not, and (iii) confirm the gold labels as the final answer. An additional emphasis block directs the teacher to focus specifically on the *legal reasoning connection* between query and paragraph—how the paragraph's rule, analysis, or interpretation supports the judicial determination—rather than surface-level topical overlap. This oracle conditioning ensures that the generated traces are factually grounded and provide rich pedagogical signal, even for cases where the gold entailment is subtle or requires domain knowledge. The teacher prompt template is shown in Figure 3.

6.4.2 Student: Qwen-3.5-9B (SFT).. The teacher-generated traces (query, paragraphs, reasoning, gold answer) are used as SFT targets for **Qwen-3.5-9B**. The student is fine-tuned to reproduce both the reasoning chain and the final label, conditioned on the dedicated system prompt shown in Figure 4. This prompt shares the same goal framing, entailment definition, and clarification as Run 1, but is more concise: it omits the four-step procedure, the tie-breaking rule, and the worked example, retaining five core selection rules. Training uses standard next-token prediction loss with a learning rate of 1×10^{-5} over 3 epochs. Training instances are drawn from the gold-filtered pool described in Section 6.1. Due to resource constraints, we generated oracle traces for and trained on the **632 training-split instances** from the gold-filtered pool (Section 6.1).

6.4.3 Result. Run 3 achieves **F1 = 0.3877**—the best performance among all three submitted runs. This result establishes that high-quality, oracle-guided reasoning traces from a stronger teacher, even in modest quantity (632 samples), provide a more effective training signal than zero-shot prompting a 3× larger model or online RL under a limited step budget.

Role: You are a legal expert performing a legal textual entailment task. **Goal:** Given the decision statement of a new case (query) and the paragraphs of a noticed case, identify which paragraph(s) logically entail the decision. **Definition:** A paragraph *entails* the query if its meaning provides the *legal rule, analysis, or reasoning* from which the judge's conclusion logically follows. **Clarification:** The "decision" in the query is *not* the final ruling. It is a judicial conclusion that must be supported by reasoning found in the noticed case. **Selection rules:**

(1) Select paragraph(s) whose legal reasoning most directly supports or entails the query.
(2) Prefer paragraphs that explain legal reasoning, apply legal rules, or interpret statutes.
(3) Do **NOT** select paragraphs containing only background facts, procedural history, general introductions, bare citations, or the final order.
(4) Return **ONE** paragraph in most cases.
(5) Return two/three **ONLY** if they form a necessary reasoning chain.

Reason step-by-step (≤10 sentences).
Output: `<reasoning>...</reasoning>`
`<answer>LABELS</answer>`

Figure 4: Student system prompt for Run 3 (Qwen-3.5-9B, SFT). This prompt conditions both teacher-trace generation and student inference.

7 Experimental Results

7.1 Implementation Details

Stage 1 fine-tuning (Qwen-Embedding-3, 4B) was conducted on a single NVIDIA V100 (32 GB), requiring approximately 6 minutes for Phase 1 and 10 minutes for Phase 2 (including hard-negative mining and refinement). Run 2 Dr.GRPO fine-tuning (Qwen-3.5-9B) was performed on a single NVIDIA RTX 5090, completing 200 optimisation steps with $G = 8$ rollouts per prompt in approximately 4 hours. Run 3 teacher inference (Qwen-3.5-35B-A3B MoE) for trace generation used a single NVIDIA RTX 6000 (96 GB), requiring approximately 20 minutes for 632 oracle traces; student SFT (Qwen-3.5-9B, 3 epochs, 632 samples) was conducted on the same GPU and required approximately 1 hour. All experiments were conducted at the University of Science, VNU-HCM.

7.2 Main Results

Table 2 summarizes the performance of all three submitted runs on the development and final test sets, including precision and recall as reported by the COLIEE 2026 organizers. The precision–recall breakdown reveals distinct behavioural profiles across the three runs. Run 1 (zero-shot 27B) is the most conservative: it achieves the highest precision (0.7400) but the lowest recall (0.2517), indicating that the parsimony constraint in the prompt causes the model to select very few paragraphs, often missing relevant ones. Run 2 (RL 9B) relaxes this tendency slightly, improving recall to 0.2619 at the cost of lower precision (0.6525). Run 3 (distillation 9B)

Table 2: Results on the COLIEE 2026 Task 2 development and test sets. All runs share Stage 1 (Recall@5 = 93.36% on dev). The *Control* row shows Qwen-3.5-9B in a zero-shot setting (dev only) to isolate the effect of training strategy from model size. Test Precision, Recall, and F1 are reported by the competition organizers. Best scores in bold; second-best underlined.

Run	Stage 2 Model	Method	Tuning Budget	Dev F1	Test P	Test R	Test F1
Control	Qwen-3.5-9B	Zero-shot prompt	—	0.7420	*(dev-only baseline)*		
Run 1	Qwen-3.5-27B	Zero-shot prompt	—	0.7979	**0.7400**	0.2517	0.3756
Run 2	Qwen-3.5-9B	Dr.GRPO (RL)	≈200 steps	0.7523	0.6525	0.2619	0.3738
Run 3	Qwen-3.5-9B	Distillation (SFT)	632 samples	**0.8014**	0.6357	**0.2789**	**0.3877**

achieves the best F1 (0.3877) by further boosting recall to 0.2789 while maintaining competitive precision (0.6357), suggesting that the teacher-generated reasoning traces help the student identify entailing paragraphs that the zero-shot and RL models tend to overlook. **Controlled baseline.** A key confound in the comparison between Run 1 (27B) and Runs 2–3 (9B) is model size. To disentangle this from training strategy, we evaluate **Qwen-3.5-9B in a zero-shot setting** (the *Control* row in Table 2), using the same structured prompt as Run 1. On the development set, this 9B zero-shot control achieves Dev F1 = 0.7420—substantially below both Run 2 (0.7523, RL-tuned) and Run 3 (0.8014, distilled), confirming that the gains from RL and distillation are attributable to training strategy rather than model size alone. The control was not submitted to the official evaluation, so test-set metrics are unavailable. The overall low recall across all runs (0.25–0.28) indicates that the primary bottleneck is *under-prediction*: all three systems miss a substantial fraction of gold entailing paragraphs. This is partly attributable to the Stage 1 retrieval ceiling (Recall@5 = 93.36% on dev) and partly to the strong one-paragraph bias encoded in the prompts, which may cause Stage 2 to under-select in multi-paragraph entailment instances.

7.3 Stage 1 Retrieval Analysis

The development-set Recall@5 of 93.36% establishes a tight upper bound for the full pipeline: the 6.64% miss rate represents instances where no gold paragraph appears in the top-5, making Stage 2 recovery impossible. Closing this gap requires either a stronger Stage 1 encoder, a larger k (at the cost of longer Stage 2 context), or a Stage 1.5 cross-encoder re-ranker. **Comparison with BM25 and phase ablation.** A primary design claim of our system is that a single fine-tuned bi-encoder can replace the BM25+cross-encoder Stage 1 used by prior systems. To substantiate this claim, Table 3 compares the Recall@5 of four retrieval configurations on the development set: (i) BM25 lexical retrieval, serving as the classical baseline; (ii) Qwen-Embedding-3 without any fine-tuning (zero-shot); (iii) Qwen-Embedding-3 fine-tuned with InfoNCE warm-up only (Phase 1); and (iv) the full two-phase curriculum with additional hard-negative refinement (Phase 1+2). The results confirm that each training phase provides a meaningful contribution. The off-the-shelf Qwen-Embedding-3 (85.30%) performs comparably to BM25 (86.16%), indicating that domain-specific fine-tuning is essential for the legal retrieval setting. Notably, the zero-shot neural encoder marginally underperforms BM25, confirming that general-purpose dense representations do not automatically surpass lexical retrieval on specialised legal text without task-specific adaptation. Phase 1 InfoNCE training improves recall by +4.11 percentage points over the zero-shot baseline, establishing initial semantic alignment. Phase 2 hard-negative refinement adds a further +3.95 points, confirming that within-case hard negatives are critical for discriminating between topically similar paragraphs (Section 5). The full two-phase pipeline (93.36%) outperforms BM25 by +7.20 points, validating our design choice to replace the classical BM25+reranker Stage 1 with a single fine-tuned bi-encoder.

Table 3: Stage 1 retrieval ablation on the development set (k=5). The two-phase fine-tuned bi-encoder is our submitted configuration.

Stage 1 Retrieval Method	Recall@5 (Dev)
BM25 [13]	86.16%
Qwen-Embedding-3, Zero-shot	85.30%
Qwen-Embedding-3, Phase 1 only (InfoNCE)	89.41%
Qwen-Embedding-3, Phase 1+2 *(ours)*	**93.36%**

7.4 Comparison with COLIEE 2026 Participants

Table 4 reports the official COLIEE 2026 Task 2 results for all participating teams, showing the best run per team alongside all three ClaUSurf submissions. Across 14 teams and 35 total submissions, our best run (Run 3, F1 = 0.3877) places 8th by team, ranking below the top cluster of teams (IAI, AIIRLab, JNLP) that achieve F1 in the 0.44–0.49 range. The top-performing teams generally achieve substantially higher recall (0.43–0.54) than our system (0.28), suggesting that their pipelines are more aggressive in selecting multiple entailing paragraphs. By contrast, our Run 1 and Run 3 achieve competitive or superior precision (0.74 and 0.64, respectively) relative to most teams, indicating that our Stage 2 reasoning quality is strong but that recall remains the binding bottleneck—consistent with the under-prediction analysis in Section 7. Notably, the NOWJ team—winners of COLIEE 2025 Task 2 (F1 = 0.3195)—achieves their best 2026 result with a high-precision, low-recall profile (P = 0.7037, R = 0.3231, F1 = 0.4429), a pattern similar to our Run 1. Across all 35 submissions, Run 1 records the third-highest precision (0.7400), behind only NOWJ's `nowj001` (0.7604) and DU's `task2_du2` (0.7529)—all three sharing the same high-precision, low-recall trade-off, suggesting this is a systemic failure mode of parsimony-biased systems rather than a ClaUSurf-specific limitation.

Table 4: Official COLIEE 2026 Task 2 leaderboard (best run per team, all ClaUSurf runs shown). Underlined metrics: ClaUSurf submissions, bold: best value per column. Results sourced from the official competition website, teams with F1 = 0.0000 omitted.

Rank	Team	Run ID	Precision	Recall	F1
1	IAI	task2_run2	0.4501	**0.5374**	**0.4899**
2	AIIRLab	task2_aiirfusion3	0.5120	0.4354	0.4706
3	JNLP	submission (3)	0.4174	0.4898	0.4507
4	NOWJ	nowj003	0.7037	0.3231	0.4429
5	UA	result_run1_ua	0.4711	0.3878	0.4254
6	JUNLLP	submission (1)	0.6721	0.2789	0.3942
7	bosch	bosch_task_2 (1)	0.3900	0.3980	0.3939
8	**ClaUSurf**	task2_clsop3d *(Run 3)*	0.6357	0.2789	0.3877
–	**ClaUSurf**	task2_cls4b27bp11 *(Run 1)*	**0.7400**	0.2517	0.3756
–	**ClaUSurf**	task2_clsp2d *(Run 2)*	0.6525	0.2619	0.3738
9	DU	task2_du3	0.6907	0.2279	0.3427
10	74688	ualbanyllm	0.6500	0.2211	0.3299
11	KeioAndrewShin	task2_submission_ii2	0.4795	0.2381	0.3182
12	CityUMO	task2_submission	0.6000	0.2041	0.3046
13	Sach	sach_task2_run1	0.3929	0.1497	0.2167

Total: 14 teams, 35 submissions

8 Discussion

8.1 Why Distillation Outperforms Prompting and RL

Run 3 (distillation) outperforms both Run 1 (zero-shot 27B) and Run 2 (RL 9B) despite using a 3× smaller model than the former and lacking the online exploration of the latter. We attribute this to three factors. **Richness of the training signal.** Oracle-guided reasoning traces encode the exact inferential steps that COLIEE Task 2 demands: identifying the applicable legal principle, matching it against each candidate paragraph, and articulating why topical similarity alone does not constitute entailment. This structured supervision is far more informative than the scalar F1 reward in RL, which forces the model to rediscover the same reasoning pattern through trial and error [17]. **Adaptation vs. scale.** Although Qwen-3.5-27B (Run 1) has substantially more parameters, zero-shot prompting cannot adapt its representations to domain-specific distinctions—such as the systematic difference between judicial reasoning and dispositional language—that recur throughout COLIEE. Fine-tuning a 9B model on even a modest set of domain-specific traces closes much of this gap, consistent with findings in Orca [18] that explanation-tuned small models can match or exceed much larger prompted models. **Limited RL training budget.** Run 2 was trained for only ∼200 steps on a relatively small pool of 632 instances. Training on a larger dataset—potentially augmented with additional COLIEE cycles or synthetic examples—and running for more optimisation steps could allow the model to better internalise legal reasoning patterns. A fully converged Dr.GRPO model trained on more data could plausibly close the gap with or exceed Run 3, since RL can in principle discover reward-maximising strategies absent from the teacher's fixed trace distribution [8].

8.2 Error Analysis

Qualitative inspection of development-set errors reveals three recurring failure modes. **Procedural paragraph confusion.** The most frequent Stage 2 error involves selecting paragraphs that recount background facts, procedural history, or the final disposition—despite explicit negative-filtering instructions in the prompt. These paragraphs share high lexical overlap with the decision fragment (common parties, statutory references, procedural terms) but lack the operative legal rule or analytical reasoning. This pattern is the dominant source of false positives across all three runs. **Distributed reasoning chains.** Roughly 15–20% of gold-labelled instances require two or more paragraphs whose *joint* content entails the query, with no single paragraph sufficing alone. Stage 2 tends to select only the most salient member of such chains, yielding partial recall. This failure is most pronounced in Run 1, where the model has no domain-specific calibration for recognising multi-paragraph dependencies. **Irreversible Stage 1 misses.** In the 6.64% of instances where the gold paragraph falls outside the top-5, Stage 2 cannot recover regardless of its reasoning quality. These cases impose a hard error floor attributable entirely to retrieval, and resolving them requires improvements to Stage 1 (Section 8.3).

8.3 Limitations and Future Directions

Stage 1 upper bound. The 6.64% miss rate imposes a hard performance ceiling. Introducing a Stage 1.5 cross-encoder re-ranker (e.g., monoT5 fine-tuned on COLIEE data) could sharpen the precision–recall trade-off without expanding the Stage 2 context window. **Extended reasoning decoding.** Hardware constraints prevented us from using Qwen-3.5-27B's native extended thinking mode. Budget-forced extended reasoning [11] would likely improve Stage 2 accuracy on cases where the legal connection between query and entailing paragraph is non-obvious. **Scaling distillation.** With

only 632 training samples (80% of the 790 gold-filtered instances), the student is data-limited. Generating oracle traces for a larger and more diverse training corpus—potentially augmented with additional COLIEE cycles—and exploring iterative, multi-round teacher–student distillation [18] are natural extensions. **Scaling RL training.** Run 2 was limited to 200 steps on 632 instances. Expanding the training corpus (e.g., with additional COLIEE cycles or data augmentation) and running for more steps would test whether RL can match distillation given sufficient data and compute. The exploration inherent in on-policy RL may enable the model to discover legal reasoning strategies not represented in the teacher's fixed trace distribution.

9 Conclusion

We presented the ClaUSurf system for COLIEE 2026 Task 2 Legal Case Entailment. Our two-stage retrieve-then-reason pipeline pairs a two-phase fine-tuned Qwen-Embedding-3 (4B) retriever (Recall@5 = 93.36% on the development set) with three alternative LLM strategies for entailment classification. Among our submitted runs, off-policy chain-of-thought distillation from a Qwen-3.5-35B-A3B MoE teacher to a Qwen-3.5-9B student achieves the best test-set performance (F1 = 0.3877), outperforming both a 3× larger zero-shot baseline and an RL-tuned peer under equivalent inference cost, and ranking 8th among 14 participating teams in the official COLIEE 2026 Task 2 leaderboard. Our results underscore the effectiveness of oracle-guided reasoning traces as a training signal and indicate that Stage 1 recall remains the binding constraint for further pipeline improvement.

Acknowledgments

The authors thank the COLIEE 2026 organizers for providing the competition data and evaluation framework, and the University of Science, VNU-HCM for computational support.

References

[1] Yoshinobu Kano, Randy Goebel, Mi-Young Kim, Calum Kwan, Ken Satoh, Hiroaki Yamada, and Masaharu Yoshioka. 2026. An Overview of the COLIEE 2026 Competition: Legal Case Law and Statute Law Information Retrieval and Entailment. In *Proceedings of the 21st International Conference on Artificial Intelligence and Law (ICAIL 2026)*.

[2] Randy Goebel, Yoshinobu Kano, Mi-Young Kim, Julien Rabelo, Ken Satoh, and Masaharu Yoshioka. 2025. An Overview of the COLIEE 2025 Competition: Legal Case Law and Statute Law. In *Proceedings of the Workshop of the Competition on Legal Information Extraction/Entailment (COLIEE '25)*. ACM. https://doi.org/10.1145/3769126.3785016

[3] NOWJ Team. 2025. NOWJ@COLIEE 2025: A Multi-stage Framework Integrating Embedding Models and Large Language Models for Legal Retrieval and Entailment. In *Proceedings of the Workshop of the Competition on Legal Information Extraction/Entailment (COLIEE '25)*. ACM.

[4] Tan Minh Nguyen, Trung Vo Hoang, Truong Do Quoc, Phuong Nguyen Thi, and Minh Le Nguyen. 2024. CAPTAIN at COLIEE 2024: Large Language Model for Legal Text Retrieval and Entailment. In *New Frontiers in Artificial Intelligence (JSAI-isAI 2023 Workshops)*, Springer, 126–142.

[5] Qwen Team. 2025. Qwen3 Technical Report. *arXiv preprint arXiv:2505.09388* (2025).

[6] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation Learning with Contrastive Predictive Coding. *arXiv preprint arXiv:1807.03748* (2018).

[7] Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul Bennett, Junaid Ahmed, and Arnold Overwijk. 2021. Approximate Nearest Neighbor Negative Contrastive Estimation for Dense Text Retrieval. In *Proceedings of the 9th International Conference on Learning Representations (ICLR 2021)*.

[8] Jian Hu, Xibin Wu, Zilin Zhu, Xianyu, Wei Shen, Ruyi Gan, and Junping Zhang. 2025. Dr. GRPO: Decomposed Rule-Guided GRPO for LLM Reasoning. *arXiv preprint arXiv:2505.15843* (2025).

[9] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Mingchuan Zhang, Y.K. Li, Y. Wu, and Daya Guo. 2024. DeepSeekMath: Pushing the Limits of Mathematical Reasoning in Open Language Models. *arXiv preprint arXiv:2402.03300* (2024).

[10] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the Knowledge in a Neural Network. In *NIPS Deep Learning and Representation Learning Workshop. arXiv preprint arXiv:1503.02531* (2015).

[11] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. In *Advances in Neural Information Processing Systems 35 (NeurIPS 2022)*.

[12] Rodrigo Nogueira, Zhiying Jiang, Ronak Pradeep, and Jimmy Lin. 2020. Document Ranking with a Pretrained Sequence-to-Sequence Model. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, 708–718.

[13] Stephen Robertson, Steve Walker, Susan Jones, Micheline Hancock-Beaulieu, and Mike Gatford. 1994. Okapi at TREC-3. In *Proceedings of the Third Text REtrieval Conference (TREC-3)*, 109–126.

[14] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of NAACL-HLT 2019*, 4171–4186.

[15] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training Language Models to Follow Instructions with Human Feedback. In *Advances in Neural Information Processing Systems 35 (NeurIPS 2022)*.

[16] DeepSeek-AI. 2025. DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning. *arXiv preprint arXiv:2501.12948* (2025).

[17] Cheng-Yu Hsieh, Chun-Liang Li, Chih-Kuan Yeh, Hootan Nakhost, Yasuhisa Fujii, Alexander Ratner, Ranjay Krishna, Chen-Yu Lee, and Tomas Pfister. 2023. Distilling Step-by-Step! Outperforming Larger Language Models with Less Training Data and Smaller Model Sizes. In *Findings of the Association for Computational Linguistics: ACL 2023*, 8003–8017.

[18] Subhabrata Mukherjee, Arindam Mitra, Ganesh Jawahar, Sahaj Agarwal, Hamid Palangi, and Ahmed Awadallah. 2023. Orca: Progressive Learning from Complex Explanation Traces of GPT-4. *arXiv preprint arXiv:2306.02707* (2023).

[19] Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-Consistency Improves Chain of Thought Reasoning in Language Models. In *Proceedings of the 11th International Conference on Learning Representations (ICLR 2023)*.

JNLP at COLIEE 2026: Multi-Stage Hybrid LLM Framework for Case Law Retrieval, Statute Entailment, and Tort Prediction

Dang Le*, Cuong Hoang*, Duy-Minh Nguyen-Tran*, Nguyen Khac Vu Hiep*, Tan-Minh Nguyen*, Do Minh Duc, Son T. Luu, Trung Vo, An Trieu, Nguyen Ngoc Minh, Dat Nguyen, The-Hai Nguyen, Trang Pham, Nguyen-Khang Le, Dinh-Truong Do, Vu Tran, and Le-Minh Nguyen
{lnhdang14,cuonghoang,ntdminh,hiep.nkv,minhnt}@jaist.ac.jp
Japan Advanced Institute of Science and Technology (JAIST)
Ishikawa, Japan

Abstract

This paper presents the JNLP team's approach to addressing these challenges in the Competition on Legal Information Extraction and Entailment (COLIEE) 2026. We address all four main tasks: case law retrieval, case law entailment, statute law retrieval, and statute law entailment, alongside the pilot tasks. For Legal Case Retrieval (Task 1), we improve upon previous approaches by reducing the search space via BM25 and utilizing alternative classification models. For Legal Case Entailment (Task 2), we propose a two-stage architecture that combines dense retrieval with LLM-based reasoning to accurately identify entailing paragraphs. For Statute Retrieval and Entailment (Task 3), we employ a three-stage retrieval pipeline integrated with LLMs, incorporating dynamic few-shot prompting and a multi-agent debate framework. For Legal Textual Entailment (Task 4), we utilize Japanese-specific LLMs with few-shot prompting and self-consistency through majority voting to assess legal validity. Finally, we formulate both Tort Prediction and Rationale Extraction as binary classification tasks, leveraging LLMs to analyze the legal context and predict the target labels. Experimental results highlight the efficiency of our proposed approaches. Notably, we achieved the highest ranking on the pilot task, third place in Task 3, and second place across all other tasks.

CCS Concepts

• **Information systems** → *Expert systems.*

Keywords

Legal Information Retrieval and Entailment, Natural Language Processing in Law, Large Language Model for Legal, COLIEE 2026

ACM Reference Format:
Dang Le*, Cuong Hoang*, Duy-Minh Nguyen-Tran*, Nguyen Khac Vu Hiep*, Tan-Minh Nguyen*, Do Minh Duc, Son T. Luu, Trung Vo, An Trieu, Nguyen Ngoc Minh, Dat Nguyen, The-Hai Nguyen, Trang Pham, Nguyen-Khang Le, Dinh-Truong Do, Vu Tran, and Le-Minh Nguyen. 2026. JNLP at COLIEE 2026: Multi-Stage Hybrid LLM Framework for Case Law Retrieval, Statute Entailment, and Tort Prediction. In *Proceedings of COLIEE 2026 workshop, June 12, 2026, Singapore, Singapore.* ACM, New York, NY, USA, 10 pages.

COLIEE 2026, June 12, 2026, Singapore

1 Introduction

With the rapid expansion of digital legal text, the need for efficient retrieval and accurate legal reasoning systems has become increasingly important. To facilitate this, the Competition on Legal Information Extraction and Entailment (COLIEE) aims to build a research community dedicated to addressing complex legal challenges [12, 14, 25]. Organized to push the boundaries of legal artificial intelligence, the 2026 competition challenges participants across four core tasks alongside a pilot task. Task 1 focuses on identifying existing legal cases that support a given query case. Task 2 requires identifying a paragraph from an existing case that can entail the decision of a new case. Task 3 focuses on retrieving articles that are relevant for a yes/no legal question. Task 4 involves determining whether relevant Civil Law articles retrieved for a legal question logically entail. Finally, the Pilot Task tackles Tort cases via two subtasks: predicting final decisions by synthesizing facts, claims, and arguments in Tort Prediction task, and classifying whether specific premise-claim pairs are accepted or rejected in Rationale Extraction task.

However, automating these tasks is difficult because legal documents have complex structures, domain-specific terminology, and often implicit reasoning steps [24]. To overcome these barriers, researchers have increasingly turned to Natural Language Processing (NLP) technologies. Large Language Models (LLMs), in particular, have demonstrated remarkable capabilities in legal reasoning and statutory analysis. Within the COLIEE, participants frequently leverage these models alongside sophisticated prompt engineering techniques—such as few-shot prompting and Chain of Thought (CoT) strategies to handle complex logical entailment [26, 27]. However, applying LLMs directly to lengthy precedents and complex frameworks often introduces new challenges, including logical hallucination and contextual overload. Therefore, there is a need to systematically explore how integrating retrieval pipelines with prompt engineering strategies can maximize both computational efficiency and reasoning accuracy.

To address these challenges, this paper presents the JNLP team's hybrid LLM-based framework for COLIEE 2026. We investigate the application of advanced prompt engineering and optimized retrieval strategies to enhance legal reasoning capabilities across the competition's objectives. For Task 1, we improve upon previous approaches by optimizing a pairwise similarity ranking framework using BM25-based search space reduction and robust classification models. For Task 2, we introduce a two-stage architecture merging

* Equal Contribution

instruction-guided dense retrieval with structured LLM reasoning to accurately isolate entailing paragraphs. For Task 3, we deploy a three-stage retrieval pipeline paired with open-weight LLMs, mitigating hallucination through a Multi-Agent Debate (MAD) paradigm and dynamic few-shot prompting. For Task 4, we leverage high-performance reasoning LLMs, employing consistency checking and label-balanced few-shot prompting to rigorously evaluate legal validity. Finally, for the Pilot Tasks, we address both tort prediction and rationale extraction by fine-tuning a multilingual LLM and employing a dual-agent prompting framework to simulate opposing legal arguments based on the same facts. By combining highly efficient retrieval mechanisms with the legal reasoning power of LLMs, our hybrid framework achieved top-tier results across the competition's tasks.

2 Related Works

Task 1. With the widespread adoption of LLMs, a plethora of techniques have emerged in the field of Information Retrieval. Generative Transformer models like LEGAL-BERT [8] are also being used to assist legal NLP research and legal technology applications. TF-IDF (Term Frequency-Inverse Document Frequency) provides the basis for another popular method for information retrieval in systems with limited GPU resources. For example, [17] demonstrated the effective application of the TF-IDF weighting method for information retrieval on a website, which offers practical insights into developing efficient and accurate retrieval systems for large-scale text data. SAILER [19] achieves significant improvements in legal case relevance assessment by incorporating a structural understanding of legal documents. These developments lay the foundation of our approach towards the information retrieval tasks in COLIEE 2026.

Task 2. In 2022, the JNLP team [5] placed second with a multi-component system that combined score fusion from LegalBERT and BM25 with Abstract Meaning Representation (AMR) to align key terms between queries and paragraphs. In 2023, the CAPTAIN team [27] achieved state-of-the-art results by fine-tuning MonoT5 using hard negative sampling and applying ensemble techniques. The THUIR team [18] employed lexical baselines (BM25 and QLD), fine-tuned language models with contrastive loss, and combined these scores through ensembling for final predictions. In COLIEE 2024, the AHMR team [28] proposed two approaches: (1) fine-tuning a LegalBERT [7] model using triplet loss [33] on the official training data for task 2, and (2) fine-tuning a MonoT5 [29], pretrained on MSMARCO [3], utilizing hard negative samples, selected via BM25 and an alternative MonoT5 variant.

Task 3. Historically, BERT variants and data augmentation have been highly effective. LLNTU [34] won in 2020 using a BERT ensemble, while OvGU [40] (2021) and HUKB [43] (2022) secured first place through Sentence-BERT augmentation and judicial interpretation-based corpus expansion, respectively. In 2023, CAPTAIN [27] won by ensembling Japanese BERT and MonoT5 with diverse data filtering. By 2024, methods increasingly integrated LLMs for refinement. AHMR [28] ranked candidates using a fine-tuned MonoT5 [29] followed by LLM post-processing. Concurrently, CAPTAIN [27] paired a hard-negative-tuned MonoT5 classifier with zero- and few-shot FLAN-T5 prompting to capture complex relationships among candidate paragraphs.

Task 4. Methods for this task have evolved from classification models to generative LLM prompting. In 2021, HUKB [42] utilized BERT with data augmentation on judicial decisions, while OvGU [40] proposed a BERT-embedded graph neural network. In 2022, JNLP [5] evaluated multiple LMs with negation-based augmentation, and LLNTU [21] introduced subsequence-similarity models using disjunctive union strings. A shift occurred in 2023 when JNLP [6] adopted generative entailment prediction using zero-shot LLMs (Flan-T5, Alpaca-T5), employing ranking-based voting and thresholding across prompt strategies. Building on this in 2024, CAPTAIN [27] further enhanced Flan-T5-XXL performance through few-shot in-context learning, Automatic Chain-of-Thought (Auto-CoT) coupled with Dense Passage Retrieval, and LLM-driven data augmentation via summaries and synthetic hypotheses.

Pilot Tasks. The automated analysis of legal judgments has evolved significantly over decades. Initial approaches relied on statistical methods, later shifting towards machine learning techniques focused on text classification and feature extraction from case documents or metadata [16]. These earlier methods often required substantial manual effort and struggled to generalize effectively [15]. More recently, researchers have leveraged neural networks, informed by natural language processing advances and legal knowledge, to develop models for specific tasks like charge prediction or generating court views [41]. However, adapting these advanced, often specialized, neural models to handle the full complexity and intricate dependencies inherent in broader legal judgment prediction remains a key challenge.

3 Problem Formalization

3.1 Task 1: Legal Case Retrieval

The primary goal of this task is to automatically identify and retrieve prior legal cases (noticed cases) that serve as relevant precedents for a new query case. Let Q denote a query case. Let $C = \{C_1, C_2, \ldots, C_n\}$ denote a comprehensive corpus of candidate target cases. The objective is to identify a specific subset of target cases $N \subset C$ that are actually cited by Q to support its legal propositions.

3.2 Task 2: Legal Case Entailment

The task is to identify which paragraph(s) in a noticed case N entail a given decision Q from a query case. Here, Q is a textual representation of a judicial conclusion, and N consists of paragraphs $P = p_1, \ldots, p_m$.

In COLIEE, the goal is to select a subset $\tilde{P} \subset P$ whose paragraphs semantically support Q. Although multiple paragraphs may be relevant, typically one (or a few) is considered central. The objective is to accurately identify the paragraph(s) that best entail Q, effectively linking a decision to its supporting precedent.

3.3 Task 3: Statute Retrieval and Entailment for Yes/No Questions

This task focuses on Statute Law Question Answering (SL-QA), where the goal is to retrieve relevant legal articles and determine

a Yes/No answer for a given query Q. Given a set of candidate statutes $\{A_1, \ldots, A_n\}$, an article A is considered relevant if it provides sufficient information to entail the answer to Q.

The task requires jointly performing retrieval and entailment reasoning, as queries may depend on multiple articles. The system must output both (1) a subset of relevant articles and (2) a final binary decision (Yes/No) indicating whether the query is legally supported.

3.4 Task 4: Legal Textual Entailment

The goal of this task is to develop a question answering system for legal queries, based on entailment from relevant legal articles, with answers restricted to "Yes" or "No". To determine the correct answer, the system must align the conditions described in the law with those presented in the query and infer the outcome based on the legal consequences stated in the articles. However, the complexity of legal language and the limited availability of training data make it challenging to accurately determine the entailment relationship between queries and legal texts.

3.5 Pilot Task: Tort Prediction

We formulate tort prediction based on three distinct input components: Undisputed Facts (F), Plaintiff's Claims (P), and Defendant's Arguments (D). Let these components be represented as sets of textual elements such that $F = \{f_1, f_2, \ldots, f_n\}$, $P = \{p_1, p_2, \ldots, p_m\}$, and $D = \{d_1, d_2, \ldots, d_k\}$, where $n, m, k \in \mathbb{N}$. The objective is to leverage these sets to predict the final tort decision.

3.6 Pilot Task: Rationale Extraction

We formulate this task as a binary classification problem based on an input pair. Let the input pair be (F, C), where $F = \{f_1, f_2, \ldots, f_n\}$ represent Undisputed Facts and $S \in \{P \cup D\}$ denotes a single statement from either the Plaintiff's Claims (P) or the Defendant's Claims (D). Given this pair, the objective is to predict whether the claim is accepted or rejected.

4 Methodology

4.1 Task 1

Modern information retrieval tasks rely heavily on measuring the embedding similarity between a query and a target document. However, the substantial length of legal cases makes retrieval based solely on embeddings both computationally and conceptually challenging. To address this, we adapt a pairwise similarity ranking framework, originally proposed by team UMNLP [11] in the 2024 competition—which models retrieval as a binary classification task.

This approach preprocesses the text to extract statistical and semantic features from each case. It then compares these features to generate a numerical feature set for every query-candidate pair, which is subsequently fed into a Multilayer Perceptron (MLP) to predict relevance. During training, pairs where the candidate case was actually cited by the query case are assigned a label of 1, while all other pairs are labeled 0. The foundational hypothesis of this approach is that, much like academic citations, court judgments cite other cases to support specific statements (propositions) rather than the case as a whole. In the dataset, actual citations within query cases are replaced with specific placeholders (e.g., **FRAGMENT_SUPPRESSED**). To leverage this, the baseline uses a fine-tuned T5 transformer model [31] to condense the context surrounding each placeholder into a short, third-person summary.

Given the proven success of this methodology—achieving top-2 [11] and top-1 [12] placements in the 2024 and 2025 competitions, respectively, we leverage it as the foundation for our solution. However, mapping every query to all potential target cases creates a severe computational bottleneck. Pairing $2,001$ queries with $5,546$ potential targets yields over 11 million training pairs ($11,097,546$). Furthermore, this induces an extreme class imbalance, with a positive-to-negative ratio of roughly $1 : 1419$. To make training more feasible and effective, we introduce some improvements to this baseline method. First, we apply BM25 to filter the target pool, selecting only the top 500 candidate pairs per query. This drastically reduces the training space while empirically maintaining a high recall of 72.32%. Second, to resolve the remaining class imbalance within these filtered pairs, we apply positive oversampling. Finally, we replace the standard MLP with more robust classification models, specifically Random Forest and XGBoost. Combined with the reduced search space, these improvements can decrease training time while boosting overall performance.

4.2 Task 2

We propose a Two-stage Retrieval and Reasoning Framework that integrates dense retrieval with LLM-based legal reasoning to accurately identify paragraphs that entail a given decision. The framework is designed to first narrow down the search space using semantic similarity and then perform fine-grained legal analysis to determine entailment.

Stage 1: Dense Retrieval. Given a query decision q and a set of paragraphs $R = \{p_1, p_2, \ldots, p_n\}$ from the noticed case, we first retrieve a subset of candidate paragraphs using a dense embedding model. Both the query and candidate paragraphs are encoded into a shared semantic space, where their relevance is measured via vector similarity.

To better align the retrieval process with the task objective, we adopt an instruction-guided encoding strategy. This explicitly frames the query as a search for legal reasoning that supports the decision, allowing the embedding model to capture not only topical similarity but also implicit reasoning signals. Based on the similarity scores, the top-k most relevant paragraphs are selected as candidates for the next stage. This retrieval step serves as an efficient filtering mechanism that maintains high recall while significantly reducing the number of paragraphs requiring deeper analysis.

Stage 2: LLM-based Reasoning. In the second stage, we employ a Large Language Model (LLM) to perform fine-grained entailment classification over the retrieved candidates. Rather than relying solely on similarity scores, the LLM is prompted to analyze the legal relationship between the decision and each candidate paragraph through a structured reasoning process.

Specifically, the model is first guided to identify the core legal principle underlying the decision and then to examine whether a

candidate paragraph expresses an equivalent or logically supporting principle. It is further encouraged to distinguish between essential reasoning and non-essential statements when making its judgment. Based on this structured analysis, the LLM produces a binary decision indicating whether each paragraph entails the given decision. This stage enables a deep semantic and logical understanding that goes beyond surface-level matching, which is crucial for complex legal entailment.

Decision Strategy. The final prediction is determined by prioritizing the paragraphs classified as positive by the LLM. When no paragraph is confidently selected, we use a fallback mechanism that leverages retrieval-stage similarity scores to ensure robustness. This hybrid strategy effectively balances the precision of LLM-based reasoning with the coverage of embedding-based retrieval.

4.3 Task 3

4.3.1 Retrieval. We propose a three-stage retrieval and ranking pipeline that balances efficiency and accuracy. First, dense embeddings enable large-scale semantic retrieval, followed by a learning-based model that filters high-recall candidates. Second, a neural re-ranker refines this set by assigning relevance scores, improving precision. Finally, a cross-encoder performs fine-grained relevance assessment through joint query‑document encoding. Predictions are aggregated at the query level using either a greedy strategy or an F2-optimized ensemble of LLMs with tuned weights, enhancing overall relevance estimation.

4.3.2 Binary Question Answering. To address the legal entailment and question-answering requirements of Task 3, we utilized state-of-the-art open-weight models, specifically Qwen2.5-72B-Instruct and a Japanese-enhanced Llama model. To manage computational efficiency, all models were deployed using 4-bit quantization.

Baseline Zero-Shot Approach. Our first configuration serves as a baseline to evaluate the zero-shot reasoning capabilities of LLMs. The system is provided with the query and the retrieved relevant articles in English or Japanese, and is prompted to provide a "Yes" or "No" answer based strictly on the provided legal text without prior examples. The zero-shot instruction in Japanese is adopted from prior state-of-the-art work [30].

Prompt Instruction: Zero-shot in English

Question: {hypothesis} Legal articles: {premise} Based on the given legal articles, answer Yes or No.

Prompt Instruction: Zero-shot in Japanese

関連条文に基づいて、問題文が正しいか誤っているかを解答してください。
問題文が正しい場合は「解答: 正しい」、誤っている場合は「解答: 誤り」と解答してください。
関連条文:{premise}
問題文:{hypothesis}

Similarity-Based Few-Shot Prompting. To provide stronger in-context guidance, the second configuration adopts a similarity-based few-shot prompting strategy. Instead of using fixed examples, the system dynamically retrieves relevant demonstrations by computing vector embeddings for both the test query and the training dataset. The three training instances with the highest cosine similarity to the query are selected and incorporated into the prompt. This dynamic selection ensures that the examples reflect legal reasoning patterns closely aligned with the query context.

Prompt Instruction: Few-shot in English

Instruction
Based on the relevant statute, please determine whether the statement is correct or incorrect.
(If the statement is correct, answer with Yes; if incorrect, answer with No)

Examples
{example_1}. Answer: {label_1}.
{example_2}. Answer: {label_2}.
{example_3}. Answer: {label_3}

Input
{test_input}
Answer:

Multi-Agent Debate and Consensus. Our final configuration employs a multi-agent debate paradigm to improve reasoning robustness and reduce hallucination. The approach conducts a single-round debate between two heterogeneous agents, allowing the system to benefit from differences in architecture and pre-training. The agents are prompted using a combination of zero-shot and 3-shot settings in both English and Japanese, capturing complementary linguistic and reasoning perspectives. This process is repeated to generate three independent outputs, and a final decision is obtained through majority voting, leading to a more reliable "Yes" or "No" prediction.

Prompt Instruction: Multi-Agent Debate in English

Instruction
Based on the relevant statute, please determine whether the statement is correct or incorrect.
(If the statement is correct, answer with Yes; if incorrect, answer with No)

Question: {hypothesis}
Legal articles: {premise}

These are the recent opinions from other agents:
One of the agents' response: {agent_1}
Another agent's response: {agent_2}

Use these opinions carefully as additional advice to revise your recent opinion to give your final answer to the question:
Question: {hypothesis}
Legal articles: {premise}
Final Answer:

4.4 Task 4

To address this, we design a system that takes a legal query Q and a set of relevant statute articles A as input. The system processes these inputs through a function f, producing an internal representation $X = f(Q, A)$. Based on X, the system generates a binary

answer $Y \in \{\text{“Yes”}, \text{“No”}\}$. The objective is to maximize the likelihood of generating a correct answer, formalized as maximizing the conditional probability $P(Y \mid X)$.

Pair ID: H25-2-I
Label: N
Query (Q):
被保佐人が，保佐人の同意を得て，自己の不動産につき第三者との間で売買契約を締結したときは，被保佐人がその売買契約の要素について錯誤に陥っており，かつ，そのことにつき重大な過失がない場合でも，その契約の無効を主張することができない。
Referenced Article -ID 95 (A):
第九十五条（錯誤）意思表示は、次に掲げる錯誤に基づくものであって、その錯誤が法律行為の目的及び取引上の社会通念に照らして重要なものであるときは、取り消すことができる。

Our approach to Task 4 focuses on leveraging high-performance Japanese-specific LLMs through ensemble techniques and reasoning-enhanced prompting. Based on an initial survey of 20 LLMs, we identified *ABEJA-QwQ32b-Reasoning-Japanese-v1.0* as the state-of-the-art baseline for Japanese legal reasoning due to its specialized training on reasoning traces.

4.4.1 Model Selection and Preliminary Evaluation. Before finalizing our submission, we conduct a comprehensive evaluation of approximately 20 open-weight LLMs using previous COLIEE Task 4 datasets (R04, R05, and R06). Based on strong zero-shot legal entailment performance, we select a core ensemble consisting of **ABEJA-QwQ32b-Reasoning-Japanese-v1.0**, **ABEJA-Qwen2.5-32b-Japanese-v1.0**, and **Qwen2.5-32B-Instruct**, which together provide complementary strengths in reasoning, instruction following, and multilingual capability.

4.4.2 Run 1: Self-Consistency via Majority Voting. The first run (`JNLP1`) employs a self-consistency strategy to mitigate the stochastic nature of LLM outputs. Using the `ABEJA-QwQ32b-Reasoning` model, we perform five independent iterations of zero-shot prompting for each test case. The final answer is determined by a simple majority vote among the five outputs. This method ensures that the final prediction is not an outlier but a consistent conclusion reached across multiple reasoning paths.

4.4.3 Run 2: Weighted Majority Voting Ensemble. The second run (`JNLP2`) utilizes a heterogeneous ensemble strategy. By combining models with different architectural biases, we aim to capture a broader range of legal interpretations.

$$Final\_Decision = \text{argmax}_{d \in \{Y,N\}} \sum_{i=1}^{3} w_i \cdot P_i(d) \tag{1}$$

Where P_i is the prediction from model i, and w_i represents the predefined weight assigned based on the model's performance in preliminary testing. Each model generates its prediction using zero-shot prompting, and the weighted results are aggregated to yield the terminal decision.

4.4.4 Run 3: Dynamic Few-Shot with Label Balancing. The third run (`JNLP3`) extends beyond zero-shot prompting by incorporating dynamically selected few-shot examples. For each query, similar cases are retrieved using BM25 from the training data, and three examples are selected to provide contextual guidance. To avoid prediction bias, the examples are balanced between positive and negative labels. The prompting process is repeated three times, and the final answer is determined via majority voting, improving both stability and generalization.

4.5 Pilot Task: Tort Prediction

This task necessitates predicting the judicial decision (tort prediction) conditioned on the undisputed facts (U) and the respective arguments presented by both parties (P for plaintiffs and D for defendants). Our methodology employs a synergistic approach of advanced prompting strategies combined with the task-specific fine-tuning of Qwen2.5-Instruct model variants. The Qwen architecture is widely recognized as a state-of-the-art multilingual model family, exhibiting exceptional proficiency in comprehending diverse languages, including English, Chinese, and Japanese. This robust linguistic capability empowers the models to profoundly comprehend and meticulously analyze the complex semantic structures of the input prompts. To ensure the model accurately adopts the adversarial context of civil litigation, the prompt is systematically segmented into four distinct informational blocks:

Prompt Instruction: Prompt Instruct

Instruction
あなたは裁判官です。民事事件を担当することになるでしょう。あなたの任務は、原告と被告の主張に基づき、不法行為が認められるかどうかを判断することです。答えは「True」か「False」のいずれかでなければならず、それ以外は問いません。対象となる事件は以下のとおりです。

User prompt
議論の余地のない事実:
{UNDISPUTED FACTS}
原告の主張：
{PLAINTIFF CLAIMS}
被告の主張：
{DEFENDANT ARGUMENTS}

Answer:
裁判所の判決:

To optimize the Qwen2.5-Instruct model's legal reasoning, the prompt utilizes a strictly compartmentalized hierarchy designed to minimize semantic entanglement. It begins by assigning a judicial persona to evaluate tort recognition, constraining the output to a binary ”True” or ”False” classification. The adversarial input is then systematically divided into three distinct components: an objective baseline of Undisputed Facts (U), the legal assertions comprising the Plaintiff's Claims (P), and the rebuttals within the Defendant's Arguments (D), with a default placeholder utilized whenever an

argument list is empty. By separating agreed-upon facts from competing legal theories, this rigid structure enables the model to effectively weigh opposing arguments against an established factual foundation.

4.6 Pilot Task: Rationale Extraction

This task requires modeling adversarial legal reasoning through a dual-agent prompting framework, where two independent roles—the Plaintiff and the Defendant—generate arguments condition-ed on a shared factual basis. Unlike conventional single-agent prompting, our approach introduces controlled divergence at the system instruction level while preserving identical user-level inputs. This design enables a systematic comparison of opposing reasoning strategies under equivalent informational conditions.

Prompt Instruction: Prompt Instruct

Instruction
Distinct Prompt Instructions for the Plaintiff and the Defendant

User prompt
【争いのない事実】
{facts text}
【評価対象の主張】
{claim description}
この主張は裁判所に認容されますか？「はい」または「いいえ」で答えてください。

The proposed framework facilitates the rigorous analysis of adversarial legal reasoning by systematically decoupling role-specific interpretations from uniform informational inputs. At the system level, models receive asymmetrical instructions: a Plaintiff prompt designed to assert liability and damages, and a Defendant prompt focused on refuting claims and challenging causality. To isolate the effects of this role conditioning, both agents process an identical user prompt consisting of a contextual Case Description and a neutral List of Claims (C, comprising $Claim_1, \ldots, Claim_n$). The interaction mechanism initializes the agents with their divergent system roles, feeds them the shared user prompt, and collects their independent outputs for subsequent evaluation. By ensuring that variations in generated responses stem exclusively from the assigned argumentative stances rather than input discrepancies, this architecture strikes a controlled balance between structural consistency and role-induced analytical diversity.

5 Experiments and Results

5.1 Task 1

5.1.1 Settings. We adopt the feature extraction pipeline, isolating the text surrounding citation placeholders (FRAGMENT_SUPPRESSED). We train a T5 transformer [31] to summarize these contexts into propositions, which are then mapped into a dense vector space using all-mpnet-base-v2 [13] to serve as our core semantic features. To mitigate class imbalance and optimize training, we apply BM25 implemented via Python's pyserini [20] to restrict the massive candidate pool to the top 500 target cases per query.

For the official evaluation, we submit three runs that share the same feature extraction pipeline. During inference, BM25 is again applied to filter the test pool, retaining the top 200 candidates per query. **Run 1 (JNLP_random_forest)** employs a Random Forest model for final binary classification, while **Run 2 (JNLP_xgboost)** replaces it with an XGBoost classifier to capture more complex decision boundaries. Finally, **Run 3 (JNLP_ensemble)** combines predictions from Random Forest, XGBoost, and a Multilayer Perceptron (MLP) through an ensemble strategy, aiming to improve robustness and overall performance.

Table 1: Leaderboard results on the Legal Case Retrieval task (ranked by individual runs).

Rank	Team	Run	F1	Precision	Recall
1	NOWJ	submission2	**0.4220**	0.4235	**0.4206**
3	JNLP	random_forest	0.4126	**0.4341**	0.3931
4	JNLP	ensemble	0.4040	0.4174	0.3914
5	JNLP	xgboost	0.4000	0.4059	0.3943
7	SIL	submission_sil2	0.3871	0.4186	0.3600
9	mezza	task_1_mezzanino	0.3289	0.3161	0.3429
10	INTIT	3intit_bm25_year	0.3165	0.3249	0.3086
11	DU	du3	0.3141	0.2945	0.3366

5.1.2 Result. Table 1 presents the official leaderboard results for Task 1. Our best-performing configuration, **JNLP_random_forest**, achieved third place overall with an F1 score of 0.4126. Notably, despite not achieving the top overall F1 score, our system successfully attained the highest Precision (0.4341) among all submissions in the competition. Our alternative configurations, **JNLP_ensemble** and **JNLP_xgboost**, also demonstrated robust performance with F1 scores of 0.4040 and 0.4000, respectively.

This high precision and lower recall are a direct result of our search space reduction strategy. While we trained our models on the top 500 candidates, and restricted the test pool to only the top 200 candidates per query. This strict filtering successfully eliminates false positives (boosting precision) but naturally limits the maximum possible recall.

By strictly limiting the candidate pool during testing, our framework dramatically decreases the time required for the models to extract features and perform pairwise classifications. In real-world legal information retrieval systems, this efficiency and reduced inference latency are highly advantageous, demonstrating that our pipeline can isolate relevant legal precedents with high confidence while conserving computational resources.

5.2 Task 2

5.2.1 Settings. We evaluate our two-stage legal entailment pipeline across the following configurations to ensure a balance between high recall retrieval and high precision reasoning. In the first stage, we employ the pre-trained embedding model `nvidia/NV-Embed-v2` to encode both the query decision and candidate paragraphs into a shared semantic space. To better align the retrieval objective with legal reasoning, we adopt an instruction-guided encoding strategy that frames the query as a search for supporting legal arguments. Candidate paragraphs are ranked using cosine similarity, and we experiment with different values of k to control the number of retrieved paragraphs passed to the reasoning stage. For the second stage, we utilize `DeepSeek-R1-Distill-Qwen-32B`

as the primary large language model for entailment classification. The model is prompted with a structured zero-shot instruction designed to guide legal reasoning, including identifying the decision's core principle and assessing whether a candidate paragraph provides supporting analysis. The temperature is set to 0.3 to ensure deterministic outputs, and the maximum generation length is set to 4096 tokens to accommodate detailed reasoning.

Table 2: Leaderboard results on the Legal Case Entailment task (ranked by individual runs).

Rank	Team	Run	F1	Precision	Recall
1	IAI	task2_run2	**0.4899**	0.4501	**0.5374**
2	AIIRLab	task2_aiirfusion3	0.4706	**0.5120**	0.4354
3	JNLP	submission (3)	0.4507	0.4174	0.4898
5	JNLP	submission (2)	0.4457	0.3879	0.5238
6	NOWJ	nowj003	0.4429	0.7037	0.3231
9	UA	result_run1_ua	0.4254	0.4711	0.3878
10	JNLP	submission (1)	0.4254	0.4711	0.3878

5.2.2 Result. Table 2 presents the official results of the Legal Case Entailment task. Our approach ranks among the top-performing systems, with both submissions achieving competitive performance compared to other leading teams. In particular, our best configuration secures a top-3 position, demonstrating the effectiveness of the proposed retrieval-and-reasoning framework.

A key observation from the results is that using a smaller set of candidate paragraphs for LLM-based reasoning leads to better overall performance. While increasing the number of retrieved candidates can improve coverage, it also introduces significant irrelevant or weakly related evidence. This additional noise makes it more difficult for the LLM to accurately identify the core legal reasoning, as it must process a larger set of potentially conflicting signals. In contrast, a more selective candidate set enables the model to focus on the most relevant paragraphs, resulting in more precise entailment predictions.

This finding highlights an important trade-off between retrieval coverage and reasoning effectiveness. Although dense retrieval is designed to preserve high recall, passing too many candidates to the reasoning stage can negatively impact performance due to context interference. The results suggest that LLMs benefit from more constrained, focused input, with the most relevant evidence prioritized over exhaustive inclusion.

5.3 Task 3

5.3.1 Settings. For the retrieval task, we used BGEM3Flag [9] as the embedding model for both articles and queries. For the pretrained ranking model, we used RankLlama-7B-passage [22], which is fine-tuned on the training split of MS MARCO Passage Ranking [2]. On the final stage, we conducted experiments on various of series of LLMs: Gemma-2 [35], E5-Mistral-7B-Instruct [38, 39], Phi-3-Medium-4K-Instruct [23], Qwen2.5 [36], Qwen3 [37], and Llama3 [1]. During the aggregation step, we performed two separate runs, with one optimized for the F2-score and the other run utilized the greedy approach.

Table 3: Task 3 Official Leaderboard: Comparison of JNLP Runs with Top Team Results (ranked by individual runs).

Rank	Team	Run	Acc.	Corr.	Prec.	Rec.
1	NOWJ	NOWJ_run1	**95.12**	**78**	0.0549	**1.0000**
2	JNLP	JNLP1	90.24	74	0.5151	**1.0000**
5	cody	codyrag	85.37	70	0.0439	**1.0000**
6	Bengo	BengoR2E2	82.93	68	0.8364	0.9797
7	HUKB	HUKBmix	81.71	67	**0.9431**	0.9553
12	SCIlab	SCIlab1	80.49	66	0.0132	0.9268
13	JNLP	JNLP2	79.27	65	0.5151	**1.0000**
14	DU	DU2	79.27	65	0.0874	0.9654
16	JNLP	JNLP3	78.05	64	0.5151	**1.0000**
19	DU1	DU1	75.61	62	0.0707	0.9776
20	AIIR	AIIRjpgen	58.54	48	0.1707	0.7785

5.3.2 Results. In the retrieval task, our methods achieve perfect recall (1.0) but only moderate precision, indicating a tendency toward over-retrieval. This results in competitive but not top-tier F2 performance, with JNLP-F2 outperforming JNLP-greedy due to better precision.

Our framework, JNLP, demonstrates an optimized balance between retrieval precision and reasoning accuracy, with the JNLP1 submission securing Rank 2 overall, presented in Table 3. While the top-ranked system attained higher accuracy, it did so with significantly lower precision compared to JNLP. This indicates that whereas the winning system utilized a broad, over-inclusive retrieval strategy, the JNLP methodology identifies relevant statutes with greater surgical efficiency, minimizing noise for the reasoning component. Furthermore, the perfect recall across all runs validates the robustness of the retrieval mechanism. The observed variance in final accuracy suggests that terminal performance is highly sensitive to the specific prompting strategies and model configurations employed during the legal entailment phase.

The experimental results demonstrate that the choice of retrieval strategy and prompting methodology significantly influences terminal QA performance. Primarily, the F2-opt retrieval method consistently outperformed or matched the greedy baseline across all tested configurations (e.g., Z1 and F6), validating that superior document retrieval directly facilitates more accurate downstream entailment. A comparison between prompting strategies reveals that Few-Shot methods generally surpass Zero-Shot approaches, particularly when paired with reasoning-optimized models; notably, the ABEJA-QwQ32B model saw a substantial accuracy increase from 84.15% in Z4 to 93.90% in F8 when provided with contextual examples. Furthermore, the Multi-Agent Debate (MAD) framework consistently exceeded the performance of its individual candidate models. This is most evident in the M11 configuration, which achieved 90.24% accuracy, significantly higher than the 80% accuracy of its constituent agents (F6 and F9). These findings suggest that the collaborative reasoning process inherent in MAD effectively mitigates the limitations of standalone models, leading to a more robust final consensus.

Table 4: Task 3 QA Results: Comparison by Prompting Method and Retrieval Strategy

ID	Configuration	Lang	Accuracy (%) F2-opt	Greedy
Zero-Shot (ZS) Methods				
Z1	Qwen2.5-72B	en	80.49	78.05
Z2	ABEJA-QwQ32B	en	89.02	–
Z3	Swallow-70B	en	–	58.54
Z4	ABEJA-QwQ32B	ja	84.15	–
Z5	Swallow-70B	ja	80.49	80.49
Few-Shot (FS) Methods				
F6	Qwen2.5-72B	en	80.49	79.27
F7	ABEJA-QwQ32B	en	84.15	–
F8	ABEJA-QwQ32B	ja	**93.90**	–
F9	Swallow-70B	ja	80.49	80.49
Multi-Agent Debate (MAD) Methods				
M10	Qwen2.5-72B (F7+F8)	en	**93.90**	–
M11	Qwen2.5-72B (F6+F9)	en	90.24	90.24
M12	Qwen2.5-72B (Z1+F9)	en	86.59	87.80
M13	Qwen2.5-72B (Z1+Z5)	en	85.37	85.37
M14	ABEJA-QwQ32B (Z4+F8)	ja	92.68	–
M15	ABEJA-Qwen2.5-32B (Z4+F8)	ja	90.24	–

5.4 Task 4

5.4.1 Setting. In the survey phase, we perform a comprehensive evaluation of multiple prompting strategies across a wide range of LLMs. The evaluated approaches include Zero-shot, Few-shot[4], multi-agent debate[10], and multi-step reasoning[32].

We conduct experiments on models of different scales, spanning from small-sized models to mid-sized models (approximately 7B parameters) and large-scale models (up to 72B parameters). This experimental setup enables us to systematically analyze the impact of both the model scale and prompting strategy on performance in the legal entailment task.

In addition, we investigate the effect of the inference config quantization models with full inference. This comparison aims to examine the trade-off between computational efficiency and predictive performance.

Table 5: Zero-shot Accuracy (%) of Candidate Models on Previous Task 4 Datasets

Model	R06	R05	R04	Avg.
ABEJA-QwQ32b-Reasoning	**93.15**	86.24	**84.16**	**88.02**
ABEJA-Qwen2.5-32b	89.04	**88.07**	80.20	85.83
Qwen2.5-14B-Instruct	87.67	81.65	76.24	81.77
Qwen3-14B	87.67	81.65	74.26	81.22
shisa-v2-qwen2.5-32b	87.67	84.40	74.26	82.17

5.4.2 Results. The preliminary evaluation on validation sets (R04-R06) demonstrates that reasoning-enhanced architectures significantly outperform standard instruction-tuned models in legal entailment tasks, reported in Table 5. ABEJA-QwQ32b-Reasoning emerged as the most robust candidate, achieving the highest accuracy on R06 and R04, with a leading average performance. While ABEJA-Qwen2.5-32b performed best on the R05 dataset, the consistent strength of the reasoning-optimized model across multiple years suggests it is better equipped to handle the complex logical structures inherent in statute law. These findings directly informed our selection of the ABEJA-QwQ32b-Reasoning model as the primary driver for our final competition submissions.

Table 6: Task 4 Official Leaderboard: Comparison of JNLP Runs with Top Team Results (ranked by individual runs).

Rank	Team	Run	Acc.	Correct
1	DU	DU1	**96.34**	**79**
3	JNLP	JNLP-3	95.12	78
3	IAI	IAIrun2	95.12	78
8	JNLP	JNLP-1	92.68	76
8	RUG	RUG-1	92.68	76
8	UA	UA2	92.68	76
14	cody	cody5	90.24	74
17	HUKB	HUKBac	87.80	72
17	NOWJ	NOWJ-1	87.80	72
22	AIIR	CrossCoT	78.05	64
25	Bengo	REF	73.17	60
29	ASHIN	LLAMA3	60.98	50
31	JNLP	JNLP-2	57.32	47

Table 6 presents the official Task 4 leaderboard results, highlighting the performance of JNLP's three submitted runs. Our best-performing submission, JNLP-3, which employed dynamic few-shot prompting with BM25-based example retrieval and label balancing, achieved 95.12% accuracy (78 correct answers), securing 3rd place and trailing the winning DUDU team by only 1.22 percentage points. JNLP-1, utilizing zero-shot self-consistency through five independent inference iterations with majority voting, ranked 8th with 92.68% accuracy (76 correct). These results demonstrate that contextually relevant few-shot examples retrieved via BM25 significantly enhance legal reasoning performance, while simple multi-model ensembling without careful calibration introduces noise rather than robustness. The strong showing of JNLP-3 validates our hypothesis that providing the model with similar training examples and balanced label distributions is more effective than relying solely on zero-shot inference or model diversity.

5.5 Pilot Task: Tort Prediction

5.5.1 Setting. Our approach for Tort Prediction (TP) combines zero-shot prompting with *DeepSeek V3.2-exp* (JAIST-LJPJT25-TP run) and *Qwen3-14B*, together with a fine-tuned *Qwen2.5-32B-Instruct* model. The fine-tuned model is trained on the training set to better capture the nuances of Japanese legal texts.

Table 7: Official Pilot Task Results at COLIEE 2026 of Tort Prediction (TP) (ranked by individual runs).

Rank	Team	Run	Accuracy
1	JNLP	JAIST-LJPJT25-TP	**72.7%**
2	UIT	TortNLP4	72.6%
3	JNLP	Qwen25	68.0%
4	NOWJ	NOWJ1	67.6%
7	KIS	KIS 1	66.6%
12	JNLP	Qwen3	62.3%

5.5.2 Results. Table 7 reports the official results for the Tort Prediction (TP) task at COLIEE 2026. The JNLP team achieves the top performance with JAIST-LJPJT25-TP, reaching 72.7% accuracy and narrowly outperforming the second-ranked system by 0.1 percentage points. This result highlights the effectiveness of the proposed configuration in capturing complex legal reasoning.

In comparison, other JNLP runs show lower performance. The Qwen25 model achieves 68.0%, trailing the top system by 4.7 points, while Qwen3 further drops to 62.3%, indicating substantial sensitivity to model choice and configuration. These gaps suggest that prompt design and task-specific optimization play a critical role beyond the base model itself.

Overall, the results demonstrate that carefully engineered prompting strategies are essential for achieving state-of-the-art performance in tort prediction, and that model selection alone is insufficient without proper system design.

5.6 Pilot Task: Rationale Extraction

5.6.1 Setting. Our approach for Rationale Extraction (TP) combines zero-shot prompting with *DeepSeek V3.2-exp* (JAIST-LJPJT25-TP run), together with fine-tuned *Qwen2.5-32B-Instruct* and *Qwen3-14B* models. These fine-tuned models are trained on the training set to better capture the nuances of Japanese legal texts.

Table 8: Official Pilot Task Results at COLIEE 2026 of Rationale Extraction (RE) (ranked by individual runs).

Rank	Team	Run	Accuracy
1	UIT	TortNLP4	**68.2%**
2	JNLP	Qwen3	67.4%
3	JNLP	Qwen25	65.6%
4	NOWJ	NOWJ2	64.8%
5	JNLP	JAIST-LJPJT25-TP	64.5%
8	KIS	KIS2	62.6%

5.6.2 Results. Table 8 presents the official results for the Rationale Extraction (RE) task at COLIEE 2026. The JNLP team shows consistent performance across runs, with smaller variation compared to the Tort Prediction task.

Among JNLP submissions, Qwen3 achieves the best result at 67.4%, ranking second overall and only 0.8 percentage points behind the top system. This indicates strong competitiveness and suggests that this configuration is well-suited for rationale extraction. Qwen25 follows with 65.6%, showing a modest decline but remaining competitive.

In contrast, JAIST-LJPJT25-TP, which ranked first in Tort Prediction, attains 64.5% and falls to mid-tier performance. This shift highlights that configurations optimized for prediction do not directly transfer to rationale extraction, reflecting the task-specific nature of reasoning and prompt design.

Overall, JNLP results fall within a narrow range (64.5%-67.4%), indicating stable but less differentiated performance across configurations. This suggests that, unlike Tort Prediction, gains in rationale extraction rely more on inherent model capabilities than on prompt engineering alone, pointing to the need for more specialized approaches for evidence extraction.

6 Conclusion

In this paper, we presented our solutions for all main and pilot tasks in the COLIEE 2026 competition. By integrating classical machine learning with LLMs and advanced prompt engineering, our framework successfully delivers both efficient document retrieval and complex legal entailment. For retrieval tasks (Tasks 1, 2, and 3), we demonstrated that adopting a "less is more" strategy to strictly limit search results is vital for removing semantic noise and establishing a foundation for accurate LLM reasoning. For complex entailment (Tasks 3 and 4), standard zero-shot models prove insufficient; instead, strategies such as Dynamic Few-Shot Prompting and Multi-Agent Debate demonstrate superior efficacy in enhancing accuracy and mitigating hallucinations. Furthermore, our results emphasize that prompt design must match the cognitive demand of the objective: predicting legal outcomes (Task 5) requires rigid, judge-like structures, while rationale extraction (Task 6) relies primarily on reading comprehension. In the end, our results suggest that the future of legal support systems lies in fine-grained hybrid pipelines that balance computational scalability with the high precision required in legal practice.

Acknowledgements

This work was partly supported by Japan Science and Technology Agency (JST) as part of Adopting Sustainable Partnerships for Innovative Research Ecosystem (ASPIRE), Grant Number JPMJAP25B2.

References

[1] AI@Meta. Llama 3 model card. 2024.
[2] Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, et al. Ms marco: A human generated machine reading comprehension dataset. *arXiv preprint arXiv:1611.09268*, 2016.
[3] Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, Mir Rosenberg, Xia Song, Alina Stoica, Saurabh Tiwary, and Tong Wang. Ms marco: A human generated machine reading comprehension dataset, 2018.
[4] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners, 2020.
[5] MQ Bui, C Nguyen, DT Do, NK Le, DH Nguyen, and TTT Nguyen. Using deep learning approaches for tackling legal' s challenges (coliee 2022). In *Sixteenth International Workshop on Juris-informatics (JURISIN)*, 2022.
[6] Quan Minh Bui, Dinh-Truong Do, Nguyen-Khang Le, Dieu-Hien Nguyen, Khac-Vu-Hiep Nguyen, Trang Pham Ngoc Anh, and Minh Nguyen Le. Jnlp coliee-2023:

Data argumentation and large language model for legal case retrieval and entailment. In *Workshop of the tenth competition on legal information extraction/entailment (COLIEE' 2023) in the 19th international conference on artificial intelligence and law (ICAIL)*, 2023.

[7] Ilias Chalkidis, Manos Fergadiotis, Prodromos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos. LEGAL-BERT: The muppets straight out of law school. In Trevor Cohn, Yulan He, and Yang Liu, editors, *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2898–2904, Online, November 2020. Association for Computational Linguistics.

[8] Ilias Chalkidis, Abhik Jana, Dirk Hartung, Michael Bommarito, Ion Androutsopoulos, Daniel Katz, and Nikolaos Aletras. LexGLUE: A benchmark dataset for legal language understanding in English. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4310–4330, Dublin, Ireland, May 2022. Association for Computational Linguistics.

[9] Jianlyu Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. M3-embedding: Multi-linguality, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Findings of the Association for Computational Linguistics: ACL 2024*, pages 2318–2335, Bangkok, Thailand, August 2024. Association for Computational Linguistics.

[10] Hyeong Kyu Choi, Xiaojin Zhu, and Sharon Li. Debate or vote: Which yields better decisions in multi-agent large language models? In *Advances in Neural Information Processing Systems*, 2025.

[11] Damian Curran and Mike Conway. Similarity ranking of case law using propositions as features. In *New Frontiers in Artificial Intelligence: JSAI International Symposium on Artificial Intelligence, JSAI-IsAI 2024, Hamamatsu, Japan, May 28-29, 2024, Proceedings*, page 156-166, Berlin, Heidelberg, 2024. Springer-Verlag.

[12] Randy Goebel, Yoshinobu Kano, Mi-Young Kim, Calum Kwan, Ken Satoh, Hiroaki Yamada, and Masaharu Yoshioka. An overview of the coliee 2025 competition: Legal case law and statute law information retrieval and entailment. In *Proceedings of the Twentieth International Conference on Artificial Intelligence and Law*, ICAIL '25, page 506-515, New York, NY, USA, 2026. Association for Computing Machinery.

[13] Hugging Face. Sentence transformers: all-mpnet-base-v2. https://huggingface.co/sentence-transformers/all-mpnet-base-v2, 2021. Accessed: 2026-04-04.

[14] Yoshinobu Kano, Randy Goebel, Mi-Young Kim, Calum Kwan, Ken Satoh, Hiroaki Yamada, and Masaharu Yoshioka. An overview of the coliee 2026 competition: Legal case law and statute law information retrieval and entailment. In *Proceedings of the 21st International Conference on Artificial Intelligence and Law (ICAIL 2026)*, 2026.

[15] Daniel Katz, Michael Bommarito, and Josh Blackman. A general approach for predicting the behavior of the supreme court of the united states. *PLOS ONE*, 12, 04 2017.

[16] BENJAMIN LAUDERDALE and TOM CLARK. The supreme court's many median justices. *American Political Science Review*, 106, 11 2012.

[17] Joon Ho Lee. Analyses of multiple evidence combination. *SIGIR Forum*, 31(SI):267-276, July 1997.

[18] Haitao Li, Weihang Su, Changyue Wang, Yueyue Wu, Qingyao Ai, and Yiqun Liu. Thuir@coliee 2023: Incorporating structural knowledge into pre-trained language models for legal case retrieval, 2023.

[19] Haitao Li, Changyue Wang, Weihang Su, Yueyue Wu, Qingyao Ai, and Yiqun Liu. Thuir@coliee 2023: More parameters and legal knowledge for legal case entailment, 2023.

[20] Jimmy Lin, Xueguang Ma, Sheng-Chieh Lin, Jheng-Hong Yang, Ronak Pradeep, and Rodrigo Nogueira. Pyserini: A Python toolkit for reproducible information retrieval research with sparse and dense representations. In *Proceedings of the 44th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2021)*, pages 2356–2362, 2021.

[21] M Lin, SC Huang, and HL Shao. Rethinking attention: an attempting on revaluing attention weight with disjunctive union of longest uncommon subsequence for legal queries answering. In *Sixteenth International Workshop on Juris-informatics (JURISIN)*, 2022.

[22] Xueguang Ma, Liang Wang, Nan Yang, Furu Wei, and Jimmy Lin. Fine-tuning llama for multi-stage text retrieval. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2421–2425, 2024.

[23] Microsoft. Phi-3 medium 4k instruct. https://huggingface.co/microsoft/Phi-3-medium-4k-instruct, 2024. 14B parameter instruction-tuned transformer model with 4K context length.

[24] Chau Nguyen, Phuong Nguyen, Thanh Tran, Dat Nguyen, An Trieu, Tin Pham, Anh Dang, and Le-Minh Nguyen. Captain at coliee 2023: Efficient methods for legal information retrieval and entailment tasks, 2024.

[25] Chau Nguyen, Thanh Tran, Khang Le, Hien Nguyen, Truong Do, Trang Pham, Son T. Luu, Trung Vo, and Le-Minh Nguyen. Pushing the boundaries of legal information processing with integration of large language models. In *New Frontiers in Artificial Intelligence: JSAI International Symposium on Artificial Intelligence, JSAI-IsAI 2024, Hamamatsu, Japan, May 28-29, 2024, Proceedings*, page 167-182, Berlin, Heidelberg, 2024. Springer-Verlag.

[26] Hoang-Trung Nguyen, Tan-Minh Nguyen, Xuan-Bach Le, Tuan-Kiet Le, Khanh-Huyen Nguyen, Ha-Thanh Nguyen, Thi-Hai-Yen Vuong, and Le-Minh Nguyen. Nowj@coliee 2025: A multi-stage framework integrating embedding models and large language models for legal retrieval and entailment, 2025.

[27] Phuong Nguyen, Cong Nguyen, Hiep Nguyen, Minh Nguyen, An Trieu, Dat Nguyen, and Le-Minh Nguyen. Captain at coliee 2024: large language model for legal text retrieval and entailment. In *JSAI International Symposium on Artificial Intelligence*, pages 125–139. Springer, 2024.

[28] Animesh Nighojkar, Kenneth Jiang, Logan Fields, Onur Bilgin, Stephen Steinle, Yernar Sadybekov, Zaid Marji, and John Licato. Amhr coliee 2024 entry: legal entailment and retrieval. In *JSAI International Symposium on Artificial Intelligence*, pages 200–211. Springer, 2024.

[29] Rodrigo Nogueira, Zhiying Jiang, and Jimmy Lin. Document ranking with a pretrained sequence-to-sequence model. *arXiv preprint arXiv:2003.06713*, 2020.

[30] Takaaki Onaga and Yoshinobu Kano. Kis: Coliee 2025 task 4 solver using japanese llm. *The Review of Socionetwork Strategies*, pages 1–19, 2026.

[31] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(1), January 2020.

[32] Mrinal Rawat, Ambuje Gupta, Rushil Goomer, Alessandro Di Bari, Neha Gupta, and Roberto Pieraccini. Pre-act: Multi-step planning and reasoning improves acting in llm agents, 2025.

[33] Matthew Schultz and Thorsten Joachims. Learning a distance metric from relative comparisons. *Advances in neural information processing systems*, 16, 2003.

[34] Hsuan-Lei Shao, Yi-Chia Chen, and Sieh-Chuen Huang. Bert-based ensemble model for statute law retrieval and legal information entailment. In *New Frontiers in Artificial Intelligence: JSAI-isAI 2020 Workshops, JURISIN, LENLS 2020 Workshops, Virtual Event, November 15–17, 2020, Revised Selected Papers 12*, pages 226–239. Springer, 2021.

[35] Gemma Team. Gemma. 2024.

[36] Qwen Team. Qwen2.5: A party of foundation models, September 2024.

[37] Qwen Team. Qwen3 technical report, 2025.

[38] Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. Text embeddings by weakly-supervised contrastive pre-training. *arXiv preprint arXiv:2212.03533*, 2022.

[39] Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. Improving text embeddings with large language models. *arXiv preprint arXiv:2401.00368*, 2023.

[40] Sabine Wehnert, Viju Sudhi, Shipra Dureja, Libin Kutty, Saijal Shahania, and Ernesto W De Luca. Legal norm retrieval with variations of the bert model combined with tf-idf vectorization. In *Proceedings of the eighteenth international conference on artificial intelligence and law*, pages 285–294, 2021.

[41] Hai Ye, Xin Jiang, Zhunchen Luo, and Wenhan Chao. Interpretable charge predictions for criminal cases: Learning to generate court views from fact descriptions. In Marilyn Walker, Heng Ji, and Amanda Stent, editors, *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1854–1864, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[42] Masaharu Yoshioka, Yasuhiro Aoki, and Youta Suzuki. Bert-based ensemble methods with data augmentation for legal textual entailment in coliee statute law task. In *Proceedings of the eighteenth international conference on artificial intelligence and law*, pages 278–284, 2021.

[43] Masaharu Yoshioka, Youta Suzuki, and Yasuhiro Aoki. Hukb at the coliee 2022 statute law task. In *New Frontiers in Artificial Intelligence: JSAI-isAI 2022 Workshop, JURISIN 2022, and JSAI 2022 International Session, Kyoto, Japan, June 12–17, 2022, Revised Selected Papers*, pages 109–124. Springer, 2023.

NOWJ@COLIEE 2026: Adaptive Pipelines for Legal Retrieval and Reasoning

Thuong-Hieu Ngo
VNU University of Engineering and Technology
Hanoi, Vietnam
25025063@vnu.edu.vn

Hoang-Trung Nguyen
VNU University of Engineering and Technology
Hanoi, Vietnam
25025010@vnu.edu.vn

Huu-Dong Nguyen
VNU University of Engineering and Technology
Hanoi, Vietnam
21020760@vnu.edu.vn

Xuan-Bach Le
VNU University of Engineering and Technology
Hanoi, Vietnam
22024506@vnu.edu.vn

Le-Dung Nguyen
VNU University of Engineering and Technology
Hanoi, Vietnam
24022633@vnu.edu.vn

Quang-Thanh Tran
VNU University of Engineering and Technology
Hanoi, Vietnam
24021625@vnu.edu.vn

Ha-Thanh Nguyen
Center for Juris-Informatics, ROIS-DS
Research and Development Center for Large Language Models, NII
Tokyo, Japan
nguyenhathanh@nii.ac.jp

Thi-Hai-Yen Vuong
VNU University of Engineering and Technology
Hanoi, Vietnam
yenvth@vnu.edu.vn

Abstract

This paper presents the methodologies and results of the NOWJ team's participation across all five tasks of the COLIEE 2026 competition. For Task 1 (Legal Case Retrieval), we propose a four-stage pipeline comprising candidate filtering, dense retrieval with complementary embedding models, cross-encoder reranking via fine-tuned generative rerankers and MLP-based pairwise classification, and adaptive per-query cutoff prediction. For Task 2 (Legal Case Entailment), we combine BM25 filtering, T5-based reranking, and LLM-based entailment verification with consensus ensemble. For Task 3 (Statute Law Retrieval and Entailment), we adopt a retrieval-augmented generation framework with dense retrieval, attention-based reranking, and few-shot prompted LLM reasoning. For Task 4 (Legal Textual Entailment), we introduce a dynamic routing pipeline that classifies query difficulty and dispatches cases to either a balanced few-shot solver or a structured zero-shot chain-of-thought solver. For the Pilot Task (Legal Judgment Prediction), we combine hierarchical transformers with CRF layers, argument relation mining, and probabilistic argumentation graph reasoning.

CCS Concepts

• **Applied Computing** → **Law**; • **Computing Methodologies** → *Natural Language Processing*.

COLIEE 2026, Singapore

Keywords

Legal Information Processing, Document Retrieval, Textual Entailment, Multi-stage, Embedding Models, LLMs

ACM Reference Format:
Thuong-Hieu Ngo, Hoang-Trung Nguyen, Huu-Dong Nguyen, Xuan-Bach Le, Le-Dung Nguyen, Quang-Thanh Tran, Ha-Thanh Nguyen, and Thi-Hai-Yen Vuong. 2026. NOWJ@COLIEE 2026: Adaptive Pipelines for Legal Retrieval and Reasoning. In *Proceedings of COLIEE 2026 workshop, June 12, 2026, Singapore.* ACM, New York, NY, USA, 10 pages.

1 Introduction

Legal text processing is a specialized field that requires expertise in both law and information science. The increasing adoption of artificial intelligence (AI) and large language models (LLMs) in judicial processes has driven growing interest in automating core legal tasks such as case retrieval, statutory reasoning, and judgment prediction. The Competition on Legal Information Extraction and Entailment (COLIEE) [8] is an annual shared task organized to advance research in this area, covering challenges ranging from document retrieval and textual entailment to judgment prediction. COLIEE 2026, the thirteenth edition of the competition, comprises five tasks that span case law, statute law, and tort law.

Task 1 (Legal Case Retrieval) requires retrieving noticed cases from a corpus of case law documents that support a given query case whose citations have been redacted. Task 2 (Legal Case Entailment) requires identifying which paragraph in a noticed case entails the decision fragment of the query case. Task 3 (Statute Law Retrieval and Entailment) requires systems to jointly return an entailment decision and the supporting articles from a collection of Japanese Civil Code articles for a given legal query. Task 4 (Legal Textual Entailment) requires determining whether a given set of gold relevant articles entails or contradicts a legal query. The Pilot

Task (LJPJT26) requires predicting whether a tort is affirmed and extracting the accepted arguments from both plaintiff and defendant claims.

This paper presents the NOWJ team's participation across all five tasks. For Task 1, we propose a four-stage pipeline of candidate filtering, dense retrieval, cross-encoder reranking with two complementary approaches, and adaptive per-query cutoff prediction. For Task 2, we employ BM25 filtering, T5-based reranking, and LLM-based entailment verification with consensus ensemble. For Task 3, we adopt a retrieval-augmented generation framework combining dense retrieval, attention-based reranking, and few-shot prompted LLM reasoning. For Task 4, we introduce a dynamic routing pipeline that classifies query difficulty and dispatches cases to specialized solvers. For the Pilot Task, we combine hierarchical transformers with CRF layers, argument relation mining, and probabilistic argumentation graph reasoning. Our system achieves competitive performance across all tasks, with particularly strong results on Task 1 and Task 3, demonstrating the effectiveness of multi-stage pipelines that integrate dense retrieval with task-specific reranking and reasoning strategies. The remainder of this paper is organized by task, with each section detailing the task formulation, our proposed methodology, experimental setup, and results.

2 Task 1: Legal Case Retrieval

2.1 Task Overview

The rapid growth of legal databases has made manual identification of relevant precedents increasingly impractical. COLIEE Task 1 addresses this by formulating legal case retrieval as follows: given a query case q with all citations redacted, the objective is to retrieve all noticed cases $\{d_1, d_2, \ldots, d_n\}$ from a corpus of case law documents, each containing the case background, facts, reasoning, and decision. The noticed cases are those originally cited in support of the decision in q. The task presents several challenges: case documents typically range from 4,000 to 60,000 words, exceeding the context window of many transformer-based models; documents lack standardized structure; and the corpus contains mixed-language content (English and French).

In COLIEE 2025, the top-performing team, JNLP [6], achieved an F1 of 0.3353 by extending the UMNLP pairwise ranking framework with BM25 and SAILER-based features in a feed-forward classifier, with BM25 pre-filtering achieving 76–85% recall on the top 100–200 candidates. The runner-up, UQLegalAI [9], employed CaseLink, a graph neural network modeling case-level connectivity through contrastive learning, attaining an F1 of 0.2962. A consistent trend across participants was the adoption of multi-stage pipelines combining sparse retrieval with neural re-ranking and LLM-based summarization for document compression.

2.2 Methodology

To address the key challenges of Legal Case Retrieval—namely the excessive document length and complex logical structure of case texts—we propose a four-stage framework. The pipeline first applies year-based filtering to exclude temporally invalid candidates, followed by dense retrieval for initial ranking and reranking for refined relevance scoring. The overall architecture is illustrated in Figure 1.

Stage 1: Preprocessing and Candidate Filtering. Since the Federal Court of Canada corpus contains bilingual documents, French-language content is first removed using an automatic language detector[1]. We then apply two filtering heuristics: (i) excluding candidates whose trial date is later than the query's, as valid noticed cases must temporally precede the query, and (ii) removing cases that also appear as queries, as these are rarely cited as noticed cases. These steps prevent temporally invalid citations and substantially narrow the candidate space.

Stage 2: Dense Retrieval. The dense retrieval stage identifies the most promising candidates from the full corpus, balancing computational efficiency with high recall. We adopt two complementary pre-trained embedding models: Octen-Embedding-8B and E5-Mistral-7B-Instruct. Each document is independently encoded into dense vectors by both models. Following an instruction-aware encoding paradigm, queries are augmented with task-specific instructions to convey the retrieval objective, while documents are encoded in a neutral format.

```
Query: Instruct: Given a legal document, retrieve
relevant legal documents\nQuery: {query text}
Document: passage: {document text}
```

Since case documents frequently exceed the embedding model's context window, both queries and documents are split into chunks and independently encoded. To compute document-level relevance, we adopt a MaxSim scoring function inspired by ColBERT [4]:

$$s(q, d) = \sum_{i=1}^{M} \max_{j \in [N]} \mathbf{q}_i \cdot \mathbf{d}_j \tag{1}$$

where $\{\mathbf{q}_i\}_{i=1}^{M}$ and $\{\mathbf{d}_j\}_{j=1}^{N}$ denote the query and document chunk embeddings, respectively. This formulation allows each query chunk to match its most relevant document segment, effectively capturing partial relevance across long documents. All embeddings are ℓ_2-normalized, making the dot product equivalent to cosine similarity. The top-k results are forwarded to the reranking stage.

Stage 3: Reranking. Following the candidate retrieval stage, we introduce a reranking module to refine the ranking of candidate cases. This stage enhances retrieval performance by incorporating more expressive relevance modeling beyond the initial similarity scores. We propose two independent reranking approaches, each capturing different aspects of semantic and structural relevance between legal cases.

Approach 1: Weighted Ensemble. Dense retrieval encodes queries and candidates independently, limiting fine-grained semantic matching. The reranking stage addresses this by jointly encoding each query-candidate pair, enabling token-level interaction for more accurate relevance estimation.

We employ Qwen3-Reranker [12], a generative reranker that produces relevance scores from the logits at the final token position:

$$s(q, d) = \frac{\exp(l_{\text{yes}})}{\exp(l_{\text{yes}}) + \exp(l_{\text{no}})} \tag{2}$$

[1] https://huggingface.co/papluca/xlm-roberta-base-language-detection

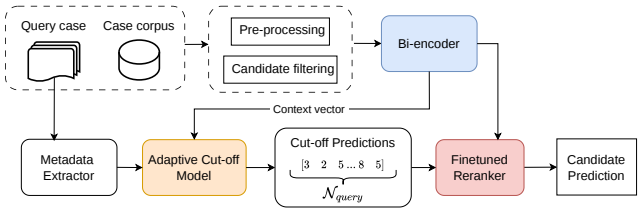


Figure 1: Four-stage pipeline including preprocessing and candidate filtering, dense retrieval, reranking, and adaptive per-query cutoff for legal case retrieval.

where l_{yes} and l_{no} are the logits of the "yes" and "no" tokens, casting relevance assessment as a binary judgment over the generated token distribution.

Both Qwen3-Reranker-4B and 8B variants are fine-tuned on COLIEE 2026 training data using ms-swift [13] with pointwise binary cross-entropy loss:

$$\mathcal{L} = -\left[y \log s(q,d) + (1-y)\log(1-s(q,d))\right] \tag{3}$$

where $y \in \{0,1\}$ is the ground-truth relevance label. Positive samples are drawn from gold noticed cases. To construct informative negatives, we apply a denoising strategy that filters out candidates whose retrieval scores fall too close to those of positive samples, ensuring that the training signal is not diluted by ambiguous near-positive examples. All training and inference follow the Qwen3-Reranker prompt template.

```
System: Judge whether the Document meets the
requirements based on the Query and the Instruct
provided. Note that the answer can only be "yes" or
"no".
User: <Instruct>: Given a legal document, retrieve
relevant legal documents \n <Query>: {query text} \n
<Document>: {candidate document text}
Assistant: {yes / no}
```

The final relevance score for each candidate integrates signals from all preceding stages—the dense retrieval score and both cross-encoder reranking scores—via a weighted combination:

$$s_{\text{fused}}(q,d) = \alpha \cdot \text{sim}(q,d) + \beta \cdot s_{4\text{B}}(q,d) + \gamma \cdot s_{8\text{B}}(q,d) \tag{4}$$

where $\text{sim}(q,d)$ is the cosine similarity from Stage 2, $s_{4\text{B}}$ and $s_{8\text{B}}$ are the reranking scores from Qwen3-Reranker-4B and 8B respectively, and α, β, γ are hyperparameters tuned on the development set ($\alpha = 0.5$, $\beta = 1.0$, $\gamma = 1.5$). This multi-signal fusion allows the system to leverage the broad recall of dense retrieval alongside the fine-grained relevance judgments of the cross-encoders.

Approach 2: MLP-based Reranking. To refine the initial retrieval results, we adopt a pairwise classification framework inspired by UMNLP [1]. Given a query–candidate pair, the model estimates a relevance probability using a multi-layer perceptron (MLP).

Candidate Construction. For each query, we build a candidate pool from the top-200 candidates retrieved in Stage 2 to reduce noise while maintaining high recall. Ground-truth relevant cases are treated as positive instances, while the remaining candidates serve as hard negatives. These top-ranked non-relevant cases are semantically closer to the query than random negatives, encouraging the model to learn finer-grained distinctions in legal reasoning.

Feature Construction. Each query–candidate pair is represented using proposition-level features following UMNLP. We further incorporate:

- **Semantic features**: Maximum cosine similarity from dense embeddings.
- **Lexical features**: BM25 scores and normalized variants.
- **Rank features**: Rank positions and reciprocal ranks from retrieval models.
- **Metadata features**: Legal domain categories and case types.

These features allow the model to jointly capture semantic similarity, lexical overlap, and structural signals from the retrieval stage.

Model Architecture and Training. The reranker is implemented as a feed-forward neural network that takes the concatenated feature vector and outputs a relevance probability $p(q,d) \in [0,1]$. The model is trained using binary cross-entropy loss:

$$\mathcal{L} = -\left[y \log p(q,d) + (1-y)\log(1-p(q,d))\right], \tag{5}$$

We train directly on the original data distribution without aggressive resampling, allowing the model to better reflect real-world retrieval conditions.

Stage 4: Adaptive Per-Query Cutoff. After reranking, the system must determine how many candidates to return for each query. This is non-trivial because the number of gold noticed cases varies significantly across queries—some have a single relevant precedent while others have several. Applying a fixed threshold would inevitably sacrifice either precision or recall depending on the query. To address this, we train an XGBoost and MLP ensemble to predict the expected number of relevant cases $|\mathcal{D}_q|$ for each query. The predictor takes as input document metadata features (e.g., document length, year) and dense embeddings from Octen-Embedding-0.6B. The predicted count is used to adaptively truncate the ranked list, enabling the system to dynamically adjust its output size per query and better balance precision and recall.

2.3 Experimental Setup

Building upon the proposed methodology, our experimental setup instantiates the four-stage pipeline with specific pre-trained models and training configurations. The first two stages—preprocessing with candidate filtering and dense retrieval—are shared across all runs. For the dense retrieval stage, we employ two complementary embedding models: Octen-Embedding-8B[2] and E5-Mistral-7B-Instruct[3], both used without task-specific fine-tuning. The three runs differ in the reranking approach applied from Stage 3 onward.

We submitted three official runs for Task 1:

- **Run 1 (submission_1):** Reranking with fine-tuned Qwen3-Reranker-4B[4] and Qwen3-Reranker-8B[5] via weighted score fusion, followed by adaptive per-query cutoff (XGBoost-/MLP). Both models are fine-tuned using ms-swift [13] with

[2]https://huggingface.co/Octen/Octen-Embedding-8B
[3]https://huggingface.co/intfloat/e5-mistral-7b-instruct
[4]https://huggingface.co/Qwen/Qwen3-Reranker-4B
[5]https://huggingface.co/Qwen/Qwen3-Reranker-8B

LoRA [3] (rank 16, α=32, dropout 0.05) on all linear layers, lr 1×10^{-6}, batch size 1, max length 16,660/8,400 tokens (4B/8B), 3 epochs with early stopping.

- **Run 2 (submission_2):** Stage-2 candidates are reranked using the MLP-based approach (Approach 2), which combines proposition-level, semantic, lexical, rank, and metadata features. Propositions are extracted with flan-t5-xl[6] and semantic embeddings are obtained via multilingual-e5-large-instruct[7], with the MLP predicting relevance scores for final reranking.
- **Run 3 (submission_3):** Combines predictions from Run 1 and Run 2 via score-level fusion. The scores from both runs are first normalized and then summed to produce final rankings, giving higher priority to cases consistently ranked highly by both methods.

2.4 Results and Discussion

Table 1 presents the recall performance of two dense retrieval models, Octen and E5, across multiple cutoff levels ranging from top-1 to top-200. Overall, both models demonstrate strong retrieval capability, with recall consistently improving as the cutoff increases.

Table 1: Recall at top-k cutoffs for Octen-Embedding-8B and E5-Mistral-7B-Instruct on the COLIEE 2026 Task 1 test set.

Top-k	Octen-Embedding-8B	E5-Mistral-7B-Instruct
R@1	0.1086	0.1103
R@3	0.2463	0.2440
R@5	0.3269	0.3309
R@10	0.4440	0.4474
R@20	0.5703	0.5680
R@50	0.7063	0.7079
R@100	0.7663	0.7863
R@200	0.8274	0.8463

Table 2 shows that our proposed four-stage pipeline achieves competitive performance, with `submission_2`, `submission_3`, and `submission_1` ranked 1st, 2nd, and 6th overall, respectively. The reranking stage improves precision by better distinguishing relevant cases from hard negatives. Among reranking strategies, the MLP feature-based reranker outperformed fine-tuned generative rerankers, as it leverages hand-crafted domain-specific features — such as cited article overlap, charge alignment, and structural case similarities — that are particularly discriminative in legal retrieval. In contrast, fine-tuned generative rerankers, while powerful in general settings, are optimized as domain-agnostic models intended to generalize across diverse data types and tasks, which may limit their ability to exploit the fine-grained, dataset-specific patterns present in legal corpora. Furthermore, the adaptive per-query cutoff enhances performance by dynamically balancing precision and recall. These findings suggest that combining domain-tailored reranking with an adaptive cutoff is a robust approach for legal case retrieval.

[6]https://huggingface.co/google/flan-t5-xl
[7]https://huggingface.co/intfloat/multilingual-e5-large-instruct

Table 2: Official results of the top runs on COLIEE 2026 Task 1.

Rank	Team	Run	F1	Precision	Recall
1	**NOWJ**	**submission_2**	**0.4220**	0.4235	0.4206
2	NOWJ	submission_3	0.4200	0.4140	**0.4263**
3	JNLP	random_forest	0.4126	**0.4341**	0.3931
4	JNLP	ensemble	0.4040	0.4174	0.3914
5	JNLP	xgboost	0.4000	0.4059	0.3943
6	NOWJ	submission_1	0.3880	0.3772	0.3994
7	SIL	submission_sil2	0.3871	0.4186	0.3600
8	SIL	submission_sil1	0.3810	0.3856	0.3766

3 Task 2: Legal Case Entailment

3.1 Task Overview

Given a decision d and a relevant case $R = \{p_1, p_2, ..., p_n\}$, Task 2 focuses on identifying the specific paragraph $p \in R$ that entails d. Unlike Task 1, which operates at the document level, this task demands fine-grained comprehension, as multiple paragraphs may discuss related legal concepts without directly supporting the final decision.

3.2 Methodology

To address this challenge, our system employs a multi-stage pipeline integrating sparse information retrieval, neural reranking, and LLM-based refinement. The pipeline consists of three core stages:

BM25 Initial Filtering: We utilize BM25, a sparse retrieval method based on term-frequency inverse-document-frequency (TF-IDF), to retrieve the top-k candidate paragraphs from R using d as the query.

Neural Reranking: The candidates are reranked using a T5-based cross-encoder. This model computes a relevance score by capturing deeper semantic interactions between d and p:

$$\text{score}(d, p) = \text{CrossEncoder}(d, p)$$

This step promotes contextually aligned paragraphs, filtering out lexical noise from the BM25 stage.

LLM Refinement and Ensemble Fusion: To make the final assessment, we deploy state-of-the-art LLMs to evaluate the highest-ranked outputs. Each LLM generates a binary prediction (entails/does not entail) based on specialized prompts. To ensure high precision, we apply a consensus-based ensemble fusion:

$$\text{final}(p) = \text{LLM}_1(p) \wedge \text{LLM}_2(p)$$

where $\text{final}(p) = 1$ indicates that the paragraph entails the decision. This mechanism mitigates individual model biases and hallucinations.

3.3 Experimental Setup

The proposed pipeline is implemented using specific pre-trained models. BM25 is used for initial filtering, followed by the MonoT5-3B model[8] for cross-encoder reranking. For the final LLM refinement, we experiment with Qwen3-235B-A22B[9] and QwQ-32B[10].

[8]castorini/monot5-3b-msmarco-10k
[9]https://huggingface.co/Qwen/Qwen3-235B-A22B
[10]https://huggingface.co/Qwen/QwQ-32B

To evaluate the effectiveness of individual LLMs and our proposed ensemble strategy, we submitted three official runs:

- **Run 1 (NOWJ001):** A single-LLM pipeline using Qwen3-235B-A22B for the final verification step.
- **Run 2 (NOWJ002):** A similar single-LLM setup, but utilizing QwQ-32B to evaluate its standalone performance.
- **Run 3 (NOWJ003):** The ensemble fusion configuration. A paragraph p is classified as entailment (final(p) = 1) if and only if both Qwen3-235B-A22B and QwQ-32B output positive predictions. Disagreements are discarded to prioritize precision.

3.4 Results and Discussion

Table 3: Leaderboard of the Case Textual Entailment task.

Team	F1	Precision	Recall
IAI	**0.4899**	0.4501	**0.5374**
AIIRLab	0.4706	0.5120	0.4354
JNLP	0.4507	0.4174	0.4898
NOWJ/nowj003	0.4429	0.7037	0.3231
UA	0.4254	0.4711	0.3878
NOWJ/nowj002	0.4029	0.7034	0.2823
JUNLLP	0.3942	0.6721	0.2789
NOWJ/nowj001	0.3744	**0.7604**	0.2483

As shown in Table 3, our ensemble approach (NOWJ/nowj003) achieved an F1 score of 0.4429, ranking fourth overall. A notable characteristic across all our submissions is the strong emphasis on precision. Run 1 (NOWJ/nowj001) recorded the highest precision among all participating teams at 0.7604, significantly outperforming the top-ranked IAI team (0.4501). Runs 2 and 3 also maintained precision above 0.70. This indicates that our two-stage reranking and LLM-based verification pipeline is highly effective at filtering out false positives. When the system predicts entailment, its confidence is consistently high.

However, this high precision comes at the expense of recall. Our recall scores ranged from 0.2483 to 0.3231, noticeably lower than those of the leading teams. This suggests that the LLMs applied overly strict criteria, resulting in a high number of false negatives. In addition, the initial BM25 retrieval stage may have filtered out lexically different but semantically relevant passages before they could be evaluated by the LLMs.

Comparing our internal runs further confirms the effectiveness of the ensemble approach. Using only the large Qwen3-235B model (Run 1) maximized precision but led to the lowest recall. The smaller QwQ-32B model (Run 2) provided a better balance, achieving a higher F1 score. Ultimately, the ensemble configuration (Run 3) proved to be the most effective setup. It stabilized the predictions, achieving the highest overall F1 and recall scores without significantly reducing precision.

For future improvements, the primary focus will be on increasing recall. This can be addressed by passing a larger candidate pool k to the cross-encoder or by relaxing the LLM prompts to capture broader entailment patterns while preserving our precision advantage.

4 Task 3: Statute Law Retrieval

4.1 Task Overview

Task 3 aims to retrieve a relevant subset of statutory articles for a legal query q from a fixed collection of Japanese civil law articles $A = \{a_1, a_2, \ldots, a_m\}$. In the end-to-end setting of COLIEE 2026, the system also returns a binary answer $\hat{y} \in \{Y, N\}$ supported by the predicted article set $\hat{A}_q \subseteq A$. The training data consist of triplets (q, A_q, y), where A_q is the gold set of relevant articles and y is the Yes/No label.

Most previous systems for statute retrieval in COLIEE follow a multi-stage pipeline. Previous systems typically begin with a retrieval step using lexical methods or compact encoder models, then apply reranking or LLM-based filtering to improve the quality of the selected articles [2]. Besides text matching, recent studies show that using the structural relations between legal articles can further improve performance. In [10], the authors show that modeling law articles as a reference network helps capture both local and long-range dependencies. This approach improves retrieval accuracy and reduces noise. These findings suggest that strong end-to-end performance depends not only on finding relevant articles, but also on filtering out irrelevant ones before making the final prediction.

4.2 Methodology

The system uses a three-stage retrieval-and-reasoning pipeline for Task 3. The goal is to automatically predict an entailment decision $\hat{y} \in \{Y, N\}$ together with a subset of supporting statutory articles $\hat{A}_q \subseteq A$ for a given query q.

Dense Retrieval. The first stage aims to identify a subset of relevant articles $\hat{A}_q$ from the full collection A. A pre-trained embedding model encodes both the input query and all legal articles into a shared vector space. Similarity between the query and each article is computed to rank the candidates. The top-ranked articles are selected as the retrieved set $\hat{A}_q$, which serves as the supporting evidence for the final decision.

Prompt Construction. To support the reasoning process, the system constructs a few-shot prompt using training data. For a given query q, similar queries in the training set are identified based on embedding similarity. Their corresponding triplets (q, A_q, y) are used as examples, where both the gold supporting articles A_q and the entailment label y are included. These examples are formatted into a prompt to guide the model in producing both a decision and the final article set.

Entailment Prediction. In the final stage, the model receives the query q, the retrieved article set $\hat{A}_q$, and the constructed prompt. Based on this input, it predicts the entailment label $\hat{y} \in \{Y, N\}$ and may refine the selected articles before producing the final output. The output thus aligns with the task requirement: generating both a binary decision and a subset of articles that support it.

4.3 Experimental Setup

Building on the proposed methodology, we use a unified retrieval-and-reasoning pipeline for the Statute Law Retrieval task. Given a query q and a statutory corpus A, the system aims to produce both an entailment decision $\hat{y} \in \{Y, N\}$ and a subset of supporting articles $\hat{A}_q \subseteq A$.

In the retrieval stage, `Qwen/Qwen3-Embedding-8B` is used to encode both queries and statutory articles into a shared vector space. Dense retrieval is applied using cosine similarity, and the top 20 articles are selected as the initial supporting set $\hat{A}_q$.

For Run 1, this set is further refined using QRHead [11] reranking. QRHead reranks candidates based on attention signals from the model, prioritizing articles more relevant to the query. Articles with higher attention scores are placed earlier, as they are more relevant to the query. This step refines the quality of the final supporting set $\hat{A}_q$.

In the reasoning stage, large language models take the query q and the retrieved article set $\hat{A}_q$ as input to produce the final decision $\hat{y}$. Prompt design is used to guide the reasoning process, and some runs include few-shot examples in the form of (q, A_q, y).

We submitted three official runs:

- **Run 1 (NOWJ_run1):** After the initial retrieval stage, QRHead is applied to re-rank the top 20 articles. The re-ranked article set $\hat{A}_q$ is then provided to `Qwen/Qwen3-235B-A22B` with a few-shot prompt for final answer prediction.
- **Run 2 (NOWJ_run2):** The top-20 retrieved articles are used without reranking. The reasoning stage uses a single model, `deepseek-ai/DeepSeek-R1-0528`, with a few-shot prompt (Prompt 1) to produce $\hat{y}$.
- **Run 3 (NOWJ_run3):** The same retrieval setup as Run 2 is used. In the reasoning stage, three different prompts (Prompt 1, Prompt 2, and Prompt 3) are applied using the same model. Each prompt produces a prediction, and the final decision $\hat{y}$ is chosen by majority voting.

4.4 Results and Discussion

As shown in Table 4, our system performs well across all three runs regarding entailment accuracy, with clear differences between the retrieval strategies. Run 1 (NOWJ_run1) achieves the highest accuracy of 0.9512 with 78 correct predictions, ranking first overall. This demonstrates that adding QRHead reranking after the initial dense retrieval helps the LLM better focus on the most relevant articles. Runs 2 and 3, which omit the reranking step, achieve a lower accuracy of 0.8902. Furthermore, the identical results between Run 2 (single prompt) and Run 3 (majority voting) indicate that ensemble prompting offers negligible benefits when the underlying retrieved context remains unimproved.

Table 4: Leaderboard of the Statute Law Retrieval task.

Run	Accuracy	Correct	F2	Recall
NOWJ/NOWJ_run1	**0.9512**	**78**	0.2224	**1.0000**
JNLP1	0.9024	74	**0.7796**	**1.0000**
NOWJ/NOWJ_run2	0.8902	73	0.2224	**1.0000**
NOWJ/NOWJ_run3	0.8902	73	0.2224	**1.0000**
codyrag	0.8537	70	0.1848	**1.0000**

While our system achieves perfect recall (1.0000) across all runs, the notably low F2 score (0.2224) highlights a clear limitation in the article selection stage. LLMs like Qwen and DeepSeek demonstrate strong deductive reasoning for the binary entailment task (achieving up to 0.9512 accuracy), yet they tend to adopt a conservative approach in evidence extraction. Rather than isolating the exact one or two necessary articles, the LLMs retain additional articles that provide background context but are not strictly legally required to formulate the $\hat{A}_q$ subset.

Compared to the JNLP1 baseline, which maintains perfect recall alongside a high F2 score (0.7796), it is evident that our current pipeline lacks a dedicated filtering mechanism. Relying solely on prompt-based extraction causes the LLM to prioritize the final Yes/No prediction over extraction precision. This suggests that future improvements should decouple the reasoning stage into two distinct phases: a strict statutory filtering phase to refine $\hat{A}_q$, followed by a separate entailment prediction phase.

5 Task 4: Legal Textual Entailment

5.1 Task Overview

Task 4 of the COLIEE focuses on Legal Textual Entailment. The primary objective is to construct a question-answering framework tailored for the legal domain. Formally, given a legal query q alongside a set of relevant statutory articles $A = \{a_1, a_2, \ldots, a_k\}$ (where $k \geq 1$), and a training dataset structured as triplets (q, A, L) with the target label $L \in \{Y, N\}$, the system must output a binary decision indicating whether A legally entails q. This task serves as a critical benchmark for investigating the capabilities of computational models in performing complex legal reasoning and precise statutory interpretation.

Reflecting on the previous iteration, the prevailing trend among top participants was the strategic adoption of Large Language Models (LLMs) to handle complex legal reasoning. The winning team, KIS [7], achieved top performance primarily by leveraging effective *few-shot prompting* strategies with a Japanese LLM. Meanwhile, the runner-up, CAPTAIN [5], enhanced their system by utilizing a *cause-effect* structural analysis tailored for civil law to generate structured examples, which were then integrated using data combination techniques. Furthermore, to prevent test-set data leakage, the competition strictly mandates that all submissions employ open-source models released prior to a specific cutoff date.

5.2 Methodology

In this task, we designed a dynamic, routing-based pipeline. Rather than applying a uniform approach, our method processes each legal query through a sequence of logical steps, adapting its reasoning strategy based on the inherent difficulty of the task.

Step 1: Difficulty Classifier. When a query and its relevant statutory articles enter our system, the first step is to evaluate their structural and logical complexity. We achieve this through a hybrid classifier. Initially, we apply rule-based heuristics: if the articles contain specific keywords indicating mutatis mutandis application, exceptions, or if the query requires synthesizing information from multiple articles, we immediately flag it as a "Hard" case. If these rules are not triggered, a lightweight LLM prompt is used as a secondary filter to scan for convoluted conditions or double negations, ultimately classifying the query as either "Easy" or "Hard". The prompt used for this step is as follows:

```
You are a legal task difficulty assessor.

Analyze the following legal clause and question, and
determine "Hard" if it is logically complex, or
"Easy" if it is a simple matching.

Hard: Negation of negation, subject mismatch, complex
conditions. If unsure, choose "Hard".
Easy: Simple question directly from the legal clause.
Answer "difficulty: Hard" if the query is Hard, and
"difficulty: Easy" if it is Easy.

Relevant articles: {articles_text}

Question: {query_text}
```

Step 2: Dynamic Routing and Solving. Based on the difficulty assessment, the query is routed to one of two specialized solvers:

- *For "Easy" Cases (Dynamic Balanced Few-Shot Solver):* Drawing inspiration from KIS[7], we optimize for pattern matching here. We embed the input using the *intfloat/multilingual-e5-large*[11] model and retrieve the most semantically similar examples from the training set. To prevent label bias, we enforce strict class balancing, retrieving exactly $k = 6$ "Yes" and $k = 6$ "No" examples. These are injected into the prompt, allowing the LLM to make a straightforward binary decision using in-context learning.
- *For "Hard" Cases (Zero-Shot Chain-of-Thought Solver):* Requiring rigorous legal deduction, we frame the LLM as a logical analyst required to articulate its steps within `<reasoning>` tags. The model first executes **Rule Extraction and Reference Resolution** to decompose statutes, followed by **Fact Subsumption** to map assertions against these rules. Additionally, strict guardrails prevent common fallacies (e.g., hallucinations, misuse of contrapositives, or misinterpreting provisos). Upon detecting any structural mismatch or logical leap, the model rejects the premise when structural mismatches are detected.

5.3 Experimental Setup

To evaluate the individual and combined contributions of our proposed modules, we structured our submission into three distinct runs. This setup allows us to compare specialized single-strategy approaches against our dynamic routing methodology:

- **Run 1 (Balanced Few-Shot):** A simplified approach bypassing the difficulty classifier entirely. All queries, regardless of their inherent complexity, are processed directly by the Dynamic Balanced Few-Shot Solver ($k = 6$). All inference in this run is powered by *DeepSeek-V3*[12]. This serves as our foundational baseline for in-context pattern matching.
- **Run 2 (Pure Zero-Shot CoT):** This run also bypasses the routing mechanism, instead applying the highly structured Zero-Shot Chain-of-Thought Solver to every query in the test set. Powered entirely by *DeepSeek-V3*[11], this configuration evaluates the model's raw logical deduction and rule extraction capabilities without the influence or aid of retrieved few-shot examples.
- **Run 3 (Hybrid Routing Pipeline):** This run deploys our complete proposed methodology. To optimize for both operational efficiency and reasoning depth, we employ a mixed-model architecture. The Difficulty Classifier utilizes *Llama-3.3-70b-versatile* for rapid categorization. Based on this assessment, queries are dynamically routed to either the Few-Shot Solver (for "Easy" cases) or the CoT Solver (for "Hard" cases). *DeepSeek-V3*[11] serves as the core reasoning engine for both solvers. This run tests the overall efficacy of dynamically adapting the prompting strategy based on query complexity.

```
You are a legal reasoning expert in
Japanese.
Answer strictly based on the
provided legal provisions. Do not
reason beyond the text.
Based on the relevant legal
provisions, answer whether the
statement is true or false.
Answer "Answer: True" if the
statement is true, and "Answer:
False" if it is false.
{few_shot_examples}
Relevant articles: {articles_text}
Query: {query_text}
Answer:
```

(a) Few-shot solver (Easy)

```
You are a logical analyst for the
Supreme Court and an expert in
rigorous legal entailment.
Based solely on the provided
"premise" determine whether the
"hypothesis" is logically correct or
contains contradictions.
1. Reference & Hierarchy Detection
2. Decomposition of Requirements and
Effects
3. Fact identification
4. Subsumption & Verification
5. Final Conclusion
<reasoning>
[Your reasoning here.]
</reasoning>
Answer: [Correct/Incorrect]
Relevant articles: {articles_text}
Query: {query_text}
Answer:
```

(b) Zero-shot CoT solver (Hard)

Figure 2: Prompt templates for Task 4 solvers: (a) balanced few-shot for "Easy" cases, (b) structured chain-of-thought for "Hard" cases.

5.4 Results and Discussion

The official evaluation results for COLIEE Task 4 are summarized in Table 5. In this highly competitive task, our team (**NOWJ**) achieved a Top 7 standing among all participating institutions.

Table 5 presents a streamlined version of the official leaderboard, displaying one representative run per rank alongside all three of our submitted configurations (NOWJ-1, NOWJ-2, and NOWJ-3). Our best-performing configuration, **NOWJ-1**, secured Rank 8 overall with an accuracy of 0.8780 (72 correct predictions).

To validate our methodology prior to the final submission, we conducted retrospective evaluations on historical COLIEE test sets from 2018, 2019, and 2020. The performance of our three configurations on these datasets is presented in Table 6.

As shown in Table 6, there is a clear performance hierarchy on the historical datasets: Run 3 consistently outperforms both Run 1 and Run 2. This strongly supported our initial hypothesis that a dynamic routing pipeline leveraging both few-shot pattern matching and rigorous CoT deduction is superior to any single-strategy approach. Furthermore, the Pure CoT approach (Run 2) generally underperformed the Few-Shot baseline (Run 1), indicating that strict logical deduction without examples is inherently challenging.

[11] https://huggingface.co/intfloat/multilingual-e5-large
[12] https://huggingface.co/deepseek-ai/DeepSeek-V3-0324

Table 5: Official Leaderboard for COLIEE Task 4 (Showing top-performing teams and our submissions).

Team	Run	Accuracy	Correct
DU	DU1	**0.9634**	**79**
JNLP	JNLP-3	0.9512	78
IAI	IAIrun1	0.9390	77
RUG	RUG-1	0.9268	76
UA	UA1	0.9146	75
cody	cody5	0.9024	74
NOWJ	NOWJ-1	0.8780	72
HUKB	HUKBac	0.8780	72
NOWJ	NOWJ-3	0.8049	66
AiR	AiIRCrossCoT	0.7805	64
Bengo	BengoREF	0.7317	60
ASHIN	ASHIN-LLAMA3	0.6098	50
NOWJ	NOWJ-2	0.5000	41

Table 6: Accuracy of the proposed configurations on historical COLIEE test sets.

Dataset	Run 1	Run 2	Run 3
COLIEE 2018	0.8060	0.7761	**0.9104**
COLIEE 2019	0.7909	0.7818	**0.8273**
COLIEE 2020	0.8889	0.9012	**0.9136**

While the historical data favored the Hybrid Pipeline (Run 3), the current year's blind test reveals a shift: the Few-Shot baseline (Run 1) is now the top performer (87.80%), significantly outperforming both the Hybrid model (80.49%) and the underperforming Pure CoT configuration (Run 2) (50.00%). The performance drop in Run 2 and 3 suggests two primary issues:

- **Rigidity of the CoT Guardrails:** The strict logical guardrails in the Zero-Shot CoT prompt, while preventing hallucinations on older data, likely failed to account for new semantic nuances, causing the model to default to "Incorrect/Neutral.".
- **Robustness of In-Context Learning over Strict Deduction:** Few-Shot in-context learning proved more resilient than isolated deduction. The retrieved examples provided a more reliable signal for "Hard" queries than the rigid step-by-step logic imposed by the CoT prompt.

Consequently, future work will shift from zero-shot prompting toward domain-specific fine-tuning. By training an LLM explicitly on complex legal cases, we aim to replace prompt-based guardrails with fine-tuned reasoning capabilities.

6 Pilot Task: Legal Judgment Prediction for Japanese Tort cases

6.1 Task Overview

This pilot task focuses on legal judgment prediction for Japanese civil cases concerning torts. In these scenarios, plaintiffs argue that a defendant's action constitutes a tort, while defendants contest the plaintiffs' arguments. The overarching objective is to process a set of case inputs to generate specific judicial outputs across two interconnected tasks: Tort Prediction and Rationale Extraction. Formally, the systems take (U, P, D) as input, where U represents undisputed facts that are agreed upon or not disputed by any parties, P denotes the arguments from the plaintiffs, and D encompasses the arguments from the defendants. Given these inputs, systems must output (T, R^P, R^D), defined by the following tasks:

- **Tort Prediction (TP)**: Predicts whether a tort is affirmed (T). Here, T is a Boolean value representing the final decision. Model performance for this task is evaluated using *accuracy*.
- **Rationale Extraction (RE)**: The final decision (T) is based on the arguments accepted by the judge, making these accepted arguments the rationales for the decision. RE identifies these accepted arguments by outputting R^P (for plaintiffs) and R^D (for defendants). Both are sequences of Boolean values denoting accepted arguments as True within P and D. This task is evaluated using the *micro-F1* measure.

6.2 Methodology

Table 7 presents key dataset statistics. The training and test sets show consistent distributions, with average claims per party stable around 3.5–3.9 claims. However, maximum complexity varies substantially: some cases contain over 100 plaintiff claims or more than 140 undisputed facts, requiring the system to handle both simple and highly complex cases. The balanced label distribution in training data, approximately 51% acceptance for both parties, indicates no inherent bias toward accepting or rejecting claims.

Table 7: Dataset characteristics for the Legal Judgment Prediction task across training and test sets.

Characteristic	Train	Test 2025	Test 2026
Cases	6,508	812	803
Avg. facts	1.33	1.37	1.30
Max. facts	134	26	141
Avg. plaintiff claims	3.88	3.83	3.77
Max. plaintiff claims	111	130	125
Avg. defendant claims	3.45	3.50	3.78
Max. defendant claims	86	50	99

Our system, illustrated in Figure 3, combines neural claim assessment with argumentation-based reasoning. A key feature of this framework is its explainability: the final tort decision follows from the set of accepted claims identified through structured reasoning over attack relations between arguments. Moreover, the system achieves strong performance without relying on large language models, making it computationally efficient and fast to deploy.

6.2.1 Hierarchical Transformer. The hierarchical transformer uses a two-stage architecture for claim assessment. The first stage employs a pre-trained Japanese language model as the word-level encoder, processing each claim and the fact sequence independently to generate contextualized embeddings. The second stage uses a span-level transformer that processes claim-level representations combining four information sources: the claim embedding from the word encoder, a positional embedding of the claim's sequential

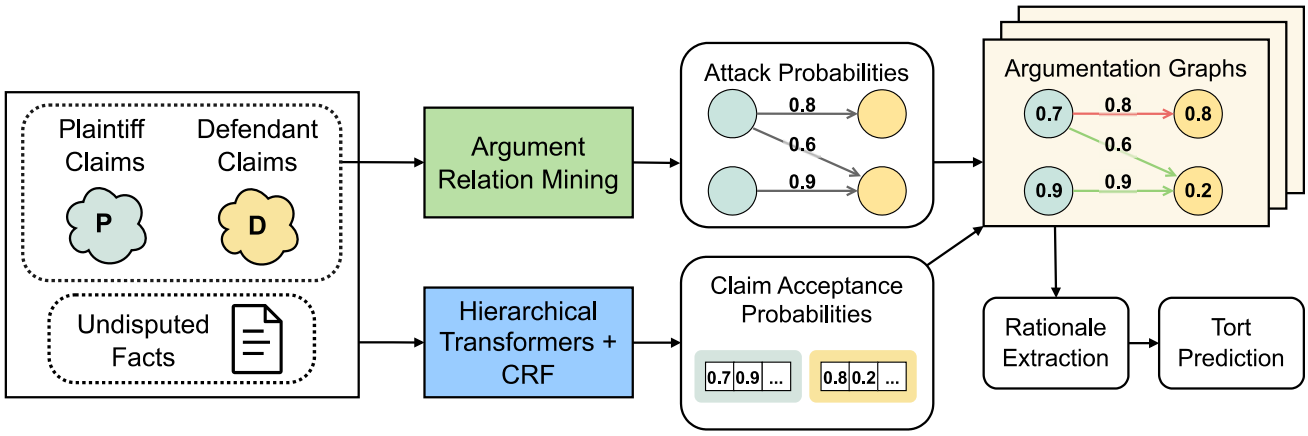


Figure 3: System architecture combining neural claim assessment and argumentation for legal judgment prediction.

position, a party-type embedding indicating plaintiff or defendant, and the representation of undisputed facts shared across all claims.

The model uses task-specific output layers. For Tort Prediction, a linear layer produces a binary court decision. For Rationale Extraction, we add a Conditional Random Field (CRF) layer above claim representations to model dependencies between claims from the same party, ensuring consistent label sequences. The training loss combines both tasks with a weighted sum to balance the two objectives. The model outputs binary predictions for the court decision via the TP head, binary label sequences for accepted claims via CRF decoding, and continuous probability scores for each claim's acceptance derived from *softmax* on the rationale logits. These probability scores serve as input to the argumentation reasoning module for conflict resolution.

6.2.2 Argument Relation Mining. To identify conflicts between opposing claims, we employ an argument relation mining model. We train the model on Japanese-translated argument mining datasets. The model analyzes all plaintiff-defendant claim pairs and performs binary classification to determine whether claims conflict with each other. We extract conflict relations above a confidence threshold as attacks, where one claim contradicts another. This produces a directed attack graph with confidence scores that represent the likelihood of each conflict relation.

6.2.3 Argumentation-Based Reasoning. This module resolves conflicts through structured reasoning over multiple graph configurations. The reasoning process operates in three stages. First, *graph ensemble generation* addresses uncertainty in attack detection. For N detected attacks, up to 2^N possible graphs exist depending on which attacks are valid. Rather than enumerating all possibilities, we use a priority queue to construct the top-K most probable graphs, where each graph $\mathcal{G}_\ell$ is assigned probability $P(\mathcal{G}_\ell)$ based on the likelihood of its included and excluded attacks. Second, *attack resolution* determines claim acceptance within each graph. An attack from claim a to claim b succeeds if a has equal or higher acceptance probability than b from the hierarchical transformer. Claims defeated by successful attacks are rejected, while unattacked claims or those surviving failed attacks retain their predicted status. Third, *probabilistic aggregation* combines results across the K graphs through weighted voting. For each claim c, we compute:

$$\rho(c) = \sum_{\ell=1}^{K} w_\ell \cdot v_\ell(c),$$

where $v_\ell(c) \in \{0, 1\}$ indicates whether c is accepted in graph $\mathcal{G}_\ell$, and w_ℓ are normalized graph probabilities satisfying $\sum_{\ell=1}^{K} w_\ell = 1$. The value $\rho(c)$ represents the total probability weight of graphs accepting c. A claim is accepted if $\rho(c) \geq 0.5$.

After determining accepted claims for both parties, the tort decision is made through a counting mechanism: the plaintiff prevails if they have more accepted claims than the defendant, reflecting the adversarial nature of tort litigation. This reasoning applies only to cases with detected conflicts; cases without attacks use direct predictions from the hierarchical transformer. The entire framework operates without LLMs, relying instead on a compact hierarchical transformer and lightweight reasoning components that enable rapid inference.

6.3 Experimental Setup

We implement the hierarchical transformer using *ModernBERT-Ja-310M*[13] as the word-level encoder. Sequence limits are set to 64 claims per case, 64 tokens per claim, and 512 tokens for concatenated facts. Missing components are handled by substituting empty sequences. The model is trained with a multi-task loss: $\mathcal{L}_{\text{total}} = 0.6 \cdot \mathcal{L}(\text{TP}) + 0.4 \cdot \mathcal{L}(\text{RE})$, where greater weight is assigned to tort prediction.

For argument relation mining, we finetune a pre-trained model[14] on Japanese-translated argument mining datasets, extracting conflict relations with a confidence threshold of 0.5. For argumentation reasoning, we use $K = 5$ graphs. Post-processing is applied to rationale predictions based on party-level consistency: if accepted claims outnumber rejected claims by a factor of $x = 1.5$, all claims from that party are set to accepted; if rejected claims outnumber

[13] https://huggingface.co/sbintuitions/modernbert-ja-310m
[14] https://huggingface.co/raruidol/ArgumentMining-EN-ARI-US2016

accepted claims by the same factor, all claims are set to rejected. Cases without a clear majority remain unchanged.

We submit three configurations to evaluate component contributions: *Run 1* uses argumentation reasoning for TP and hierarchical transformer predictions for RE; *Run 2* adds post-processing to RE; *Run 3* uses only the hierarchical transformer as an ablation baseline.

6.4 Results and Discussion

Table 8 shows official results on the COLIEE 2026 test set. Our system achieved 3rd place for both tasks. For TP, our best runs (Runs 1–2) achieved 67.6% accuracy, while for RE, our best run (Run 2) achieved 64.8% F1.

Table 8: Official results for the COLIEE 2026 Pilot Task.

Team	Accuracy
JNLP	**72.7%**
UIT	72.6%
NOWJ (run 1)	67.6%
NOWJ (run 2)	67.6%
NOWJ (run 3)	67.1%
KIS	66.6%

(a) Tort Prediction

Team	F1 score
UIT	**68.2%**
JNLP	67.4%
NOWJ (run 2)	64.8%
NOWJ (run 3)	63.4%
NOWJ (run 1)	63.4%
KIS	62.6%

(b) Rationale Extraction

Comparing the three runs reveals each component's contribution. For TP, Runs 1 and 2, which incorporate argumentation reasoning, achieve 67.6% accuracy, outperforming Run 3 (67.1%) by 0.5 percentage points. For RE, Run 2, which adds post-processing, achieves 64.8% F1, outperforming Runs 1 and 3 (both 63.4%) by 1.4 percentage points. The performance gap to the top team is 5.1 percentage points for TP and 3.4 percentage points for RE, presenting clear opportunities for future improvement. The argumentation reasoning component contributed a 0.5 percentage point gain in TP accuracy, suggesting that conflict modeling is a promising direction for further development.

A key strength of our framework lies in its explainability. Unlike black-box approaches, our system produces decisions that are traceable: the tort outcome follows directly from the set of accepted claims identified through explicit argumentation graphs. This transparency enables practitioners to inspect which claims were accepted or rejected and how conflicts between opposing arguments were resolved. Furthermore, the framework achieves competitive performance without relying on large language models, instead using a compact hierarchical transformer and lightweight reasoning components that enable rapid inference with modest computational requirements. Future work to enhance performance includes several directions. First, training argument relation models on native Japanese data to better capture language-specific argumentation patterns. Second, incorporating richer attack resolution strategies that consider claim semantics and supporting evidence, and integrating undisputed facts directly into the reasoning stage to ground conflict resolution in factual context. Third, exploring alternative argumentation semantics such as preferred or stable extensions, and integrating external legal knowledge to further improve prediction accuracy while maintaining the interpretability that our framework provides.

7 Conclusion

This paper described the NOWJ team's participation in all five tasks of COLIEE 2026, achieving 1st place on Task 1 (F1: 0.4220) and Task 3 (accuracy: 0.9512), 4th on Task 2 (F1: 0.4429), 8th on Task 4 (accuracy: 0.8780), and 3rd on the Pilot Task (Accuracy: 67.6%, F1: 64.8%). Key findings include: MLP-based reranking with handcrafted features outperformed generative cross-encoders (Task 1), LLM consensus verification yields high precision but limited recall (Task 2), attention-based reranking via QRHead substantially improves LLM reasoning accuracy (Task 3), difficulty-based routing is vulnerable to distribution shift (Task 4), and argumentation graph reasoning provides modest but explainable gains for tort prediction (Pilot Task). Future work will focus on improving retrieval recall through domain-specific fine-tuning, relaxing entailment verification thresholds, decoupling article filtering from entailment prediction, replacing zero-shot prompting with fine-tuned models, and training argument relation classifiers on Japanese data.

References

[1] Damian Curran and Mike Conway. 2024. Similarity Ranking of Case Law Using Propositions as Features. In *New Frontiers in Artificial Intelligence (Lecture Notes in Computer Science)*. Springer, 156–166. doi:10.1007/978-981-97-3076-6_11

[2] Randy Goebel, Yoshinobu Kano, Mi-Young Kim, Calum Kwan, Ken Satoh, Hiroaki Yamada, and Masaharu Yoshioka. 2025. An Overview of the COLIEE 2025 Competition: Legal Case Law and Statute Law Information Retrieval and Entailment. In *Proceedings of the Twentieth International Conference on Artificial Intelligence and Law*. 506–515.

[3] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. LoRA: Low-Rank Adaptation of Large Language Models. arXiv:2106.09685 [cs.CL] https://arxiv.org/abs/2106.09685

[4] Omar Khattab and Matei Zaharia. 2020. ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 39–48.

[5] Dat Nguyen, Minh-Phuong Nguyen, Quang-Huy Chu, Son T. Luu, Nguyen Hoang Chu, Trung Vo, and Le-Minh Nguyen. 2025. CAPTAIN at COLIEE 2025: Enhancing Legal Text Processing and Structural Analysis with Large Language Models. In *Proceedings of COLIEE 2025 Workshop*. ACM, Chicago, USA.

[6] Hai Nguyen, Hiep Nguyen, Trang Pham, Minh Nguyen, An Trieu, Dinh-Truong Do, Nguyen-Khang Le, and Le-Minh Nguyen. 2025. JNLP@COLIEE 2025: Hybrid Large Language Model-based Framework for Legal Information Retrieval and Entailment. In *Proceedings of the Workshop on the Twelfth International Competition on Legal Information Extraction and Entailment (COLIEE 2025)*. ACM, Chicago, USA, 57–66.

[7] Takaaki Onaga and Yoshinobu Kano. 2026. KIS: COLIEE 2025 Task 4 Solver Using Japanese LLM. *The Review of Socionetwork Strategies* (2026). doi:10.1007/s12626-026-00209-w

[8] Juliano Rabelo, Randy Goebel, Mi-Young Kim, Yoshinobu Kano, Masaharu Yoshioka, and Ken Satoh. 2024. Overview and Discussion of the Competition on Legal Information Extraction/Entailment (COLIEE) 2023. *The Review of Socionetwork Strategies* 18, 1 (2024), 27–47.

[9] Yanran Tang, Ruihong Qiu, and Zi Huang. 2025. UQLegalAI@COLIEE2025: Advancing Legal Case Retrieval with Large Language Models and Graph Neural Networks. arXiv:2505.20743 [cs.IR] https://arxiv.org/abs/2505.20743

[10] Thi-Hai-Yen Vuong, Hai-Long Nguyen, Tan-Minh Nguyen, Ha-Thanh Nguyen, Le-Minh Nguyen, and Xuan-Hieu Phan. 2025. Uncovering connections: a reference network approach to statute law retrieval. *Applied Intelligence* 55, 13 (2025), 936.

[11] Wuwei Zhang, Fangcong Yin, Howard Yen, Danqi Chen, and Xi Ye. 2025. Query-focused retrieval heads improve long-context reasoning and re-ranking. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*. 23802–23816.

[12] Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, Fei Huang, and Jingren Zhou. 2025. Qwen3 Embedding: Advancing Text Embedding and Reranking Through Foundation Models. arXiv:2506.05176 [cs.CL] https://arxiv.org/abs/2506.05176

[13] Yuze Zhao, Jintao Huang, Jinghan Hu, Xingjun Wang, Yunlin Mao, Daoze Zhang, Hong Zhang, Zeyinzi Jiang, Zhikai Wu, Baole Ai, Ang Wang, Wenmeng Zhou, and Yingda Chen. 2025. SWIFT:A Scalable lightWeight Infrastructure for Fine-Tuning. arXiv:2408.05517 [cs.CL] https://arxiv.org/abs/2408.05517

UIT-TortNLP: Multi-task Learning and Large Language Models Approaches for Japanese Tort Cases

Lam M. H. Nguyen
University of Information Technology, Ho Chi Minh City, Vietnam
Vietnam National University, Ho Chi Minh City, Vietnam
23520834@gm.uit.edu.vn

Thang Q. Pham*
University of Information Technology, Ho Chi Minh City, Vietnam
Vietnam National University, Ho Chi Minh City, Vietnam
23521431@gm.uit.edu.vn

Ngoc N. H. Ho*
University of Information Technology, Ho Chi Minh City, Vietnam
Vietnam National University, Ho Chi Minh City, Vietnam
23521021@gm.uit.edu.vn

Son T. Luu
University of Information Technology, Ho Chi Minh City, Vietnam
Vietnam National University, Ho Chi Minh City, Vietnam
sonlt@uit.edu.vn

Abstract

Legal Judgment Prediction has emerged as an active research topic in Legal Natural Language Processing (NLP), aiming to automatically infer judicial decisions from legal case documents. In this paper, we present our system for the COLIEE 2026 pilot task on Legal Judgment Prediction for Japanese Tort Cases, which consists of two subtasks: Tort Prediction and Rationale Extraction. We explore two approaches to address this problem. The first approach adopts a multi-task learning framework that jointly models the dependencies between tort prediction and rationale extraction. The second approach leverages instruction tuning of large language models (LLMs) with QLoRA, enabling parameter-efficient adaptation of pretrained models to the legal domain. Experimental results show that the multi-task learning framework performs competitively, while the instruction-tuned LLM approach achieves the best overall performance. In particular, our instruction-tuned system ranks Top-2 in Subtask 1 with 72.6% accuracy and first in Subtask 2 with an F1-score of 68.2%, highlighting the effectiveness of instruction tuning with parameter-efficient fine-tuning for legal judgment prediction.

CCS Concepts

• **Computing methodologies** → **Information extraction**; • **Applied computing** → *Law*.

Keywords

COLIEE2026, Legal Judgment Prediction, Multi-task Learning, Large Language Model, Instruction Tunning, QLoRa

*These authors contributed equally to this research.

COLIEE 2026, June 12, 2026, Singapore

ACM ISBN 978-x-xxxx-xxxx-x/YYYY/MM

ACM Reference Format:
Lam M. H. Nguyen, Thang Q. Pham, Ngoc N. H. Ho, and Son T. Luu. 2026. UIT-TortNLP: Multi-task Learning and Large Language Models Approaches for Japanese Tort Cases. In *Proceedings of COLIEE 2026 (COLIEE 2026, June 12, 2026).* ACM, New York, NY, USA, 7 pages.

1 Introduction

Recent advances in Artificial Intelligence, particularly in Natural Language Processing, have significantly expanded the ability of machines to process and analyze large volumes of textual data. These techniques have been increasingly applied to real-world applications, especially in the legal domain, where documents such as statutes, contracts, and court decisions are predominantly text-based. NLP provides promising tools to assist legal practitioners and researchers in tasks including legal information retrieval, case analysis, and decision support.

To further advance automation and research in legal NLP, the Competition on Legal Information Extraction and Entailment (COLIEE) [3] provides shared tasks for evaluating NLP methods in legal reasoning and analysis. In this competition, we focus on the pilot task Legal Judgment Prediction for Japanese Tort Cases (LJPJT) [17], which aims to support automated analysis of civil court decisions related to tort disputes. This task consists of two subtasks: Tort Prediction (TP), which predicts whether a tort claim is affirmed given the facts and arguments from both parties, and Rationale Extraction (RE), which identifies the accepted arguments that support the final judgment.

In this work, we address the Legal Judgment Prediction for Japanese Tort Cases task [6] by considering both final decisions and their supporting arguments. We explore two different approaches for this setting. The first is a multi-task learning framework designed to jointly model the interactions among case facts, party arguments, and judgment outcomes. The second is an LLM-based approach that applies instruction tuning with QLoRa [2], allowing efficient adaptation to the task under limited computational resources.

Our main contributions are summarized as follows:

- We propose a multi-task learning framework that jointly learns Tort Prediction and Rationale Extraction, allowing shared representations to capture useful signals across the two subtasks.

- We investigate a parameter-efficient LLM-based approach using instruction tuning with QLoRA, enabling large language models to be effectively adapted to the legal judgment prediction task while reducing training overhead.
- We provide an empirical comparison between multi-task learning and parameter-efficient LLM adaptation for the Legal Judgment Prediction for Japanese Tort Cases task.

2 Related Works

2.1 The LJPJT 2026 Pilot Task

The LJPJT 2026 Pilot Task addresses the problem of predicting judicial decisions in civil tort cases under the *Japanese Civil Code (Art. 709)*, where a tort refers to a negligent or intentional infringement of rights that causes harm to a plaintiff. Each case involves two parties: plaintiffs, who claim that the defendant's actions constitute a tort, and defendants, who contest these claims. The problem is formulated using three main components: undisputed facts (U), plaintiffs' arguments (P), and defendants' arguments (D). The goal of Tort Prediction (TP) is to determine whether the tort claim is affirmed (T), while Rationale Extraction (RE) aims to identify the arguments accepted by the judge as supporting reasons for the final decision. Thus, the task can be summarized as predicting the judgment outcome and extracting accepted arguments given the input (U, P, D).

Several approaches have been proposed to address the LJPJT Pilot Task. CAPTAIN [8] adopted a straightforward approach by fine-tuning a large language model to jointly solve both Tort Prediction (TP) and Rationale Extraction (RE), using a best-performing model according to the Open Japanese LLM Leaderboard [7] at that time. KIS [5] fine-tuned a Japanese variant of ModernBERT [13] capable of handling longer input sequences. In addition to single-model fine-tuning, they explored ensemble strategies by combining multiple models trained with different hyperparameters and through cross-validation. NOWJ [9] employed a hierarchical Transformer architecture, where a ModernBERT [14] model was used for word-level encoding and a Transformer for span-level reasoning. TP was formulated as a binary classification task, while RE was treated as sequence labeling with a CRF layer to capture relationships between claims. OVGU [15] proposed a two-stage pipeline in which RE was first solved using large language models through prompting, and TP was subsequently determined using a rule-based scoring function derived from correlations observed in the training data.

2.2 Multi-task Learning

Multi-task learning (MTL) has been widely studied in machine learning and natural language processing as an effective paradigm for improving model generalization by jointly learning multiple related tasks. The central idea of MTL is to share representations among tasks so that the model can capture common underlying patterns and transfer useful knowledge between them. Early work by [1] demonstrated that learning tasks in parallel with shared representations can significantly improve generalization performance compared to training each task independently.

In the legal domain, MTL has been extensively explored, where several subtasks such as law article prediction, charge prediction, and penalty prediction are inherently interdependent. For instance, the TopJudge [20] framework models the dependencies among these subtasks using a topological structure and demonstrates that jointly learning them improves prediction performance. Similarly, Multi-Perspective Bi-Feedback Network [18] leverages bidirectional information flow between subtasks to capture their mutual dependencies and improve legal judgment prediction. More recent studies further explore sophisticated multi-task architectures for legal reasoning. For example, a hierarchical MTL framework has been proposed to model logical dependencies between legal subtasks and simulate judicial reasoning processes, showing improved prediction accuracy on real-world legal datasets [17, 19]. These studies suggest that jointly modeling related legal tasks can enhance both predictive performance and interpretability by encouraging models to learn shared semantic representations across different components of legal cases.

2.3 Instruction Tuning of Large Language Models with QLoRA

Large Language Models have demonstrated strong performance across a wide range of natural language processing tasks due to their ability to learn general language representations from large-scale pretraining corpora. To adapt these models to downstream tasks, instruction tuning has emerged as an effective paradigm in which models are trained on instruction–response pairs, enabling them to better follow task instructions and generalize to new tasks [16].

However, fine-tuning large models with billions of parameters often requires substantial computational resources. To address this challenge, Low-Rank Adaptation (LoRA) [4] was proposed as a parameter-efficient fine-tuning method that injects trainable low-rank matrices into transformer layers while keeping the original model parameters frozen. Building upon this idea, QLoRA [2] further improves efficiency by combining LoRA with 4-bit quantization, enabling large models to be fine-tuned with significantly reduced memory consumption while maintaining performance comparable to full fine-tuning. This approach has been widely adopted for adapting LLMs to domain-specific tasks, including applications in legal NLP.

3 Proposed Method

3.1 Problem Formulation

A case instance consists of three components:

- *Undisputed facts* U, describing the factual background agreed upon by both parties;
- *Plaintiff claims* $\mathcal{P} = \{p_1, \ldots, p_M\}$;
- *Defendant claims* $\mathcal{D} = \{d_1, \ldots, d_N\}$.

The system solves two subtasks simultaneously. For **RE**, it predicts a binary acceptance probability $\hat{r}_i^P \in [0, 1]$ for each plaintiff claim p_i and $\hat{r}_j^D \in [0, 1]$ for each defendant claim d_j, indicating whether the court accepts that claim. For **TP**, it predicts a verdict score $\hat{T} \in [0, 1]$. Throughout this section, d denotes the hidden dimensionality of the pre-trained encoder.

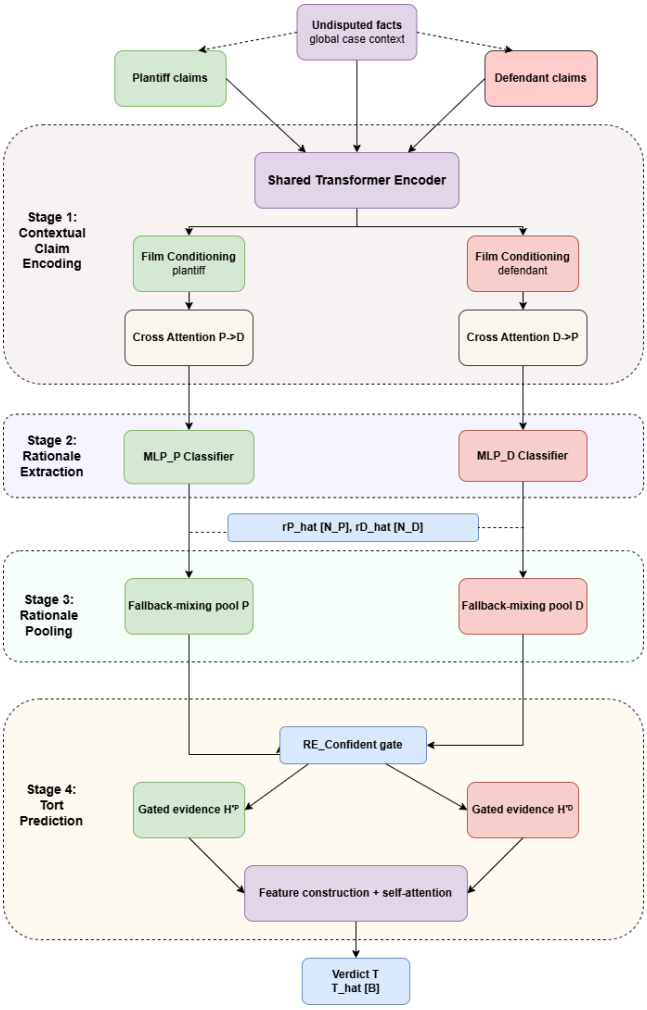


Figure 1: Overview of the proposed multi-task learning architecture.

3.2 Multi-task Learning Architecture

We propose a four-stage neural architecture for jointly solving *Rationale Extraction* (RE) and *Tort Prediction* (TP). The model is designed as a hierarchical pipeline: it first constructs context-aware representations for individual claims, then predicts claim-level rationales, aggregates them into case-level evidence, and finally performs verdict prediction. To stabilise training, we employ a teacher forcing mechanism that gradually transitions from ground-truth supervision to model predictions. An overview of the architecture is provided in Figure 1

3.2.1 Stage 1: Contextual Claim Encoding. The first stage transforms raw claim texts into context-enriched vector representations through three sequential operations.

Shared Text Encoder and FiLM Conditioning. All texts are encoded by a single shared **ModernBERT-ja-310M** encoder [14], with CLS pooling applied to obtain fixed-size representations. Specifically, the undisputed facts U are encoded once to produce a global case vector:

$$\mathbf{H}_u = \mathrm{Enc}(U), \tag{1}$$

and each claim is encoded independently using the same encoder weights:

$$\mathbf{h}_i^P = \mathrm{Enc}(p_i), \quad \mathbf{h}_j^D = \mathrm{Enc}(d_j). \tag{2}$$

Sharing encoder parameters across all texts ensures that claims and facts reside in a common semantic space, and encoding U only once reduces computational cost for cases with large claim sets.

The global case vector $\mathbf{H}_u$ is then used to condition each claim representation via *Feature-wise Linear Modulation* (FiLM) [11]. Rather than simply concatenating $\mathbf{H}_u$ with each claim embedding — which treats all feature dimensions uniformly — FiLM learns per-dimension scale and shift parameters from $\mathbf{H}_u$ and applies them element-wise:

$$\tilde{\mathbf{h}}_i^P = \underbrace{\boldsymbol{\gamma}(\mathbf{H}_u)}_{\text{scale}} \odot \mathbf{h}_i^P + \underbrace{\boldsymbol{\beta}(\mathbf{H}_u)}_{\text{shift}}, \tag{3}$$

where $\boldsymbol{\gamma}, \boldsymbol{\beta} : \mathbb{R}^d \to \mathbb{R}^d$ are learned linear projections. The same transformation is applied symmetrically to defendant claims. This multiplicative conditioning allows the model to selectively amplify or suppress specific semantic dimensions of each claim based on the case context — analogous to how the same argument may carry different legal weight depending on the specific factual background. In practice, $\mathbf{H}_u$ is mapped to each claim using an index mapping, enabling efficient batch processing without recomputing the fact representation for every claim.

Inter-Party Cross-Attention. After FiLM conditioning, we model adversarial interactions between opposing claims through bidirectional cross-attention. The strength of a plaintiff claim depends not only on its intrinsic merit but also on the specific defendant claims it must overcome, and vice versa.

Given a set of query claim representations $\mathbf{h}_q$ and key-value representations from the opposing party $\mathbf{h}_{kv}$, scaled dot-product attention is computed as:

$$\mathrm{Attn}(\mathbf{h}_q, \mathbf{h}_{kv}) = \mathrm{softmax}\left(\frac{QK^\top}{\sqrt{d}}\right)V, \tag{4}$$

where $Q = \mathbf{h}_q W_Q$, $K = \mathbf{h}_{kv} W_K$, and $V = \mathbf{h}_{kv} W_V$ are learned linear projections. The output is integrated with the original claim representation via a residual connection followed by layer normalisation:

$$\mathbf{h}' = \mathrm{LayerNorm}(\mathbf{h} + \mathrm{Attn}(\mathbf{h}, \mathbf{h}_{\text{other}})). \tag{5}$$

Two independent cross-attention modules capture this bidirectional interaction: the P→D module allows each plaintiff claim to attend over the full set of defendant claims in the same case, and the D→P module does the converse. Since the number of claims per case varies, attention is computed per-case individually using a claim-to-case index map, avoiding padding artefacts.

3.2.2 Stage 2: Rationale Extraction. Given the context-enriched claim representations from Stage 1, two independent binary classifiers predict per-claim rationale scores. Each classifier is a multilayer perceptron (MLP) with GELU activations and dropout, applied independently to each claim:

$$\hat{r}_i^P = \sigma\left(\mathrm{MLP}_P(\mathbf{h}_i^{P'})\right), \quad \hat{r}_j^D = \sigma\left(\mathrm{MLP}_D(\mathbf{h}_j^{D'})\right), \tag{6}$$

where $\mathbf{h}_i^{P'}$ and $\mathbf{h}_j^{D'}$ are the Stage 1 outputs, and σ denotes the sigmoid function. Logits are clamped before sigmoid for numerical stability. Here, $r_i^P \in \{0, 1\}$ denotes the ground-truth rationale label, while $\hat{r}_i^P$ denotes the corresponding predicted probability.

The two classifiers do *not* share parameters. This design choice reflects the structural asymmetry of legal argumentation: plaintiff claims must affirmatively establish facts such as causation and damages, while defendant claims primarily seek to rebut or negate, resulting in systematically different linguistic and semantic patterns between accepted and rejected claims on each side.

3.2.3 Stage 3: Rationale Pooling. Stage 3 aggregates the variable-length per-claim representations and rationale scores into fixed-size case-level evidence vectors for consumption by the TP head.

Fallback-Mixing Pooling. A straightforward approach would pool claim representations by weighting each claim proportionally to its predicted rationale score. However, this is unreliable in early training when the RE module has not yet converged and all predicted scores are near 0.5 — in this regime, the weighted pool carries no more information than a uniform mean pool, while gradients become unstable.

We therefore adopt a *fallback-mixing* strategy that blends rationale-weighted pooling with uniform mean pooling in equal proportions. The softmax-normalized rationale scores provide soft pooling weights:

$$\alpha_i = \frac{\exp(\hat{r}_i^P)}{\sum_k \exp(\hat{r}_k^P)}. \tag{7}$$

The final pooled representation is then:

$$\mathbf{H}_{\text{re}} = \frac{1}{2} \underbrace{\sum_i \alpha_i \, \mathbf{h}_i^{P'}}_{\text{rationale-weighted}} + \frac{1}{2} \underbrace{\frac{1}{M} \sum_i \mathbf{h}_i^{P'}}_{\text{mean pool}}. \tag{8}$$

This design ensures that even when RE confidence is low, the pooled representation remains a meaningful average of all claims, providing a stable gradient signal to the upstream encoder throughout training. The defendant-side representation is computed symmetrically.

Rationale Statistics. In addition to the pooled embedding vectors, four summary statistics per side are computed from the rationale scores and passed directly to the TP head as explicit confidence signals: maximum score, mean score, sum of scores, and entropy of the softmax pooling weights. High maximum scores and low entropy jointly indicate that the RE module is confident and that evidence is concentrated in a few strongly accepted claims — a pattern that should directly influence the verdict prediction.

Formally, the plaintiff-side rationale statistics are represented as

$$\mathbf{s}_P = [\max(\hat{r}^P),\ \text{mean}(\hat{r}^P),\ \text{sum}(\hat{r}^P),\ \mathcal{H}(\alpha^P)], \tag{9}$$

where $\mathcal{H}(\cdot)$ denotes entropy over the softmax pooling weights. The defendant-side statistics $\mathbf{s}_D$ are computed analogously.

3.2.4 Stage 4: Tort Prediction. The TP head synthesises the pooled evidence representations, the RE confidence statistics, and the global case vector into a final verdict prediction.

RE-Confidence Gating. Before being passed to the verdict predictor, the evidence vectors are modulated by a learned confidence gate. A small MLP takes the concatenated RE statistics from both sides as input and produces a gate vector $\mathbf{g} \in (0, 1)^d$ via sigmoid:

$$\mathbf{g} = \sigma(\text{MLP}([\mathbf{s}_P; \mathbf{s}_D])). \tag{10}$$

The gate upweights the plaintiff evidence representation by $\mathbf{g}$ and simultaneously downweights the defendant representation by $(1 - \mathbf{g})$:

$$\mathbf{H}'_P = \text{LayerNorm}(\mathbf{H}_P \odot \mathbf{g}), \quad \mathbf{H}'_D = \text{LayerNorm}(\mathbf{H}_D \odot (1 - \mathbf{g})). \tag{11}$$

Intuitively, when the RE module signals high confidence that plaintiff claims are accepted and defendant claims are rejected — reflected by high maximum scores and low entropy on the plaintiff side — the gate shifts the evidence balance toward the plaintiff, directly encoding the logical implication that strong plaintiff rationales should increase the probability of a plaintiff verdict.

Feature Construction and Attention. Rather than concatenating the evidence vectors into a flat input for the verdict classifier, we construct a set of semantic feature vectors and treat them as a short token sequence:

$$\mathbf{X} = \left\{\mathbf{H}_u,\ \mathbf{H}'_P,\ \mathbf{H}'_D,\ \mathbf{H}'_P - \mathbf{H}'_D,\ \mathbf{H}'_P \odot \mathbf{H}'_D\right\}. \tag{12}$$

The difference token captures directional contrast between plaintiff and defendant evidence, while the product token captures their agreement; both follow the interaction feature design of natural language inference models. These tokens are passed through a self-attention layer, enabling the model to discover non-trivial interactions among features — for instance, that a large difference is diagnostic only when the product token is also small. A learned query vector then aggregates the processed token sequence into a single pooled representation, in lieu of simple mean pooling.

During training, a label token derived from the ground-truth RE statistics is prepended to this sequence:

$$\mathbf{z}_{\text{label}} = \text{LayerNorm}\Big(\text{Linear}([\mathbf{s}_P^{gt}; \mathbf{s}_D^{gt}])\Big). \tag{13}$$

This provides an auxiliary supervision signal to the TP head during early epochs when the RE module is still learning, serving as a form of multi-modal teacher forcing within the feature attention block.

Verdict Prediction. The pooled representation is concatenated with both sides' rationale statistics and passed through a final MLP:

$$\hat{T} = \sigma(\text{MLP}([\mathbf{z}; \mathbf{s}_P; \mathbf{s}_D])). \tag{14}$$

3.2.5 Training Strategy.

Loss Function. We use binary cross-entropy loss for both tasks, combined in a weighted sum:

$$\mathcal{L} = \lambda_{\text{RE}} \, \mathcal{L}_{\text{RE}} + \lambda_{\text{TP}} \, \mathcal{L}_{\text{TP}}. \tag{15}$$

The loss weights λ_{RE} and λ_{TP} were selected empirically based on validation performance. Since rationale extraction is computed over multiple claims per case, its loss magnitude is naturally larger than the case-level TP loss. To prevent the RE objective from dominating optimisation, we assign a higher relative weight to the TP objective. In practice, we found that setting $\lambda_{\text{RE}} = 1.0$ and $\lambda_{\text{TP}} = 2.0$ produced the most stable convergence and best validation performance.

The TP loss is assigned a higher weight to compensate for the scale difference: $\mathcal{L}_{\text{RE}}$ is averaged over all claims (typically 10–30 per case), while $\mathcal{L}_{\text{TP}}$ is computed once per case. Without this correction, the RE gradient would dominate and the TP head would receive insufficient training signal. Samples with missing labels are excluded from loss computation via masking.

Teacher Forcing. Training a pipeline in which downstream stages depend on upstream predictions is susceptible to error propagation: a miscalibrated RE module produces poor pooled representations, which in turn cause the TP head to learn from noisy inputs. We address this through a teacher forcing schedule.

During training, the rationale scores used to compute pooling weights in Stage 3 are replaced by a convex combination of ground-truth labels and model predictions:

$$\tilde{r}_i^P(e) = \eta(e) \cdot r_i^P + \big(1 - \eta(e)\big) \cdot \hat{r}_i^P, \tag{16}$$

where $\eta(e)$ is the mixing coefficient at epoch e, linearly annealed from 1.0 to 0.0 over the first portion of training. In early epochs, the TP head thus receives near-oracle pooled evidence, allowing it to develop a meaningful decision boundary before the RE module has converged. As training progresses, the mixing shifts gradually toward the model's own predictions, culminating in fully end-to-end training.

Optimization. The full model is optimised using AdamW with layer-wise learning rate scaling: the pre-trained encoder backbone is fine-tuned at a reduced rate to preserve pre-trained representations, while all newly initialised modules use the full base rate. A linear warm-up followed by cosine decay is applied throughout. Additional training stabilisers include gradient clipping, gradient accumulation over multiple steps, and mixed-precision training for computational efficiency.

3.3 Instruction Tuning for Judgement Legal Prediction

We frame the joint RE and TP problem as an instruction-following classification task, following recent research on instruction tuning for legal classification tasks. Rather than treating RE and TP as separate structured prediction tasks, we convert each legal case into a natural language prompt and train the LLM to generate the classification decisions as text responses.

Given a structured legal case comprising undisputed facts $\mathcal{U}$, plaintiff claims $\mathcal{P} = \{P_0, P_1, \ldots, P_{n-1}\}$, and defendant claims $\mathcal{D} = \{D_0, D_1, \ldots, D_{m-1}\}$, we construct a prompt that instructs the model to: (i) identify which claims are accepted by the court (rationale extraction) and (ii) predict the overall tort decision. The model is fine-tuned on prompt which response pairs of labels so that it learns to produce the appropriate classification when given a legal case description.

3.3.1 Prompt Design. Each training instance follows the `ChatML` template with three turns: *system*, *user* and *assistant*. Figure 2 illustrates the complete structure.

System turn. Defines the model's role as a *legal reasoning assistant* and enforces the two-stage protocol: (i) extract rationales for all claims, then (ii) predict the tort decision. The strict JSON-only output schema ensures deterministic parsing at inference time.

```
{"messages": [
{
"role": "system",
"content": "You are a legal reasoning assistant.
Given the undisputed facts, plaintiff claims, and defendant
claims of a case, predict the final result in JSON format.
Output schema:
{
"tort_affirmed": <boolean>,
"accepted_plaintiff_claims": [<list of plaintiff claim IDs>],
"accepted_defendant_claims": [<list of defendant claim IDs>]
}

Rules:
- Return valid JSON only.
- Do not add explanation.
- Plaintiff claim IDs must use format like "P0", "P1".
- Defendant claim IDs must use format like "D0", "D1"."
},

{
"role": "user",
"content": "Case version: train001.json

Undisputed Facts:
- F0: <fact text>
- F1: <fact text>
...

Plaintiff Claims:
- P0: <claim text>
- P1: <claim text>

Defendant Claims:
- D0: <claim text>
- D1: <claim text>
...
},

{
"role": "assistant",
"content": "{
"tort_affirmed": true,
"accepted_plaintiff_claims": ["P0", "P1"],
"accepted_defendant_claims": ["D1", "D10"]
}"
}
]}
```

Figure 2: Prompt template for LLM-based joint RE + TP in JSON messages format. The System turn defines the task protocol and output schema. The User turn presents the structured legal case. The Assistant turn is the generation target; only this turn contributes to the training loss.

User turn. Presents the structured case with three labelled sections: `Undisputed Facts`, `Plaintiff Claims` and `Defendant Claims`. Each claim is prefixed with its identifier for unambiguous reference.

Assistant turn (target). The target is a JSON object. Only assistant tokens contribute to the loss via completion-only masking.

3.3.2 Experimental Settings. We evaluate our instruction-tuning approach on the COLIEE 2026 LJPJT dataset. Specifically, we fine-tune two open-weight instruction-tuned large language models with different model scales and characteristics:

Table 1: Training hyperparameters and settings for instruction fine-tuning experiments.

Hyperparameter	gpt-oss-20b	Qwen3.5-27B
LoRA rank r	16	16
LoRA α	32	32
LoRA dropout	0.05	0.05
LoRA target layers	q,k,v,o,gate,up,down	q,k,v,o,gate,up,down
Optimizer	Paged AdamW (32-bit)	AdamW (8-bit)
Learning rate	2×10^{-4}	2×10^{-4}
Weight decay	1×10^{-3}	1×10^{-3}
Max gradient norm	0.3	0.3
LR scheduler	Cosine	Linear
Warmup steps	5% of total steps	5% of total steps
Training epochs	2	1

- **`openai/gpt-oss-20b` [10]**: a 20-billion-parameter dense decoder with strong multilingual reasoning capabilities, selected for its performance on Japanese reasoning tasks.
- **`Qwen/Qwen3.5-27B` [12]**: a recently released open-weight large language model that achieves competitive performance among state-of-the-art open-source LLMs and supports long-context processing, making it suitable for legal document understanding.

All experiments are conducted on a single NVIDIA A100 GPU 80GB. The QLoRA configuration enables us to fine-tune both 20B and 27B parameter models within this memory constraint. Generated outputs are parsed with a regex-based JSON extractor that identifies the last valid JSON block; claim identifiers are extracted from `is_accepted` fields, and absent claims are treated as rejected.

4 Main results

We submitted three runs to the LJPJT 2026 pilot task:

- **TortNLP**: a multi-task learning model that jointly learns tort prediction and rationale extraction.
- **TortNLP2**: an instruction-tuned large language model based on `gpt-oss-20b` using QLoRA fine-tuning.
- **TortNLP4**: an instruction-tuned large language model based on `Qwen3.5-27B` using QLoRA fine-tuning.

Table 2 and 3 presents the experimental results of our three approaches on the development set for both TP and RE tasks. Our best-performing system, UIT_run3 based on `Qwen3.5-27B`, achieves 72.6% accuracy on TP and 68.2% F1-Score on RE.

Performance Comparison. The `Qwen3.5-27B` model (TortNLP4) demonstrates superior performance across both subtasks, exceeding the multi-task learning pipeline (TortNLP) by 8.6% on TP and 7.8% on RE. Compared to `gpt-oss-20b` (TortNLP2), `Qwen3.5-27B` shows improvements of 8.3 and 15.2% on TP and RE, respectively. This consistent advantage suggests that the larger model capacity and more recent architectural improvements in `Qwen3.5-27B` translate into better legal reasoning capabilities.

Multi-task Learning vs. LLM-based Approaches. The multi-task learning pipeline (TortNLP) achieves balanced performance on both tasks (64.0% TP, 60.4% RE), demonstrating the effectiveness of our hierarchical architecture with FiLM conditioning and cross-attention mechanisms. While this approach does not match the absolute performance of large language models, it offers several advantages: (i) interpretability through explicit modeling of claim-level rationales, (ii) computational efficiency with only 310M parameters, and (iii) stable training through teacher forcing and fallback-mixing pooling strategies.

Interestingly, `gpt-oss-20b` (TortNLP2) shows competitive performance on TP (64.3%) but lags significantly on RE (53.0%). This discrepancy suggests that while the model can capture high-level patterns for verdict prediction, it struggles with the fine-grained task of identifying which specific claims are accepted by the court. This may be attributed to the model's training objective not being specifically optimized for structured legal reasoning tasks.

Table 2: Formal Run Results of Tort Prediction.

Rank	Run	Team	Accuracy
1	**JAIST-LJPJT25-TP**	**JNLP**	**72.7%**
2	**TortNLP4**	**UIT-TortNLP**	**72.6%**
3	Qwen25	JNLP	68.0%
4	NOWJ1	NOWJ	67.6%
9	TortNLP2	UIT-TortNLP	64.3%
11	TortNLP	UIT-TortNLP	64.0%

Table 3: Formal Run Results of Rationale Extraction.

Rank	Run	Team	F1 (All)
1	**TortNLP4**	**UIT-TortNLP**	**68.2%**
2	Qwen3	JNLP	67.4%
3	Qwen25	JNLP	65.6%
4	NOWJ2	NOWJ	64.8%
9	TortNLP	UIT-TortNLP	60.4%
12	TortNLP2	UIT-TortNLP	53.0%

Qwen3.5-27B Advantages. The strong performance of `Qwen3.5-27B` can be attributed to several factors. First, as a recently released model, it incorporates state-of-the-art architectural improvements and has been trained on more recent, high-quality data. Second, its enhanced capacity for processing long legal documents allows it to capture comprehensive case information without aggressive truncation. Third, the instruction-tuning paradigm combined with QLoRA fine-tuning enables efficient adaptation to the legal domain while preserving the model's general reasoning abilities.

Notably, `Qwen3.5-27B`'s performance gap between TP and RE (4.4%) is smaller than that of `gpt-oss-20b`(11.3%), indicating more balanced capabilities across both subtasks. This suggests that the model successfully learns to perform joint reasoning—extracting supporting rationales while simultaneously predicting the final verdict, which aligns with the natural judicial reasoning process.

These findings suggest potential for future work combining the strengths of both paradigms: leveraging the interpretability and efficiency of structured neural architectures alongside the powerful reasoning capabilities of large language models.

5 Conclusion

In this paper, we presented our systems for the COLIEE 2026 pilot task on Legal Judgment Prediction for Japanese Tort Cases, which consists of two subtasks: Tort Prediction and Rationale Extraction. We investigated two different approaches to address this problem: a structured multi-task learning framework that jointly models rationale extraction and verdict prediction, and an instruction-tuned large language model approach using QLoRA for parameter-efficient adaptation. Experimental results demonstrate that while the multi-task learning architecture provides competitive performance with strong interpretability and efficiency, instruction-tuned large language models achieve the best overall results. In particular, our Qwen3.5-27B based system ranks second in Subtask 1 with 72.6% accuracy and first in Subtask 2 with an F1-score of 68.2%. These findings highlight the effectiveness of instruction tuning with parameter-efficient fine-tuning for legal judgment prediction tasks.

For future work, we plan to further explore hybrid approaches that combine the structured reasoning capabilities of multi-task learning models with the powerful contextual understanding of large language models. Such integration may improve both predictive performance and interpretability in complex legal reasoning tasks.

References

[1] Rich Caruana. 1997. Multitask Learning. *Machine Learning* 28 (07 1997). doi:10.1023/A:1007379606734

[2] Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. QLoRA: Efficient Finetuning of Quantized LLMs. arXiv:2305.14314 [cs.LG] https://arxiv.org/abs/2305.14314

[3] Randy Goebel, Yoshinobu Kano, mi-young Kim, Juliano Rabelo, Ken Satoh, and Masaharu Yoshioka. 2024. Overview and Discussion of the Competition on Legal Information, Extraction/Entailment (COLIEE) 2023. *The Review of Socionetwork Strategies* 18 (01 2024). doi:10.1007/s12626-023-00152-0

[4] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. LoRA: Low-Rank Adaptation of Large Language Models. arXiv:2106.09685 [cs.CL] https://arxiv.org/abs/2106.09685

[5] Kazuma Kadowaki and Yoshinobu Kano. 2026. Japanese Legal Judgment Prediction Using ModernBERT and Generative AI-based Information Extraction. *The Review of Socionetwork Strategies* (03 2026). doi:10.1007/s12626-026-00201-4

[6] Yoshinobu Kano, Randy Goebel, Mi-Young Kim, Calum Kwan, Ken Satoh, Hiroaki Yamada, and Masaharu Yoshioka. 2026. An Overview of the COLIEE 2026 Competition: Legal Case Law and Statute Law Information Retrieval and Entailment. In *Proceedings of the 21st International Conference on Artificial Intelligence and Law (ICAIL 2026)*.

[7] LLM-jp, :, Akiko Aizawa, Eiji Aramaki, Bowen Chen, Fei Cheng, Hiroyuki Deguchi, Rintaro Enomoto, Kazuki Fujii, Kensuke Fukumoto, Takuya Fukushima, Namgi Han, Yuto Harada, Chikara Hashimoto, Tatsuya Hiraoka, Shohei Hisada, Sosuke Hosokawa, Lu Jie, Keisuke Kamata, Teruhito Kanazawa, Hiroki Kanezashi, Hiroshi Kataoka, Satoru Katsumata, Daisuke Kawahara, Seiya Kawano, Atsushi Keyaki, Keisuke Kiryu, Hirokazu Kiyomaru, Takashi Kodama, Takahiro Kubo, Yohei Kuga, Ryoma Kumon, Shuhei Kurita, Sadao Kurohashi, Conglong Li, Taiki Maekawa, Hiroshi Matsuda, Yusuke Miyao, Kentaro Mizuki, Sakae Mizuki, Yugo Murawaki, Akim Mousterou, Ryo Nakamura, Taishi Nakamura, Kouta Nakayama, Tomoka Nakazato, Takuro Niitsuma, Jiro Nishitoba, Yusuke Oda, Hayato Ogawa, Takumi Okamoto, Naoaki Okazaki, Yohei Oseki, Shintaro Ozaki, Koki Ryu, Rafal Rzepka, Keisuke Sakaguchi, Shota Sasaki, Satoshi Sekine, Kohei Suda, Saku Sugawara, Issa Sugiura, Hiroaki Sugiyama, Hisami Suzuki, Jun Suzuki, Toyotaro Suzumura, Kensuke Tachibana, Yu Takagi, Kyosuke Takami, Koichi Takeda, Masashi Takeshita, Masahiro Tanaka, Kenjiro Taura, Arseny Tolmachev, Nobuhiro Ueda, Zhen Wan, Shuntaro Yada, Sakiko Yahata, Yuya Yamamoto, Yusuke Yamauchi, Hitomi Yanaka, Rio Yokota, and Koichiro Yoshino. 2024. LLM-jp: A Cross-organizational Project for the Research and Development of Fully Open Japanese LLMs. arXiv:2407.03963 [cs.CL] https://arxiv.org/abs/2407.03963

[8] Dat Nguyen, Phuong Nguyen, Huy Chu, Son Luu, Nguyen-Hoang Chu, Trung Vo, and Le Nguyen. 2026. Enhancing Legal Text Processing and Structural Analysis with Large Language Models at COLIEE 2025. *The Review of Socionetwork Strategies* (03 2026). doi:10.1007/s12626-026-00211-2

[9] Hoang-Trung Nguyen, Tan-Minh Nguyen, Xuan-Bach Le, Tuan-Kiet Le, Khanh-Huyen Nguyen, Ha-Thanh Nguyen, Thi-Hai-Yen Vuong, and Le-Minh Nguyen. 2025. NOWJ@COLIEE 2025: A Multi-stage Framework Integrating Embedding Models and Large Language Models for Legal Retrieval and Entailment. arXiv:2509.08025 [cs.CL] https://arxiv.org/abs/2509.08025

[10] OpenAI. 2025. gpt-oss-120b & gpt-oss-20b Model Card. arXiv:2508.10925 [cs.CL] https://arxiv.org/abs/2508.10925

[11] Ethan Perez, Florian Strub, Harm de Vries, Vincent Dumoulin, and Aaron Courville. 2018. FiLM: Visual Reasoning with a General Conditioning Layer. In *Proceedings of the AAAI Conference on Artificial Intelligence*. https://arxiv.org/abs/1709.07871

[12] Qwen Team. 2026. Qwen3.5: Towards Native Multimodal Agents. https://qwen.ai/blog?id=qwen3.5

[13] Hayato Tsukagoshi, Shengzhe Li, Akihiko Fukuchi, and Tomohide Shibata. 2025. ModernBERT-Ja. https://huggingface.co/collections/sbintuitions/modernbert-ja-67b68fe891132877cf67aa0a. https://huggingface.co/collections/sbintuitions/modernbert-ja-67b68fe891132877cf67aa0a

[14] Benjamin Warner, Mehdi Cherti, Artem Kochurov, Dan Jean, Antoine Bruni, Vinay Chiley, Jacob Portes, Nicholas Hall, Mansheej Paul, Luke Sykora, Sylvain Gugger, and Jonathan Frankle. 2024. ModernBERT: Smarter, Better, Faster, Longer. *arXiv preprint arXiv:2412.13663* (2024). https://arxiv.org/abs/2412.13663

[15] Sabine Wehnert, Bhavya Chovatta, and Ernesto De Luca. 2025. LLMs, Knowledge Graphs, and Hybrid Search: Task-Specific Approaches to Legal AI in COLIEE.

[16] Jason Wei, Maarten Bosma, Vincent Y. Zhao, et al. 2022. Finetuned Language Models Are Zero-Shot Learners. *arXiv preprint arXiv:2109.01652* (2022).

[17] Hiroaki Yamada, Takenobu Tokunaga, Ryutaro Ohara, Akira Tokutsu, Keisuke Takeshita, and Mihoko Sumida. 2024. Japanese tort-case dataset for rationale-supported legal judgment prediction. *Artificial Intelligence and Law* 33, 3 (May 2024), 783–807. doi:10.1007/s10506-024-09402-0

[18] Wenmian Yang, Weijia Jia, Xiaojie Zhou, and Yutao Luo. 2019. Legal Judgment Prediction via Multi-Perspective Bi-Feedback Network. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence (IJCAI-2019)*. International Joint Conferences on Artificial Intelligence Organization, 4085–4091. doi:10.24963/ijcai.2019/567

[19] Yunong Zhang, Xiao Wei, and Hang Yu. 2024. HD-LJP: A Hierarchical Dependency-based Legal Judgment Prediction Framework for Multi-task Learning. *Knowledge-Based Systems* 299 (05 2024), 112033. doi:10.1016/j.knosys.2024.112033

[20] Haoxi Zhong, Zhipeng Guo, Cunchao Tu, Chaojun Xiao, Zhiyuan Liu, and Maosong Sun. 2018. Legal Judgment Prediction via Topological Learning. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii (Eds.). Association for Computational Linguistics, Brussels, Belgium, 3540–3549. doi:10.18653/v1/D18-1390

SCIlab at COLIEE 2026: Knowledge Graph-Augmented Retrieval and Reasoning for Legal Question Answering

Jumma Nishida
Institute of Science Tokyo
Tokyo, Japan
nishida.j.207e@m.isct.ac.jp

Ziwei XU
Institute of Science Tokyo
Tokyo, Japan
xu.ziwei@iee.eng.isct.ac.jp

Ryutaro Ichise
Institute of Science Tokyo
Tokyo, Japan
ichise@iee.e.titech.ac.jp

Abstract

In Legal QA, conventional text retrieval struggles with complex statutory structures and citation relationships. To address this, we propose a multi-stage retrieval and reasoning approach using a legal-domain-specific knowledge graph (KG) for the COLIEE 2026 SL-QA task. Our KG, based on the Japanese Civil Code, explicitly structures provisions, keywords, and citations. We conduct an initial text-based retrieval, followed by graph-based expansion leveraging citation edges, capturing provisions missed by superficial similarity. Feeding this structural context into an LLM outperforms baselines like Text-RAG and LightRAG, highlighting the effectiveness of integrating legal-specific structures into retrieval and reasoning pipelines.

CCS Concepts

• **Computing methodologies** → **Information extraction**; • **Applied computing** → **Law**; • **Information systems** → **Question answering**.

Keywords

Legal Knowledge Graph, Information Extraction, Retrieval-Augmented Generation, Large Language Models, Question Answering

1 Introduction

Recent advancements in natural language processing (NLP) technologies, particularly large language models (LLMs), have accelerated research in Legal NLP. Among these, legal question answering (Legal QA) stands out as a core task for automating legal assistance for practitioners and the general public. Unlike general open-domain QA, Legal QA requires not only generating the correct answer but also identifying the exact statutory provisions (legal basis) to be relied upon, followed by rigorous logical reasoning based on those provisions.

However, legal documents present unique challenges distinct from general texts. First, the heavy reliance on technical terminology frequently results in a superficial vocabulary mismatch between the user query and the statutory provisions. Second, legal documents exhibit a highly complex structure where a single provision rarely stands alone. Meaning is often dispersed across conditional clauses, exceptions, and explicit cross-references to other provisions (e.g., "the preceding article" or "Articles X through Y"). Due to these characteristics, conventional retrieval-augmented generation (RAG) approaches—which naively divide text into chunks and compute similarities—sever the vital context and dependencies between provisions. Consequently, it becomes exceedingly difficult to comprehensively acquire the information necessary for accurate legal reasoning.

To rigorously evaluate how well a system can navigate these domain-specific complexities, we focus on a standardized benchmark. The target of this study is Task 3 (Statute retrieval and entailment for yes/no questions: SL-QA) of COLIEE 2026.[5] This task involves determining the truth value of queries based on the Japanese Bar Exam short-answer questions, utilizing the provisions of the Japanese Civil Code. To derive the correct answer in this task, it is essential to possess the reasoning capability to comprehensively and precisely retrieve multiple relevant provisions and correctly interpret their interrelationships.

Recently, GraphRAG, which represents entities and relationships within documents as a graph structure, has gained attention as an approach to handle complex structural dependencies. However, conventional GraphRAG methods designed for general open domains struggle to accurately extract and represent the domain-specific hierarchical structures (e.g., provisions and headings) and the dense cross-reference relationships unique to legal documents. Meanwhile, recent advancements in knowledge graph (KG)-based retrieval demonstrate the effectiveness of executing local text retrieval and global structural retrieval separately before integrating them. To achieve high-precision complex reasoning (SL-QA) in the legal domain, a mechanism that integrates not only the superficial semantics of the text but also these structural dependencies is indispensable.

Therefore, rather than relying on a generic framework, this study proposes a multi-stage retrieval approach tailored specifically to the legal domain. This approach integrates multifaceted textual and structural information by constructing a custom legal knowledge graph. Focusing on the Japanese Civil Code, we construct a Knowledge Graph that defines "Article," "Caption," and "Keyword" as entities. In addition to the conceptual relationships connecting them, we explicitly model the citation relationships (`refer_to`) between provisions. During inference, our system retrieves lexically and semantically relevant provisions by combining node retrieval based on keywords extracted by the LLM with vector retrieval based on the full query text. Furthermore, we perform a 1-hop graph expansion using the citation relationships on the KG to complement

COLIEE 2026, Singapore

the referenced provisions. Finally, the context integrating the retrieved text and the structural information of the graph is fed into the LLM to perform the truth-value prediction.

Through these efforts, our primary contributions are threefold. First, we propose a multi-stage retrieval and reasoning framework tailored to the complex structural dependencies of statutory texts. Second, we construct a domain-specific knowledge graph for the Japanese Civil Code that explicitly models citation relationships. Finally, we show that this approach significantly outperforms baselines like Text-RAG and LightRAG on the COLIEE 2026 SL-QA task.

2 Task Description

2.1 SL-QA Task

Task 3 of COLIEE 2026, titled "Statute retrieval and entailment for yes/no questions" (SL-QA), is formulated as a problem combining statutory provision retrieval with legal reasoning. In this task, given the provisions of the Japanese Civil Code and a legal statement (query) written in Japanese, the system is required to determine whether the query is true or false based on the provided statutes.

Specifically, for each query, the system must determine its truth value using the relevant provisions as the basis. The output is a binary classification, returning "Y" if the query is legally correct and "N" otherwise.

This task requires not only understanding the semantic relationship between the query and the provisions but also performing legally valid reasoning based on their content and explicitly identifying the supporting provisions. Therefore, the primary objective of this task is to evaluate the capabilities of legal document understanding and complex legal reasoning.

2.2 Datasets

For the SL-QA task, a dataset comprising the provisions from Parts 1 to 4 of the Japanese Civil Code (787 provisions) and queries answerable based on these provisions is provided. The queries are derived from the choices of true/false questions in the Japanese Bar Exam's short-answer test.

The dataset contains a total of 1,223 queries for training and evaluation. Each query is annotated with a ground-truth label ("Y" or "N") and the IDs of the provisions that serve as the basis for the judgment. In this study, we utilize 73 queries extracted from the Reiwa 6 (2024) Bar Exam as the evaluation data.

2.3 Metrics

System performance is evaluated based on the agreement between the submitted predictions and the ground-truth labels. In this task, $WS\_Accuracy$ and the F_2 score, as defined by the workshop, are utilized as evaluation metrics.

$$WS\_Accuracy = \frac{N_{correct}}{N_{all}} \tag{1}$$

Here, $N_{correct}$ represents the number of queries for which all relevant provisions were successfully retrieved and the truth value of the query was correctly predicted based on that retrieved information, while N_{all} represents the total number of queries.

$$F_2 = \frac{5 \times Precision \times Recall}{4 \times Precision + Recall} \tag{2}$$

The primary evaluation metric is $WS\_Accuracy$, and in cases where the $WS\_Accuracy$ is tied between systems, the F_2 score is employed as a secondary metric.

3 Related Work

To clarify the background of our proposed method, this section reviews three related research areas. First, Section 3.1 discusses Legal Article Retrieval, which focuses on the challenge of finding the relevant statutory provisions. Second, Section 3.2 covers Legal QA, which focuses on how to reason and make judgments based on those retrieved provisions. Finally, Section 3.3 introduces GraphRAG. While the first two sections describe the domain-specific tasks and their challenges, GraphRAG represents the core methodology we adopt to solve them by explicitly modeling the structural relationships of legal texts.

3.1 Legal Article Retrieval

Information retrieval targeting legal documents is a fundamental challenge in Legal NLP, where the search and identification of relevant statutory provisions constitute a critical research theme. It has been pointed out that, compared to general texts, legal documents possess unique characteristics: they are inherently lengthy, heavily laden with technical terminology, and their semantics are often dispersed across multiple provisions due to conditions, exceptions, and cross-reference relationships[1].

The Competition on Legal Information Extraction/Entailment (COLIEE) is an international competition aimed at advancing foundational technologies for legal document processing[12]. Traditionally, Task 3 (Statute Law Retrieval) in COLIEE has been formulated as the task of retrieving relevant provisions from a corpus of Japanese Civil Code articles in response to a legal query[3]. In recent iterations of Task 3, multi-stage retrieval pipelines combining lexical search methods like BM25[13] with dense retrieval and reranking have become the mainstream approach. For instance, JNLP[11] proposed a multi-stage pipeline comprising pre-retrieval based on dense retrieval, fine-tuning for query-article relevance judgment, and reranking, thereby demonstrating high retrieval performance. Similarly, CAPTAIN[10] achieved strong results in Task 3 through an approach that combines zero-shot retrieval using LLMs with fine-tuned LLMs.

Furthermore, INFA[9] proposed a method that represents the hierarchical structure of statutes as a graph and enhances provision representations using Graph Neural Networks. This underscores the utility of explicitly leveraging the structural information of provisions in legal retrieval. Additionally, OVGU[15] proposed an approach that constructs and retrieves from a knowledge graph integrating statutory provisions, commentaries, and precedent data.

Consequently, beyond mere text similarity-based retrieval, the utilization of multi-stage retrieval and structural information has become crucial in legal provision retrieval. Building upon these prior studies, this research aims to improve relevant provision retrieval by employing a KG that explicitly integrates headings, keywords, and citation relationships.

3.2 Question Answering in the Legal Domain

The domain-specific difficulties of legal documents discussed in the previous subsection are particularly pronounced in legal question answering (Legal QA). Specifically, adequately retrieving relevant provisions for a given question and, when necessary, integrating multiple provisions or paragraphs to make a judgment constitutes a central challenge. Recent studies emphasize that Legal QA is not merely about document retrieval or answer generation; rather, the critical issue lies in how to bridge the acquisition of supporting provisions with the subsequent reasoning based upon them[1].

To address this challenge, numerous studies have adopted retrieval-based frameworks. Khazaeli et al. improved answer retrieval performance in Legal QA by combining sparse retrieval via BM25 with dense retrieval using semantic vectors, followed by reranking with BERT[6]. Louis et al. proposed a retrieve-then-read approach where relevant statutes are first retrieved before an LLM generates the answer[8]. Their method enhances transparency by presenting the supporting statutory paragraphs alongside the answer. These studies demonstrate the paramount importance of acquiring appropriate legal grounding as a first step in Legal QA.

On the other hand, a single provision is often insufficient in Legal QA, necessitating the joint handling of multiple pieces of evidence. Li et al. proposed a method utilizing a Graph-Based Evidence Retrieval and Aggregation Network (GESAN) to perform reasoning while aggregating multiple retrieved pieces of evidence[7] . Furthermore, Sovrano et al. introduced DiscoLQA[14], which focuses on discourse structures such as Elementary Discourse Units and Abstract Meaning Representations to more accurately retrieve crucial segments within legal documents. These works illustrate that considering the relationships between evidence and document structure, rather than relying solely on superficial lexical matching, is highly effective in Legal QA.

The SL-QA task targeted in this study is also positioned as a type of Legal QA, as it involves determining the truth value of a query based on statutory provisions. However, rather than generating fluent responses, the crux of this task lies in appropriately acquiring the relevant supporting provisions and determining the truth value by taking into account the interrelationships among those provisions. Therefore, by explicitly providing the LLM with structural information, including citation relationships, this study aims to realize more robust truth-value prediction based on multiple supporting provisions.As discussed in the previous subsections, leveraging the structural information of legal documents is essential for effective retrieval. To achieve this, GraphRAG, which utilizes a knowledge graph to map these relationships, is a highly effective solution. Therefore, this study adopts a GraphRAG-based approach as our foundational framework.

3.3 GraphRAG

Retrieval-Augmented Generation (RAG) is widely used as a framework to provide LLMs with external knowledge, thereby generating responses while mitigating knowledge deficits and hallucinations. Conversely, conventional text-based RAG typically retrieves documents as isolated chunks, making it difficult to adequately handle inter-document relationships and structural dependencies. Particularly in legal documents, where cross-references between provisions and the hierarchical structure of legal codes are critical, searches relying solely on superficial text similarity are prone to limitations.

To overcome this challenge, GraphRAG has been proposed, which represents entities and relationships within documents as a KG to be utilized for retrieval and generation. Edge et al. enabled answering global questions across entire documents by hierarchically clustering extracted entities into communities and generating prior summaries using LLMs[2]. However, the enormous computational cost of summary generation and the difficulty of dynamic updates remained significant issues. In response, Guo et al. proposed LightRAG[4], which employs a dual-level retrieval integrating local entity search and global relationship search. This approach eliminates the need for high-cost prior summarization, enabling low-computational-overhead and incremental knowledge updates.

These studies demonstrate the efficacy of explicitly handling relationships between entities and document structures in RAG, going beyond mere text retrieval. Aligned with this trajectory, the present study aims to improve relevant provision retrieval and truth-value prediction by employing a KG that integrates headings, keywords, and citation relationships specific to statutory provisions.

4 Methodology

4.1 Overview

In the SL-QA task, it is necessary to understand the semantic relationship between a query and statutory provisions to determine the truth value of the query based on those provisions. However, because legal documents possess a complex structure involving citation relationships between provisions (e.g., “the preceding article” or “Articles X through Y”), it is often difficult to appropriately retrieve the relevant provisions using simple text retrieval alone. These citation relationships are frequently described implicitly in the text, making it challenging to explicitly identify which provisions are being referenced. Therefore, to correctly retrieve relevant provisions and utilize them during reasoning, it is imperative to explicitly interpret and leverage these citation relationships.

To address this, we propose a method utilizing a knowledge graph (KG) to explicitly represent the citation relationships between provisions. Furthermore, by combining conceptual relevance based on keywords and headings with contextual similarity via vector search, our approach aims to retrieve relevant provisions from multiple perspectives.

Figure 1 illustrates the overview of the proposed system. Our method primarily consists of three steps. First, we construct a legal knowledge graph using statutory provisions and external data. Second, for an input query, we retrieve relevant provisions through a multi-stage retrieval process that combines keyword-based search and vector search, followed by graph expansion based on citation relationships. Finally, the retrieved provision chunks and the entity-relation information obtained from the KG are integrated into a structured context. This context is then fed into a large language model (LLM) to perform reasoning and determine the truth value of the query.

4.2 KG Construction

In this study, we construct a KG to structurally represent statutory provisions and their related information, thereby explicitly

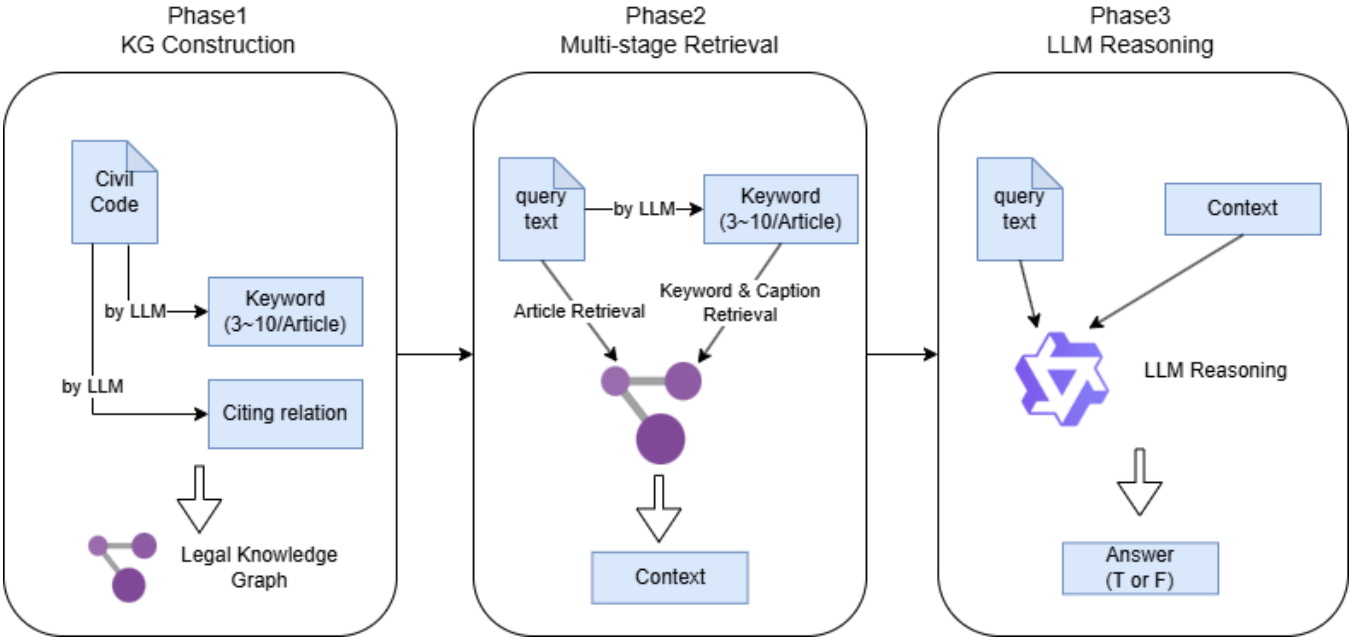


Figure 1: Overview of the proposed system

handling the relationships between provisions. In legal documents, individual provisions rarely exist in isolation; rather, their meanings are supplemented, or specific conditions are appended, by referencing other provisions. To adequately handle such structural dependencies, it is effective to represent them not merely as a collection of texts but as a graph structure comprising entities and relations.

Our KG centrally features provision entities and integrates conceptual information, such as headings and keywords, along with the citation relationships between provisions. This allows the system to process information from three distinct perspectives: the content, the theme, and the structure of the provisions.

4.2.1 Entity Definition. First, we categorize and explain the entities created for the KG. The entities in our KG are classified into the following three types:

- **Article**: An entity representing each statutory provision, corresponding to the text of the provision. In this study, the title of each entity is set to "Article X," and the description stores the corresponding text. The target comprises 787 provisions, resulting in the generation of 787 Article entities. These serve as the central nodes in our research.
- **Caption**: An entity representing the heading of a provision. Most provisions are accompanied by a corresponding heading. The Caption functions as a concise indicator of the subject or main point of the provision. Among the 787 provisions, 742 have corresponding headings, which are generated as Caption entities.
- **Keyword**: An entity representing keywords related to the content of a provision, introduced to capture the conceptual relevance between provisions. We automatically extract these keywords from each provision using an LLM.

Specifically, we prompt the LLM with the text of each provision to extract keywords across multiple categories, taking into account the semantic structure of the provision. We define the following categories:

– **Rights/Duties**: Legally recognized rights and imposed obligations (e.g., ownership, obligation to pay).
– **Persons**: Entities that are the subjects of rights or duties (e.g., manager, pledgee).
– **Legal Effects/Outcomes**: Legal effects or conclusions that arise when certain requirements are met (e.g., invalidity, rescission).
– **Legal Actions**: Acts or procedures that hold legal significance (e.g., claim for damages, mandate).
– **Legal States/Situations**: Legal states or situations concerning specific subjects or objects (e.g., mental disability, minor).
– **Legal Concepts**: Abstract concepts or objects used in law (e.g., co-owned property, appurtenance).

Extracting keywords by category in this manner allows for the explicit representation of elements such as "subjects," "actions," "concepts," and "effects" contained within the provisions. This enables a more structural understanding of the provision's content.

Although there is no gold-standard dataset for legal keyword extraction and category assignment in this task, manual inspection confirmed that the extracted keywords generally captured legally salient concepts and were assigned to appropriate categories.

Furthermore, analysis of the `has_keyword` relations showed that more than 50to only a small number of provisions, indicating that many keywords captured provision-specific legal

concepts rather than overly generic terms. These unique keywords functioned effectively as retrieval anchors, helping connect semantically related queries to the corresponding statutory provisions even when direct lexical overlap was limited.

By introducing these entities, we can represent auxiliary information, such as themes and concepts, on the graph in addition to the superficial textual information of the provisions.

4.2.2 Relation Definition. Next, we describe the relations defined to construct the KG. To explicitly represent the connections between entities, we classify relations according to their roles into the following three types:

- **has_caption**: A relation connecting an Article entity to its corresponding Caption entity. It serves an auxiliary role in expressing the subject or main point of the provision.
- **has_keyword**: A relation connecting an Article entity to a related Keyword entity. It expresses conceptual information contained in the provision and captures thematic connections between relevant provisions through shared keywords.
- **refer_to**: A relation representing citation relationships between provisions, explicitly denoting references to other provisions.

While `has_caption` and `has_keyword` supplement the content of a provision by adding thematic and conceptual information, `refer_to` expresses the structural relationship between provisions and is a particularly crucial relation in this study.

Legal documents frequently refer to other provisions using elliptical expressions such as "the preceding article," "the following article," or "Articles X through Y." While humans can easily interpret such expressions from context, text-based methods struggle to accurately identify the referenced targets and utilize them properly in reasoning.

In this study, we analyze these referential expressions and assign `refer_to` relations between the corresponding Article entities, thereby explicitly representing the citation relationships on the KG. This enables the mechanical handling of dependencies and connections between provisions.

Furthermore, these explicitly defined citation relationships are not only utilized to retrieve an expanded set of relevant provisions in subsequent retrieval stages but also function as auxiliary information during LLM reasoning. This relation plays an especially vital role when understanding a particular provision requires reference to another.

4.2.3 Construction Process. The KG is constructed from the statutory provision texts and related information through the following procedure.

First, each provision is registered as an Article entity, holding the provision text as its description. Additionally, the heading information accompanying each provision is generated as a Caption entity and connected to the corresponding Article entity via the `has_caption` relation.

Next, using the text of each provision as input, we perform keyword extraction using an LLM to generate Keyword entities based on categories such as persons, actions, concepts, and effects.

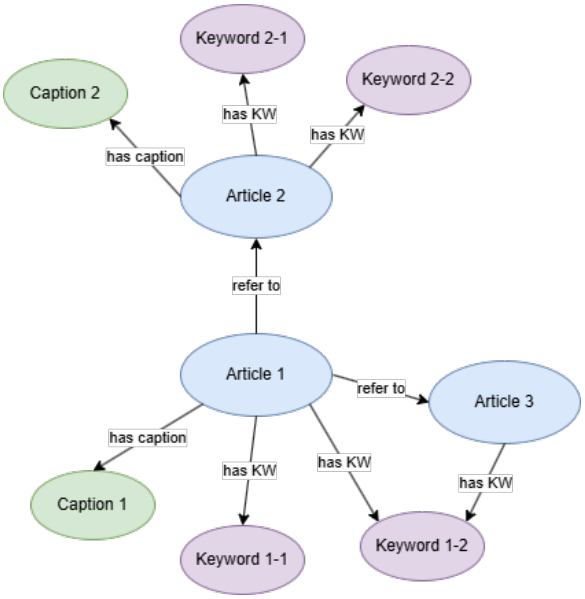


Figure 2: Structure of the constructed knowledge graph

These keywords are connected to the corresponding Article entities using the `has_keyword` relation.

Furthermore, referential expressions (e.g., "the preceding article," "the following article," "Articles X through Y") within the provisions are extracted through a combination of LLM processing and manual verification. The LLM demonstrated high reliability in identifying these relations; manual verification was required for only a few percent of the cases to resolve minor ambiguities or complex citations, ensuring the integrity of the `refer_to` edges.

Ultimately, by integrating these entities and relations, we construct a KG that encompasses the content, themes, conceptual information, and structural relationships of the provisions. As illustrated in Figure 2, this KG features a structure where Article entities serve as the central nodes, with Caption and Keyword entities connected peripherally, and Article entities interconnected via `refer_to` relations.

4.3 Multi-stage Retrieval

We propose a multi-stage retrieval approach utilizing the KG to progressively acquire Article entities relevant to the query. To complementarily retrieve relevant provisions that are difficult to obtain using a single retrieval method, we combine a keyword-based search with a dense retrieval method using the full text of the query.

The proposed method broadly consists of the following three stages.

4.3.1 Keyword-based Node Retrieval. First, we use an LLM to extract keywords from the input query. The extracted keywords fall under categories such as persons, legal actions, legal concepts, and legal effects, effectively representing the semantic features of the query. The keyword categories and prompts used during extraction are identical to those described in Section 4.2.

Next, using the extracted keywords, we perform a similarity search against the Keyword and Caption entities on the KG. In this process, embeddings are generated using each extracted keyword string on the query side, and the names of the entities on the target side. Entities with high cosine similarity between these vectors are selected.

Subsequently, we traverse the `has_keyword` relations connected to the retrieved Keyword entities and the `has_caption` relations connected to the Caption entities to obtain the corresponding Article entities. This process allows the system to acquire conceptually related provisions or provisions that define crucial vocabulary as background knowledge, even when there is low superficial lexical overlap with the query.

4.3.2 Vector-based Article Retrieval. Following this, we perform a vector search using the full text of the query as input to retrieve highly relevant Article entities based on text similarity. Here, the full query text is used on the query side, while a concatenation of the Article entity is used as the target. These are vectorized by the embedding model, and entities with high cosine similarity are selected.

4.3.3 Graph-based Expansion. Finally, starting from the Article entities retrieved in the preceding stages (Section 4.3), we expand the set of relevant provisions by performing a 1-hop traversal of the `refer_to` relations on the KG.

Specifically, for each Article entity, we retrieve its referenced provisions and add them as new relevant provision candidates. This process makes it possible to complementarily acquire relevant provisions via citation relationships, even if those provisions were not directly retrieved in the initial searches.

In legal documents, the content of a provision is frequently stipulated based on the premise of other provisions. The referenced provisions may restrict or expand the scope of application of the referring provision. In such cases, failing to retrieve the referenced provisions makes it difficult to accurately interpret the meaning of the referring provision, potentially leading to errors in the final judgment.

Therefore, expanding the provision set through graph traversal using the `refer_to` relation enables a more comprehensive retrieval of provisions that accounts for inter-provision dependencies. Such expansion based on structural relationships plays a vital role in supplementing the prerequisite knowledge required for subsequent reasoning.

Through this multi-stage retrieval, our system integrates keyword-level precision with broader structural context. While keywords serve as efficient anchors to pinpoint relevant nodes, the subsequent graph expansion re-integrates the logical relationships—such as those between legal requirements and effects—by traversing citation edges. Furthermore, by providing the LLM with the complete original text of all identified provisions rather than isolated keywords, we ensure that the full legal discourse is preserved for the final reasoning stage.

4.4 LLM Reasoning

In this study, we integrate the KG elements and textual information acquired through the multi-stage retrieval to perform reasoning using an LLM. Rather than simply inputting the provision texts, we construct a context that integrates the structural information obtained during the retrieval process, thereby realizing more accurate legal reasoning.

First, through the multi-stage retrieval described in the previous section, relevant entities are acquired via keyword-based node retrieval, vector-based article retrieval, and graph expansion. Specifically, Caption and Keyword entities related to the keywords, Article entities derived from the full-query search, and additional Article entities expanded via the `refer_to` relation are aggregated.

Next, these retrieval results are transformed into a text format processable by the LLM. Rather than inputting the raw graph structure, we construct the context by enumerating entity information (name, type, and description), the relationships between entities (source, target, and relation type), and the associated provision texts.

By explicitly providing this integrated information from the KG and retrieval results, the LLM can perform reasoning that accounts for inter-provision relationships and auxiliary information. In particular, the dependencies between provisions made explicit by the `refer_to` relation function effectively to supplement interpretations in cases where individual provisions alone are insufficient.

The constructed context is formatted into a prompt in the following order: "Instruction," "Retrieved Context," "Query," and "Output Constraints" and then fed into the LLM. As an output constraint, we specify that the answer must be exclusively either "Y" or "N". To prevent excessive bloating of the context, we impose a maximum token limit of 4,096 tokens on the entity and relation information.

As described above, by converting the KG retrieval results into a text context suitable for reasoning and inputting it into the LLM, our method achieves more consistent legal reasoning that considers structural information, compared to simple text input.

5 Results

5.1 Experimental Settings

Our team, SCIlab, submitted our system under the run ID SCIlab-1. In our experiments, we used Qwen/Qwen3-32B-AWQ[1] as the LLM for keyword extraction and truth-value prediction. For the dense retrieval component, we employed Qwen/Qwen-embedding-8b[2] to generate vector representations for both the queries and the legal provisions.

5.2 Overall Performance

In this section, we evaluate the overall performance of the proposed method. First, we present the final prediction performance of our approach on the SL-QA task, comparing it with external participating systems and representative baseline methods. Note that the comparison with external systems utilizes the results on the official test set, while the comparison with baseline methods and the subsequent ablation studies employ the evaluation set.

First, Table 1 presents the comparison results with top-ranked external systems on the official test set (R07). As the evaluation metric, we use *WS_Accuracy*, defined in Section 2.3. Our proposed system, SCIlab-1, achieved a *WS_Accuracy* of 0.8049 and a Recall of

[1] https://huggingface.co/Qwen/Qwen3-32B-AWQ
[2] https://huggingface.co/Qwen/Qwen3-Embedding-8B

Table 1: Comparison with top-ranked external systems on the official test set (R07).

Method	WS_Accuracy	F2 score	Recall
NOWJ	0.9512	0.2224	1.0000
JNLP	0.9024	0.7796	1.0000
SCIlab-1(ours)	0.8049	0.0612	0.9268

Table 2: Comparison with baseline methods on the evaluation set (R06).

Method	Accuracy (R06)
Text-RAG	0.671
LightRAG	0.658
SCIlab-1(ours)	**0.753**

0.9268. While this recall indicates that the overall retrieval pipeline, including graph expansion, succeeds in retrieving the majority of the necessary provisions, it falls short of the perfect recall (1.0000) shown by the top-ranked systems, NOWJ and JNLP.

To further analyze the contribution of each retrieval stage, we conducted an analysis on the R07 dataset. Among the 90 gold relevant provisions, 79 were successfully retrieved through keyword-based retrieval and full-text provision retrieval prior to graph expansion. In addition, graph expansion retrieved 5 additional relevant provisions that were not obtained before expansion.

Furthermore, among the 79 retrieved provisions, 6 were identified only through keyword-based retrieval and were not retrieved by full-text provision matching alone.

These results suggest that keyword-based retrieval and graph expansion complement conventional full-text retrieval by improving coverage of semantically related legal provisions.

Next, Table 2 shows the comparison results with representative baseline methods on the evaluation set (R06). In this study, we established the following methods for comparison:

- **Text-RAG**: A method that does not utilize a KG. It retrieves the top provisions based solely on the vector similarity between the full query text and the provision texts, and inputs them into the LLM for reasoning.
- **LightRAG**: A method based on a general-purpose KG-RAG framework. It constructs a KG using triples extracted from legal provision texts via an LLM, and performs retrieval and reasoning.
- **Proposed Method**: Our approach, which integrates keyword retrieval, full-text vector retrieval, and graph expansion via `refer_to` relations.

It should be noted that Table 2 reports simple True/False (T/F) accuracy instead of `WS_Accuracy`. Since our method uses dynamic graph expansion, the number of retrieved provisions varies per query, unlike fixed-chunk baselines. Furthermore, as the LightRAG baseline decomposes provisions into sub-article entities during KG construction, it is inherently difficult to measure whether specific legal provisions were "correctly" retrieved in their entirety. Therefore, to purely evaluate the effect of structured context on the LLM's reasoning performance without the confounding factor of retrieval penalties, we adopted T/F accuracy as the most appropriate metric for this comparison.

The proposed method demonstrates higher performance compared to the Text-RAG baseline, confirming the effectiveness of retrieval and context construction that account for the structural relationships unique to legal documents. Furthermore, our method outperforms LightRAG, suggesting that a KG explicitly incorporating the reference structure, headings, and keyword information of legal provisions is more effective for this task than a general-purpose, triple-based KG construction.

From the above results, the proposed method exhibits competitive performance on the official test set and achieves higher accuracy on the evaluation set compared to both simple text similarity-based retrieval and a general-purpose KG-RAG method. These findings underscore the effectiveness of considering the domain-specific structural relationships inherent in legal documents.

5.3 Failure Case Analysis

In this section, we conduct a qualitative analysis to understand the behavior of the proposed method in greater detail. Specifically, we perform a failure analysis targeting misclassified cases and examine how the retrieved set of provisions influenced the final judgment. Table 3 illustrates representative failure cases.

Through the analysis of these failure cases, we identified that the causes of misclassification can be primarily categorized into the following three factors:

First, the limitations of the LLM's legal reasoning capabilities and condition application (e.g., cases similar to R06-01-A in Table 3). Although the necessary provisions are successfully retrieved, instances were observed where the LLM could not accurately interpret or integrate their application conditions and exception clauses. In the legal domain, principles and exceptions are frequently described across multiple provisions, requiring strict differentiation of similar requirements (e.g., the subtle difference between "lacking the capacity to discern matters of fact" and being "significantly impaired"). Although this type of failure has substantially decreased compared to the Text-RAG baseline, limitations in the LLM's zero-shot reasoning remain and represent a challenge for future work.

Second, the bloating of the context and the introduction of noise caused by graph expansion (e.g., 06-05-U in Table 3). While expansion via the `refer_to` relation is effective for comprehensive retrieval, not all referenced targets are directly necessary for determining the truth value of a given query. Consequently, the retrieved context becomes excessively long, causing the crucial provisions that serve as the basis for the correct answer to be relatively buried. As a result, even though the relevant provisions themselves are provided to the LLM, it fails to appropriately extract them as the basis for reasoning.

Third, retrieval omissions in the initial stage caused by the accuracy of legal concept extraction and orthographical variations. Inadequate extraction of legal concepts from the query occasionally led to the system overlooking provisions necessary for the correct answer during the keyword-based retrieval stage. Additionally, a structural issue was identified during KG construction, wherein

Table 3: Examples of failure cases and their possible causes.

Query ID	Query	Gold	Pred.	Possible cause
R06-01-A	When a petition is filed to establish guardianship for a person who, due to a mental disability, is in a state where they lack the capacity to discern matters of fact, the Family Court may issue an order establishing guardianship.	N	Y	The LLM failed to distinguish the subtle legal distinction between a person who "lacks the capacity to discern matters of fact" and one whose capacity is "significantly impaired."
R06-05-U	If a contract is entered into between A and B stating, "If B marries C, A shall transfer Plot A, which A owns, to B," then B may not transfer B's rights under the contract between A and B to a third party while the fulfillment of that condition remains undetermined.	N	Y	Relevant provisions were correctly retrieved from the KG, but the excessive length of the provided context prevented the LLM from identifying the correct basis for reasoning.

substantially identical legal concepts extracted from different provisions were registered as distinct entities due to slight orthographical variations. Future prospects for addressing these issues include introducing a process to normalize and aggregate synonyms by clustering the extracted entity groups, as well as optimizing the weighting in the ensemble with vector retrieval. These enhancements are expected to improve retrieval comprehensiveness and further elevate the overall accuracy of the system.

6 Conclusion

In this study, we proposed a multi-stage retrieval and reasoning approach utilizing a legal-domain-specific knowledge graph for the statute law-based truth-value prediction task (SL-QA) at COLIEE 2026. To address the complex dependencies unique to legal documents, we constructed a KG that explicitly represents the citation relationships (`refer_to`) between provisions, in addition to the conceptual connections among provisions, headings, and keywords. By combining graph expansion retrieval leveraging this structural information with LLM-based reasoning, our approach achieved performance superior to conventional text-based RAG and general-purpose KG-RAG methods, thereby demonstrating the critical importance of structural context in the legal domain.

For future work, the primary challenge lies in mitigating the noise introduced by graph expansion. To address this, dynamic relevance filtering must be introduced to restrict the expansion scope to provisions strictly necessary for reasoning. Additionally, further enhancements require optimizing the LLM's capability to handle complex legal conditions through advanced prompting, alongside refining the initial retrieval stage via legal concept clustering.

References

[1] Farid Ariai, Joel Mackenzie, and Gianluca Demartini. 2025. Natural Language Processing for the Legal Domain: A Survey of Tasks, Datasets, Models and Challenges. *Comput. Surveys* (2025). arXiv:2410.21306.

[2] Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, Dasha Metropolitansky, Robert Osazuwa Ness, and Jonathan Larson. 2024. From Local to Global: A GraphRAG Approach to Query-Focused Summarization. *arXiv:2404.16130* (2024).

[3] Randy Goebel, Yoshinobu Kano, Calum Kwan, Mi-Young Kim, Juliano Rabelo, Ken Satoh, Hiroaki Yamada, and Masaharu Yoshioka (Eds.). 2025. *Proceedings of the Workshop on the Twelfth International Competition on Legal Information Extraction and Entailment.*

[4] Haiyan Guo, Ruihong Qiu, Jianing Wang, et al. 2024. LightRAG: Simple and Fast Retrieval-Augmented Generation. *arXiv:2410.05779* (2024).

[5] Yoshinobu Kano, Randy Goebel, Mi-Young Kim, Calum Kwan, Ken Satoh, Hiroaki Yamada, and Masaharu Yoshioka. 2026. An Overview of the COLIEE 2026 Competition: Legal Case Law and Statute Law Information Retrieval and Entailment. In *Proceedings of the 21st International Conference on Artificial Intelligence and Law (ICAIL 2026).*

[6] Soha Khazaeli, Janardhana Punuru, Chad Morris, Sanjay Sharma, Bert Staub, Michael Cole, Sunny Chiu-Webster, and Dhruv Sakalley. 2021. A Free Format Legal Question Answering System. In *Proceedings of the Natural Legal Language Processing Workshop.* 107–113.

[7] Yanling Li, Jiaye Wu, and Xudong Luo. 2024. BERT-CNN based evidence retrieval and aggregation for Chinese legal multi-choice question answering. *Neural Computing and Applications* 36 (2024), 5909–5925.

[8] Antoine Louis, Gijs van Dijck, and Gerasimos Spanakis. 2024. Interpretable Long-Form Legal Question Answering with Retrieval-Augmented Large Language Models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38. 22266–22275.

[9] Takao Mizuno and Yoshinobu Kano. 2025. Hierarchical and Referential Structure-Aware Retrieval for Statutory Articles using Graph Neural Networks. In *Proceedings of the Workshop on the Twelfth International Competition on Legal Information Extraction and Entailment.*

[10] Dat Nguyen, Minh Phuong Nguyen, Quang Huy Chu, Thanh Son Luu, Quang Nguyen Hoang Chu, Thien Trung Vo, and Le Minh Nguyen. 2025. CAPTAIN at COLIEE 2025: Enhancing Legal Text Processing and Structural Analysis with Large Language Models. In *Proceedings of the Workshop on the Twelfth International Competition on Legal Information Extraction and Entailment.*

[11] The Hai Nguyen, Khac Vu Hiep Nguyen, Ngoc Anh Trang Pham, Ngoc Minh Nguyen, Hoang An Trieu, Nguyen Khang Le, Dinh Truong Do, and Le Minh Nguyen. 2025. JNLP at COLIEE 2025: Hybrid Large Language Model-based Framework for Legal Information Retrieval and Entailment. In *Proceedings of the Workshop on the Twelfth International Competition on Legal Information Extraction and Entailment.*

[12] Juliano Rabelo, Randy Goebel, Mi-Young Kim, Yoshinobu Kano, Masaharu Yoshioka, and Ken Satoh. 2024. Overview and Discussion of the Competition on Legal Information Extraction/Entailment (COLIEE) 2023. *The Review of Socionetwork Strategies* 18, 1 (2024), 27–47.

[13] Stephen E. Robertson and Hugo Zaragoza. 2009. The Probabilistic Relevance Framework: BM25 and Beyond. *Foundations and Trends in Information Retrieval* 3, 4 (2009), 333–389.

[14] Francesco Sovrano, Monica Palmirani, Salvatore Sapienza, et al. 2025. DiscoLQA: zero-shot discourse-based legal question answering on European Legislation. *Artificial Intelligence and Law* 33 (2025), 323–359.

[15] Sabine Wehnert, Bhavya Baburaj Chovatta Valappil, and Ernesto William De Luca. 2025. LLMS, Knowledge Graphs, and Hybrid Search: Task-Specific Approaches to Legal AI in COLIEE. In *Proceedings of the Workshop on the Twelfth International Competition on Legal Information Extraction and Entailment.*

A Keyword Extraction Prompt

The following prompt template was used to extract legal keywords from statutory provisions for knowledge graph construction. For brevity, we present a simplified version of the actual prompt.

> You are an expert in legal knowledge graph construction.

Given a Japanese statutory provision, extract 3–10 important legal keywords that can function as nodes in a legal knowledge graph.
Prioritize legally salient concepts over overly generic terms.
Preferred categories include:

- Rights / Duties
- Persons
- Legal Effects / Outcomes
- Legal Actions
- Legal States / Situations
- Legal Concepts

Additional constraints:

- Prefer concise legal expressions appearing in the original text.
- Avoid references to specific article numbers or expressions such as “the preceding article” or “this Act.”
- Return JSON format only.

Output Format:

```
{
  "article_id": "...",
  "keywords": [
    {
      "name": "...",
      "role": "..."
    }
  ]
}
```

Input:

```
Article ID: ...
Provision Text: ...
```

HUKB@COLIEE 2026: A Reasoning-Guided Framework for Statute Retrieval and Legal Entailment

Gota Sakakibara
Faculty of Information Science and Technology,
Hokkaido University
Hokkaido, Japan
sakakibara.gota.y6@elms.hokudai.ac.jp

Masaharu Yoshioka
Faculty of Information Science and Technology,
Hokkaido University
Hokkaido, Japan
yoshioka@ist.hokudai.ac.jp

Abstract

Legal entailment based on statutory articles constitutes an important challenge in legal information systems. The COLIEE statute law tasks (Task 3 and 4) are designed to address this problem using Japanese Bar Examination data and civil law articles. This paper presents our approach to Task 3, which adopts a multi-stage framework that integrates retrieval and reasoning, rather than treating them as independent components.

The proposed framework first lets a large language model (LLM) attempt to solve the question directly and uses the generated reasoning output for the query to retrieve corresponding articles in the database. The retrieved candidates are then merged with articles obtained from the original query, expanded through *mutatis mutandis* relations, and refined by an explicit article selection step before the final entailment decision is made. One practical advantage of this design is that it improves retrieval quality without relying on embedding-based retrieval or additional task-specific tuning.

Official COLIEE 2026 results show that our Task 3 runs achieved strong retrieval performance, with the best run attaining the highest F_2 among the representative runs reported in this paper. At the same time, the official Task 3 accuracy and Task 4 results indicate that strong retrieval quality alone is not sufficient for the best end-to-end performance, and that the entailment stage remains a major source of error. Ablation studies on the 2024 and 2025 datasets further support this interpretation: reasoning-guided retrieval and *mutatis mutandis* expansion mainly improve recall and gold-article coverage, whereas article selection improves precision, F_2, and the quality of the final evidence set. The implementation is available at https://github.com/kawauso24/coliee2026.

CCS Concepts

• **Applied computing** → **Law**; • **Information systems** → **Question answering**; • **Computing methodologies** → *Natural language processing*.

COLIEE 2026, Singapore

Keywords

COLIEE competition, Legal Information Processing, Document Retrieval, Textual Entailment, Large Language Models,

ACM Reference Format:
Gota Sakakibara and Masaharu Yoshioka. 2026. HUKB@COLIEE 2026: A Reasoning-Guided Framework for Statute Retrieval and Legal Entailment. In *Proceedings of COLIEE 2026 workshop, June 12, 2026, Singapore*. ACM, New York, NY, USA, 10 pages.

1 Introduction

The Competition on Legal Information Extraction/Entailment (COLIEE) [18] provides a shared benchmark for legal information retrieval, legal entailment, and related reasoning tasks. Among its subtasks, COLIEE 2026 Task 3 [12] addresses statute retrieval and entailment for Yes/No legal questions, requiring systems to return both relevant statutory articles and an entailment decision.

A common approach to legal information retrieval is to rank candidate provisions according to lexical or semantic similarity between the query and the article text. Typical lexical approaches rely on sparse retrieval methods such as BM25 [19], whereas semantic approaches often rely on dense retrieval models [13] to capture contextual similarity [9, 16]. While effective in many cases, these approaches do not fully capture the nature of legal problem-solving. Articles that are necessary for resolving a legal question are not always those that are most similar to the wording of the query. A legally important provision may be expressed indirectly, may be connected through a referenced or applicable article, or may become necessary only after the legal issue is properly interpreted.

This suggests that, especially in integrated legal question answering, retrieval should not be viewed merely as an independent ranking step that precedes reasoning [10]. Rather, statute retrieval should be guided not only by similarity to the query, but also by the role that candidate articles play in the overall problem-solving process.

Motivated by this perspective, we propose a framework that incorporates end-to-end problem-solving into statute retrieval. Instead of relying only on the original query, we first let a large language model (LLM) [4, 21] attempt to solve the legal question directly. Although such outputs often contain hallucinated or legally inaccurate content [3, 11], they may still provide useful retrieval cues in the form of terms, concepts, and issue formulations [8, 22]. We retrieve statutory articles from these generated cues using BM25 and combine them with articles retrieved from the original query. We further expand the candidate set by adding articles connected through *mutatis mutandis* (Junyo) relations, and then perform reasoning conditioned on the candidate articles. To reduce noise from

irrelevant retrieved articles [20], the model is required to explicitly select the articles it actually relied on, inspired by prior work on evidence selection [25]. The final entailment decision is then made using only those selected articles.

We evaluate the proposed framework on COLIEE 2026 Task 3, analyze each module by ablation, and report Task 4 results to better understand the entailment component.

2 Related Work

2.1 Legal Information Retrieval for Statute Retrieval

Legal information retrieval has commonly been formulated as a ranking problem based on lexical or semantic similarity between a query and candidate legal texts. Lexical approaches are typically implemented with sparse retrieval methods such as BM25 [19], while semantic approaches often rely on dense retrieval models [13] trained to capture contextual similarity.

In legal retrieval, both classes of methods have been actively studied, and competition settings such as COLIEE have further encouraged the development of retrieval systems for statute and case law tasks [9, 16]. These approaches have shown that both sparse and dense retrieval can be effective depending on the legal domain and the structure of the underlying texts. From an operational perspective, sparse retrieval methods are also attractive in statutory domains, since updates to the target corpus can be reflected directly through re-indexing, whereas dense models may require additional adaptation or retraining to stay aligned with revised legal texts.

However, existing retrieval methods generally treat retrieval as a similarity-based ranking problem, where the main objective is to retrieve texts that resemble the query at the lexical or semantic level. In contrast, our work focuses on retrieving articles that are actually useful for solving the legal question, even when they are not the most similar ones on the surface.

2.2 LLMs for Legal Reasoning

Large language models [4, 21] have recently been applied to a wide range of legal tasks, including legal question answering, legal reasoning, and support for legal research [3]. Their generative ability makes them attractive for end-to-end problem-solving, especially in settings where the system is expected to interpret a question and produce a final answer directly.

However, the legal domain also makes the limitations of LLMs visible. Prior work has shown that legal hallucinations are frequent and can seriously undermine the reliability of model outputs [5, 11]. This issue is especially critical when the generated answer is treated as legal authority or as a substitute for grounded legal reasoning.

Rather than using the initial LLM output as the final answer, our work uses it as an intermediate signal for grounded retrieval and subsequent reasoning. This allows the framework to exploit potentially useful legal concepts and issue formulations while avoiding direct reliance on ungrounded generated answers.

2.3 Retrieval-Augmented Legal Question Answering

Recent research has increasingly studied legal question answering in settings where retrieval and reasoning are combined. Retrieval-augmented legal QA frameworks use retrieved legal provisions as evidence for answer generation [15], and recent benchmark work has argued that legal retrieval should be evaluated in the context of realistic downstream legal research tasks rather than as an isolated ranking problem [26]. These studies suggest that legal retrieval should be assessed with respect to its contribution to legal problem-solving.

Many existing pipelines follow a retrieve-then-reason structure, where retrieval is performed first and reasoning is applied to the resulting candidates. Our framework is related to this line of work, while using an initial reasoning attempt as an intermediate signal for subsequent retrieval.

3 Task Description

3.1 COLIEE Task3

COLIEE 2026 Task3 [12] addresses statute law retrieval and entailment for Yes/No questions. Given a legal query and an appropriate subset of articles from the Japanese Civil Code, a system is required to return an entailment decision together with the relevant statutory articles. An article is considered relevant if the query can be answered with *Yes* or *No* and the answer is entailed by the meaning of that article.

In this task, the training data consist of legal queries, sets of relevant articles, and entailment labels, while the test data contain only the queries. Therefore, a participating system must automatically retrieve supporting articles from the statute collection and determine whether the query is entailed. Following the task setting, no human intervention is allowed during test-time processing.

3.2 COLIEE Task4

COLIEE 2026 Task 4 [12] is a legal textual entailment task. Given a legal Yes/No question together with a set of statutory articles, a system is required to determine whether the question is entailed by the provided articles. The output of the task is a binary entailment decision for each query.

3.3 Evaluation Metrics

For Task 3, we report precision, recall, F_2, and accuracy [18]. Let G_q and R_q denote the sets of gold and retrieved relevant articles for query q, respectively. The precision and recall for each query are defined as

$$P_q = \frac{|G_q \cap R_q|}{|R_q|}, \qquad R_q = \frac{|G_q \cap R_q|}{|G_q|}.$$

Following the official evaluation setting, Task 3 precision and recall are computed as macro-averages over all queries:

$$\text{Precision} = \frac{1}{|Q|}\sum_{q\in Q} P_q, \qquad \text{Recall} = \frac{1}{|Q|}\sum_{q\in Q} R_q,$$

where Q is the set of all queries. The F_2-measure is then defined as

$$F_2 = \frac{5 \cdot \text{Precision} \cdot \text{Recall}}{4 \cdot \text{Precision} + \text{Recall}}.$$

For Task 3, the primary official metric is accuracy, which counts a query as correct only when the system predicts the correct entailment result together with a sufficient set of relevant articles, i.e., Recall = 1. Formally, Task 3 accuracy is defined as

$$\text{Accuracy}_{\text{T3}} = \frac{\left|\left\{q \in Q \;\middle|\; \hat{y}_q = y_q \text{ and } R_q = 1\right\}\right|}{|Q|},$$

where y_q and $\hat{y}_q$ denote the gold and predicted entailment labels for query q, respectively.

For Task 4, the evaluation metric is accuracy over binary entailment decisions:

$$\text{Accuracy}_{\text{T4}} = \frac{\left|\left\{q \in Q \;\middle|\; \hat{y}_q = y_q\right\}\right|}{|Q|}.$$

4 Proposed Method

4.1 Overview

Our framework is designed to improve statute retrieval in COLIEE 2026 Task 3 [12] by incorporating end-to-end legal problem-solving into the retrieval process. Instead of relying only on the original query, we first let a large language model (LLM) [4, 21] attempt to solve the legal question directly. The resulting output is not used as the final answer; rather, it is treated as an intermediate signal for identifying legal concepts and statutory grounds that may be useful for retrieval. However, enriching the candidate set in this manner can also introduce irrelevant or weakly related articles, which may act as noise and negatively affect the accuracy of downstream entailment prediction [14, 20]. For this reason, the framework later includes an explicit article selection step to reduce such noise before the final entailment decision.

As illustrated in Figure 1, the framework proceeds in five steps. In step (a), the LLM first generates an initial end-to-end reasoning output for the query, and BM25 retrieval is applied to that generated output to obtain articles associated with the reasoning result. In step (b), BM25 retrieval is also applied directly to the original query in order to obtain another set of candidate articles. The articles obtained in steps (a) and (b) are then merged. In step (c), the merged article set is further expanded through *mutatis mutandis* mapping so that applicable referenced articles are added to the candidate set. In step (d), the LLM performs reasoning over the candidate articles and selects the articles actually used in its reasoning. Finally, in step (e), the model makes the entailment decision using only the selected articles.

4.2 Initial End-to-End Reasoning and Retrieval from Reasoning Output

Let q denote an input legal query. Given q, we first prompt the LLM to solve the legal question in an end-to-end manner. The purpose of this step is not to obtain a directly reliable final answer, but to obtain a list of candidate articles for the entailment, since LLM outputs may contain hallucinated or legally inaccurate content.

To identify the original articles corresponding to the generated article texts, we treat each generated article text as a query and retrieve candidate articles from the database. The motivation is that an initial reasoning attempt can uncover terms, legal concepts, issue formulations, and candidate statutory grounds that may be necessary for answering the question, as reflected in the generated article-level text. These texts may differ from surface-level expressions appearing in the query itself, and may therefore help retrieve articles that would be missed by similarity-based retrieval alone.

Let $g(q) = \{c_1, \ldots, c_m\}$ denote the structured output generated by the initial LLM reasoning step, where each c_i is an article-level generated text or statutory ground suggested by the model.

For a query x and an article d, the BM25 score is defined as

$$\text{BM25}(x, d) = \sum_{t \in x} \text{IDF}(t) \cdot \frac{f(t,d)\,(k_1 + 1)}{f(t,d) + k_1 \left(1 - b + b\frac{|d|}{\text{avgdl}}\right)},$$

where

$$\text{IDF}(t) = \log \frac{N - n(t) + 0.5}{n(t) + 0.5}.$$

Here, $f(t,d)$ denotes the frequency of term t in article d, $|d|$ denotes the article length, avgdl is the average article length in the corpus, N is the number of articles in the corpus, and $n(t)$ is the number of articles containing term t. The parameter k_1 determines how strongly repeated occurrences of a term increase the score, whereas b adjusts the extent to which the score is normalized by article length. In our implementation, we used the library default values, $k_1 = 1.5$ and $b = 0.75$.

Using the generated reasoning output $g(q)$ as a retrieval cue, we apply BM25 to each statutory article mentioned in $g(q)$ and obtain the top-k_g retrieved articles for each of them:

$$g(q) = \{c_1, \ldots, c_m\}, \qquad C_g = \bigcup_{c \in G(q)} \text{TopK}_{k_g}\{\text{BM25}(c, d) \mid d \in D\}.$$

Figure 2 shows the user prompts used in this step.

4.3 Retrieval from the Original Query

In parallel with step (a), we also apply BM25 retrieval directly to the original query q. This step is intended to preserve articles that are strongly supported by the surface form of the query itself, independently of the LLM-generated reasoning output. The top-k_q retrieved articles are defined as

$$C_q = \text{TopK}_{k_q}\{\text{BM25}(q, d) \mid d \in D\}.$$

The articles obtained in steps (a) and (b) are then merged by union:

$$C_{\text{base}} = C_q \cup C_g.$$

4.4 Candidate Set Expansion via *Mutatis Mutandis* Mapping

In Japanese statutory interpretation, *mutatis mutandis* application refers to a relation in which the content of one provision is applied, with necessary modifications, to another legal context. In this paper, we refer to articles connected through such applicable references as *mutatis mutandis* articles.

As illustrated in Figure 3, even when an article required for answering a query is not directly retrieved, it may become relevant through a *mutatis mutandis* relation from a retrieved article. This motivates us to explicitly incorporate such referenced articles into the candidate set.

Let C_{base} denote the set of articles obtained by merging the results of BM25 retrieval from the original query and from the

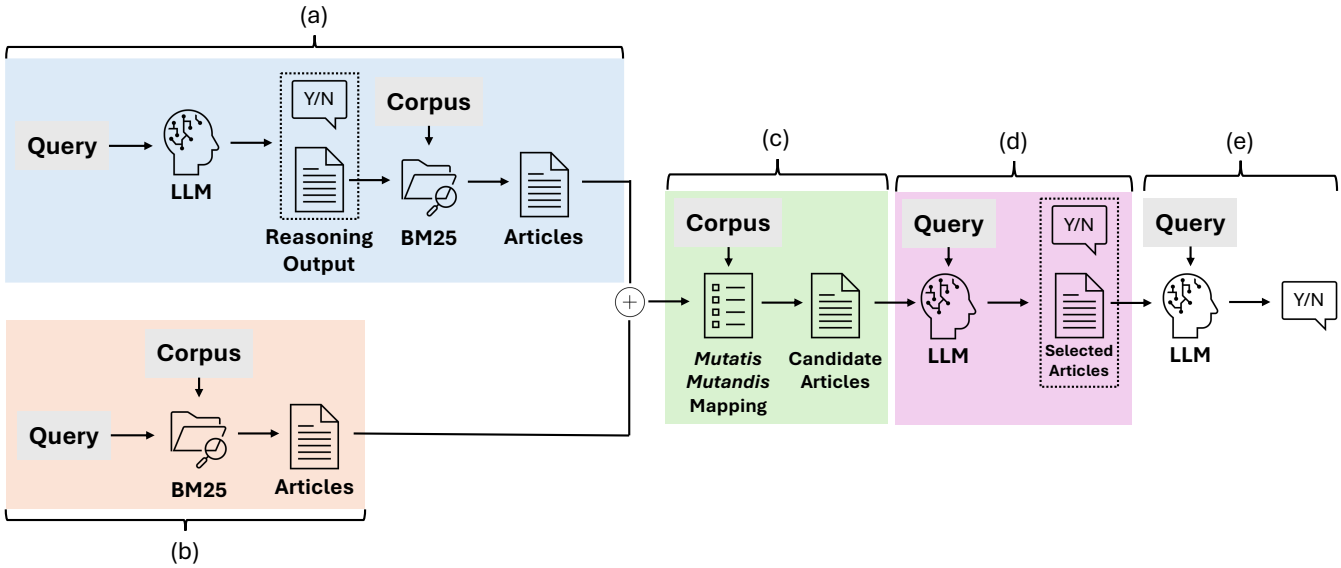


Figure 1: Overview of the framework: (a) initial end-to-end reasoning and retrieval from generated rationales, (b) retrieval from the original query, (c) *mutatis mutandis* expansion, (d) reasoning over candidate articles with article selection, and (e) final entailment decision using the selected articles.

User Prompt for Initial End-to-End Reasoning

Japanese (used in the experiments).

日本の司法試験の民法に関する択一回答式問題です。
記述 (t2) が正しいかどうかを判定し、その回答根拠と
なる民法条文の該当箇所を引用して示してください。
法律に関する記述 (t2)
`{t2_text}`

English translation (for reference).

This question is based on the Japanese Bar Examination and concerns the Civil Code. Determine whether statement (t2) is true or false, and cite the relevant passages of the Civil Code that support the answer.
Statement on a legal issue (t2)
`{t2_text}`

Figure 2: User prompt for the initial end-to-end reasoning step. The experiments used the Japanese prompt; an English translation is shown for reference. The JSON Schema is omitted for brevity.

generated reasoning output. To expand this set, we introduce a mapping function

$$\mathcal{M}(a),$$

which returns the set of articles that are applicable through *mutatis mutandis* relations from article a.

In our implementation, this mapping is constructed manually based on explicit referential descriptions in the statutory text, where each article is associated with the provisions that are applied *mutatis mutandis*.

Using this mapping, the set of referenced articles is defined as

$$C_{\text{ref}} = \bigcup_{a \in C_{\text{base}}} \mathcal{M}(a).$$

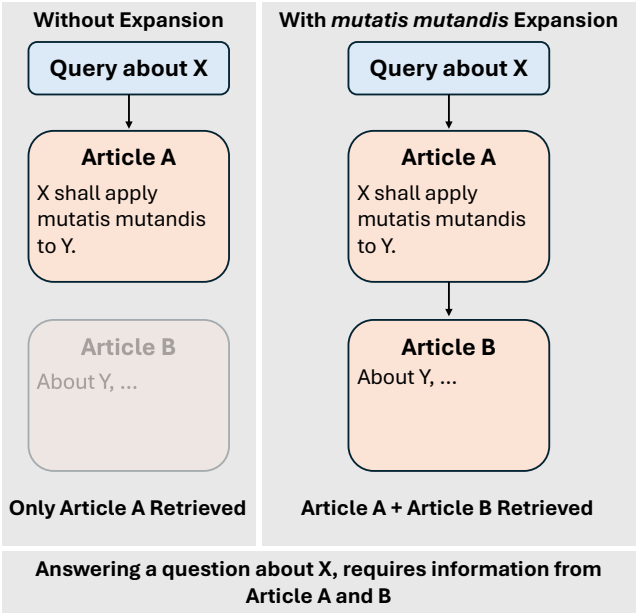


Figure 3: Illustration of *mutatis mutandis* application. Article A is directly retrieved from the query about X, while Article B is not retrieved but becomes relevant through a *mutatis mutandis* relation with Article A.

The final candidate set is then defined as

$$C = C_{\text{base}} \cup C_{\text{ref}}.$$

This expansion enables the model to capture legally necessary provisions that are not explicitly mentioned in the query or the

User Prompt for Article Selection

Japanese (used in the base method).

日本の司法試験の民法に関する択一回答式問題です。与えられた条文情報のみを利用して記述 (t2) が正しいかどうかを判定してください。また実際に判定に利用した条文番号を、与えられた参照条文番号のリストから選んで回答してください。

```
# 参照する条文情報
{retrieved_articles}
# 参照する条文番号リスト
{retrieved_article_ids}
# 法律に関する記述 (t2)
{t2_text}
```

English translation (for reference).

This is a Japanese Bar Examination question on the Civil Code. Using only the provided statutory information, determine whether statement (t2) is correct, and select from the given article list only the article numbers actually used in the judgment.

Figure 4: User prompt for the article selection step, shown with an English translation for reference.

generated reasoning output, but are required through referential structure.

4.5 Reasoning over Candidate Articles and Article Selection

Given the query q and the candidate article set C, we prompt the LLM again to answer the legal question with explicit reference to the candidate articles. At this stage, the model is required not only to produce an entailment-oriented response, but also to identify which articles in C were actually used in its reasoning process.

Formally, this step produces two outputs: (1) an intermediate reasoning result, and (2) a subset of selected articles,

$$C_{\text{sel}} \subseteq C,$$

where C_{sel} contains the articles that the model explicitly cites as supporting its conclusion.

The purpose of this step is to reduce the effect of irrelevant or weakly related articles in the candidate set. By forcing the model to identify the articles actually used, we attempt to retain useful evidence while suppressing noise. Figure 4 shows the user prompts used in this step.

4.6 Final Entailment Decision

In the final stage, we provide the query q together with only the selected articles C_{sel} to the LLM and ask it to make the final entailment decision. Thus, the final prediction is based on a reduced and more focused evidence set rather than on the full candidate set C.

Let $\hat{y}$ denote the final entailment prediction. Then the final decision is computed as

$$\hat{y} = f(q, C_{\text{sel}}),$$

where f denotes the LLM-based entailment prediction conditioned on the query and the selected articles.

Table 1: Models used in our experiments

	llama4:128x17b	**llm-jp-3.1-8x13b**
Parameters (Active)	401.6B (17B)	73.2B (22.2B)
Quantization	Q4_K_M	Q4_K_M
Release Date	2025.04.05	2025.05.31

This design preserves the recall-oriented benefit of the earlier expansion stages while restricting the final reasoning context to a focused evidence set.

5 Experiments

5.1 Experimental Settings

We evaluated the proposed framework on COLIEE 2026 Task 3 and Task 4 [12]. Task 3 is the main target of this study because it requires both retrieval of relevant statutory articles and final entailment prediction. Task 4 is used as a complementary setting to examine the reasoning component when the relevant articles are given.

We followed the official evaluation setting of COLIEE. For Task 3, we report precision, recall, F_2, and accuracy, and for Task 4, we report accuracy. The definitions of these metrics are given in Section 3.

For lexical retrieval, we used BM25 [19] on the statute corpus. BM25 was applied both to the original query and to the LLM-generated initial reasoning output. For LLM-based inference, we used Llama 4 (`llama4:128x17b` [2, 17]) as the main reasoning model and a Japanese-oriented llm-jp model (`llm-jp-3.1-8x13b-instruct4` [6, 7]) in auxiliary settings. Specifically, in Task 3, the llm-jp model was used in the initial reasoning stage to examine whether a Japanese-oriented model can improve retrieval cue generation, while in Task 4, it was used together with Llama 4 in the adaptive consistency [1] setting. Table 1 summarizes the models used in our experiments. We explicitly list the release or public availability information because the COLIEE 2026 regulations require that LLMs used for Task 3 and Task 4 must have been released before July 15, 2025. Llama 4 was publicly announced by Meta on April 5, 2025. For the Japanese-oriented model, we used the publicly available GGUF checkpoint `mmnga/llm-jp-3.1-8x13b-instruct4-gguf`, which is based on `llm-jp/llm-jp-3.1-8x13b-instruct4`. Both the base model and the GGUF checkpoint used in our experiments were publicly available by May 31, 2025.

All experiments were conducted locally with models served through Ollama. Because the Japanese-oriented model was not available in the official Ollama library, we used the Hugging Face GGUF-converted checkpoint described above. Unless otherwise stated, all runs used the same execution environment and pipeline structure.

The experiments consist of three parts: official COLIEE 2026 submissions for Task 3 and Task 4, retrieval hyperparameter selection, and component-wise ablation analysis on previous COLIEE datasets (Section 6).

5.2 Retrieval Hyperparameter Selection

For article retrieval, we used BM25 for both retrieval from the original query and retrieval from the LLM-generated initial reasoning

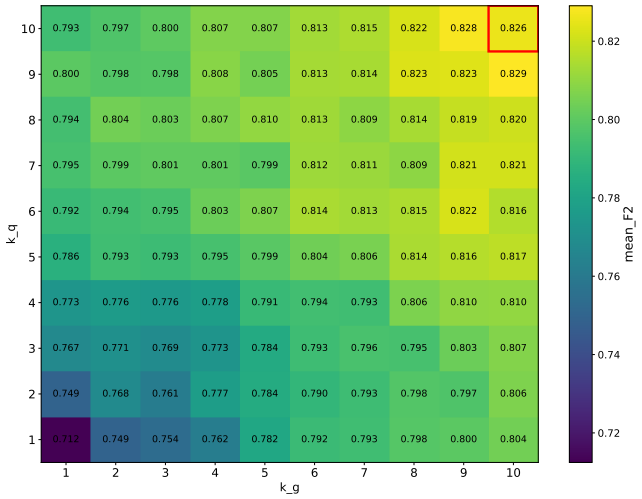


Figure 5: Mean F_2 over the previous five COLIEE datasets for each (k_q, k_g) pair. The selected configuration $(10, 10)$ is highlighted.

output. The numbers of retrieved articles from these two sources were treated as hyperparameters, denoted by k_q and k_g, respectively. To determine their values, we performed a grid search over $k_q \in \{1, \ldots, 10\}$ and $k_g \in \{1, \ldots, 10\}$.

For each parameter pair (k_q, k_g), we evaluated F_2 on the COLIEE datasets from 2021 to 2025. Inspired by prior work on robust hyperparameter selection under temporal distribution shifts, which considers both average performance and worst-case performance across chronological validation sets [24], we adopted a two-stage selection strategy. We first ranked all parameter pairs by their mean F_2 over these five years and retained the top 10 candidates. Among these top candidates, we then compared the worst F_2 across the five years and selected the pair with the best worst-case performance. Following this procedure, we adopted $k_q = 10$ and $k_g = 10$ for the main Task 3 experiments and the ablation analyses reported in Section 6.

Figure 5 shows the mean F_2 obtained for each (k_q, k_g) pair over the previous five COLIEE datasets. The high-scoring region is concentrated near larger values of both k_q and k_g, suggesting that the two retrieval sources provide complementary information and that broad candidate generation is beneficial before the later filtering stage.

5.3 Adaptive Consistency Setting for Task 4

For Task 4 **HUKBac**, we adopted an adaptive consistency setting based on self-consistency and adaptive consistency [1, 23]. The method repeatedly samples entailment predictions from Llama 4 and the llm-jp model and adaptively increases the number of samples when the predictions are unstable or the two models disagree. The final label is determined by model agreement or by the prediction of the model with higher empirical confidence.

Table 2 summarizes the parameters used in this setting. If no decision was resolved at the maximum sample size, we selected the label of the model with higher confidence; ties were resolved in favor of Llama 4, which served as the main reasoning model in our experiments.

Table 2: Parameters used for adaptive consistency in Task 4.

Parameter	Value	Description
Temperature	0.7	Sampling temperature
top-p	0.9	Nucleus sampling threshold
K	10	Initial number of samples
$K_{\max}$	40	Maximum number of samples
growth factor	2	Sample size multiplier
high_conf	0.8	Confidence threshold for agreement
super_conf	0.9	Confidence threshold for one-sided decision
margin_low	0.1	Threshold for unstable predictions
p_gap_high	0.4	Threshold for model disagreement

5.4 Submission Results on COLIEE 2026

To evaluate the proposed framework in the official shared-task setting, we submitted three runs for Task 3 and three runs for Task 4.

For Task 3, the three submitted runs were as follows:

- **HUKBbase**: the full proposed framework using Llama 4 in all LLM-based stages;
- **HUKBmix**: a hybrid setting in which the initial end-to-end reasoning stage used an llm-jp model, while the downstream reasoning and final entailment stages used Llama 4;
- **HUKBsoft**: a variant of the proposed framework in which the article selection stage was relaxed so that the model was allowed to select articles broadly related to the entailment decision, rather than restricting selection to only the articles explicitly used in the final conclusion.

These runs were designed to examine two aspects of the proposed pipeline: the model used in the initial reasoning stage and the strictness of article selection.

Table 3 reports the official Task 3 results together with several representative shared-task runs for reference. Among our submissions, **HUKBmix** achieved the highest precision, recall, and F_2, and **HUKBmix** and **HUKBsoft** achieved the highest Task 3 accuracy among our runs. The comparison between **HUKBbase** and **HUKBmix** suggests that the model used in the initial reasoning stage can affect the quality of reasoning-derived retrieval cues, since replacing Llama 4 with an llm-jp model in that stage improved all four Task 3 metrics. By contrast, **HUKBsoft** achieved the same official accuracy as **HUKBmix** despite lower precision and F_2, which suggests that a broader evidence set may remain acceptable under the Task 3 evaluation criterion as long as it preserves the articles needed for the final judgment.

At the same time, the official results indicate that high recall alone is not sufficient to ensure high Task 3 accuracy. Among the reference runs in Table 3, some runs achieved very high recall but relatively low precision, and their Task 3 accuracy did not necessarily exceed that of runs with a more balanced retrieval profile. One possible interpretation is that, even when many relevant articles are retrieved, including too many irrelevant articles may make the final entailment step more difficult. This interpretation is consistent with prior observations that irrelevant evidence can act as noise in downstream reasoning.

Table 3: Results on COLIEE 2026 Task 3.

Run	Precision	Recall	F_2	Accuracy
HUKBbase	0.9248	0.9431	0.9304	0.8049
HUKBmix	0.9431	0.9553	0.9446	0.8171
HUKBsoft	0.8465	0.9492	0.9076	0.8171
NOWJ_run1	0.0549	1.0000	0.2224	0.9512
JNLP1	0.5151	1.0000	0.7796	0.9024
codyrag	0.0439	1.0000	0.1848	0.8537
BengoR2E2	0.8364	0.9797	0.9242	0.8293
DU2	0.0874	0.9654	0.3168	0.7927

Our results also suggest that retrieval quality itself was already relatively strong. In particular, **HUKBmix** achieved the highest F_2 score among the runs shown in Table 3, which indicates that the proposed framework was effective at retrieving answer-relevant articles with a good precision–recall balance. From this perspective, the remaining room for improvement in Task 3 appears to lie more in the final entailment stage and in the interaction between retrieved evidence and downstream reasoning than in retrieval quality alone. This is consistent with our pipeline design.

For Task 4, we submitted three additional runs to examine how well legal entailment can be solved when the relevant articles are already given. All Task 4 runs used only the gold Civil Code article(s) provided by the task and the statement (t_2), without fine-tuning, in-context demonstrations, or external resources beyond the COLIEE corpus.

The three Task 4 runs were as follows:

- **HUKBllama4**: prompt-based entailment prediction using Llama 4 as the reasoning model;
- **HUKBllmjp**: prompt-based entailment prediction using the llm-jp model as the reasoning model;
- **HUKBac**: adaptive consistency combining Llama 4 and the llm-jp model.

Table 4 shows the official Task 4 results. Among our submissions, **HUKBllama4** and **HUKBac** achieved the same accuracy, whereas **HUKBllmjp** performed substantially worse. This suggests that, under gold-article conditions in our setting, Llama 4 provided a more stable basis for entailment prediction than the llm-jp model. At the same time, adaptive consistency did not improve over Llama 4 alone, which indicates that simple aggregation across the two models was not sufficient to produce an additional gain.

Taken together, the Task 3 and Task 4 results suggest that end-to-end performance in COLIEE Task 3 depends not only on retrieving relevant articles, but also on controlling the quality of the evidence set presented to the final entailment model. In our results, retrieval quality was already relatively strong from the viewpoint of Task 3 F_2, whereas the remaining gap in official accuracy suggests that there is still room for improvement in the final entailment stage and in the interaction between retrieved evidence and downstream reasoning.

6 Ablation Study

To analyze the contribution of each module in the proposed pipeline, we conduct a cumulative ablation study over four stages: *Original query*, *+ Initial reasoning retrieval*, *+ Mutatis Mutandis addition*, and *+ Article selection*. We report precision (P), recall (R), F_2, official Task 3 accuracy (T3), and entailment accuracy (Ent) on the 2024 and 2025 datasets and their average. In addition, we report incremental gold-article coverage (ΔCov.) to measure how many new gold articles are introduced by each expansion step.

Table 4: Results on COLIEE 2026 Task 4.

Run	Accuracy
HUKBllama4	0.8780
HUKBllmjp	0.7805
HUKBac	0.8780
DU1	0.9634
JNLP-3	0.9512
IAIrun2	0.9512
RUG-1	0.9268
UA2	0.9268

Table 5 summarizes the main ablation results, and Table 6 shows the incremental coverage added by *Initial reasoning retrieval* and *Mutatis Mutandis* addition. Overall, the results indicate a clear division of roles among the modules: the expansion stages mainly improve recall and article coverage, whereas the article selection stage substantially improves precision and the quality of the final evidence set.

Using only the original query yields relatively high recall but very low precision, as shown in Table 5. On the averaged results over the two years, the *Original query* stage achieves $R = 0.832$ and $T3 = 0.590$, while precision remains as low as $P = 0.098$ and $F_2 = 0.333$. This indicates that original-query retrieval can cover many relevant articles, but the retrieved set is still highly noisy. The same tendency appears in both yearly subsets: the 2025 set shows slightly higher recall, whereas the 2024 set yields slightly better entailment accuracy, suggesting that the baseline retrieval alone is unstable as a complete solution across years.

6.1 Effect of Initial Reasoning Retrieval

As shown in Table 5, adding *Initial reasoning retrieval* improves coverage-oriented metrics. Compared with the *Original query* stage, the averaged recall rises from 0.832 to 0.883, and T3 increases from 0.590 to 0.655. At the same time, Table 6 shows that this stage introduces additional gold articles, yielding an averaged incremental coverage of ΔCov. = 0.052.

These results suggest that the reasoning step is effective as a query expansion mechanism. Although precision decreases from 0.098 to 0.068, the gains in recall, T3, and incremental coverage indicate that reasoning-derived retrieval cues successfully broaden the candidate set toward answer-relevant statutory provisions. This tendency is also observed in both yearly subsets: the averaged improvement is not driven by only one dataset, but appears consistently across the 2024 and 2025 evaluations. In particular, the stronger T3 gain on the 2024 set (0.578 → 0.679) suggests that the reasoning-derived cues are especially useful when the original query alone is insufficient to expose the legally relevant issue structure.

Stage	2024					2025					Avg.				
	P	R	F_2	T3	Ent	P	R	F_2	T3	Ent	P	R	F_2	T3	Ent
Original query	0.094	0.803	0.319	0.578	0.725	0.103	0.861	0.348	0.603	0.699	0.098	0.832	0.333	0.590	0.712
+ Initial reasoning retrieval	0.065	0.867	0.250	0.679	0.780	0.070	0.900	0.268	0.630	0.740	0.068	0.883	0.259	0.655	0.760
+ *Mutatis Mutandis* addition	0.059	0.881	0.232	0.642	0.725	0.062	0.927	0.245	0.699	0.781	0.061	0.904	0.239	0.670	0.753
+ Article selection	0.735	0.798	0.785	0.587	0.743	0.879	0.874	0.875	0.699	0.849	0.807	0.836	0.830	0.643	0.796

Table 5: Main ablation results on the 2024 and 2025 datasets and their average. P and R denote precision and recall, T3 official Task 3 accuracy, and Ent entailment accuracy. The expansion stages mainly improve recall, whereas article selection improves precision and F_2.

Stage	2024	2025	Avg.
Original query	–	–	–
+ Initial reasoning retrieval	0.064	0.039	0.052
+ *Mutatis Mutandis* addition	0.014	0.027	0.021
+ Article selection	–	–	–

Table 6: Incremental gold-article coverage (ΔCov.) at each expansion stage. ΔCov. denotes the proportion of gold articles newly added relative to the previous stage.

Another important observation is that the gain from this stage is not limited to recall alone. The increase in T3 without any improvement in precision indicates that the additional candidates introduced by reasoning contain more answer-critical gold articles than would be expected from a purely noisy expansion. This supports our central hypothesis that an initial LLM attempt, even if imperfect as a final legal judgment, can still provide useful retrieval cues for downstream statute search.

Takeaway for Initial Reasoning Retrieval

Benefit. Initial reasoning retrieval uses an initial LLM attempt to retrieve answer-relevant provisions that may be missed by lexical or surface-level similarity alone, thereby improving recall, T3 accuracy, and gold-article coverage.
Limitation. The generated reasoning cues can also introduce additional non-gold articles, lowering precision.

6.2 Effect of *Mutatis Mutandis* Addition

Table 5 shows that adding the *Mutatis Mutandis* expansion step mainly improves recall-oriented retrieval performance. On average, recall increases from 0.883 to 0.904, and T3 also increases slightly from 0.655 to 0.670. Table 6 further shows that this step contributes additional gold articles beyond those recovered by the reasoning-based retrieval stage, with an average incremental coverage of ΔCov. = 0.021.

However, the role of this module becomes clearer when the problems are separated according to whether handling *Mutatis Mutandis* relations is actually necessary. As shown in Table 7, the number of such questions is limited: 12 out of 109 questions in the 2024 set and 12 out of 73 questions in the 2025 set. This suggests that the effect of *Mutatis Mutandis* addition may be difficult to interpret from all-question averages alone, because most questions do not require explicit handling of such relations.

On questions that require handling *Mutatis Mutandis* relations, this expansion step increases gold-article coverage in 2 cases in 2024 and 4 cases in 2025. It also improves T3 in 1 case in 2024 and 3 cases in 2025, and improves entailment correctness in 1 case in 2025. Moreover, in 2025, all gold articles newly introduced by this step remain in the final selected evidence set, whereas in 2024, 1 of the 2 newly added gold articles is retained after article selection. These results indicate that *Mutatis Mutandis* expansion can help recover answer-relevant provisions in a subset of questions for which statutory applicability relations are genuinely needed.

At the same time, this module is not high-precision as a retrieval step by itself. Among the articles newly added by *Mutatis Mutandis* expansion for questions that require such relations, only 2 of 24 in 2024 and 4 of 96 in 2025 are gold articles. Furthermore, on questions that do not require *Mutatis Mutandis* handling, the module still adds extra articles in 80 cases in 2024 and 53 cases in 2025, and these additions do not increase gold coverage in any of those cases. This shows that the module frequently introduces non-gold candidates when statutory applicability relations are not actually needed.

Nevertheless, most of this additional noise is removed by the article selection stage. For questions that do not require *Mutatis Mutandis* handling, article selection removes all added non-gold articles in 78 of 80 noisy cases in 2024 and all 53 noisy cases in 2025. The negative effect on the final decision is therefore limited, although some degradation remains in 2024, where 5 such cases become incorrect after *Mutatis Mutandis* addition.

The yearly aggregate results are consistent with this picture. On the 2024 set, recall improves only slightly from 0.867 to 0.881, while T3 decreases from 0.679 to 0.642, with ΔCov. = 0.014. In contrast, on the 2025 set, both recall and T3 improve (0.900 → 0.927 and 0.630 → 0.699), with a larger incremental coverage (0.027). Taken together, these results suggest that *Mutatis Mutandis* addition should be understood primarily as a coverage-oriented but low-precision legal expansion step. Its benefit appears in a limited subset of questions where statutory applicability relations are genuinely relevant, while its broader use requires robust downstream filtering to prevent non-gold expansions from harming end-to-end performance.

Takeaway for *Mutatis Mutandis* Addition

Benefit. *Mutatis mutandis* addition adds statutorily applicable provisions that cannot be recovered by lexical or reasoning-derived retrieval alone.
Limitation. Since such questions are limited, applying this expansion broadly can introduce non-gold articles as noise and may reduce downstream entailment or official Task 3 accuracy.

Measure	2024	2025
Questions requiring MM handling	12	12
Coverage increased on such questions	2	4
T3 improved on such questions	1	3
Ent improved on such questions	0	1
Extra articles added on other questions	80	53
Ent worsened on other questions	5	0
Noise removed by selection	78/80	53/53

Table 7: Subset-based analysis of *Mutatis Mutandis* addition. Questions are manually divided into those that require *Mutatis Mutandis* handling and those that do not. The table summarizes how often the expansion improves coverage or downstream performance on the former subset and introduces noise on the latter.

Measure	2024	2025
Prediction changed	16	7
Incorrect → Correct	9	6
Correct → Incorrect	7	1
Net gain	2	5

Table 8: Changes in entailment correctness after article selection. "Prediction changed" denotes cases where the final entailment label differs before and after article selection. Net gain is computed as Incorrect → Correct minus Correct → Incorrect.

6.3 Effect of Article Selection

The largest change appears at *Article selection*, as shown in Table 5. After selecting only the articles judged to be necessary for the final decision, the averaged precision increases sharply from 0.061 at the *Mutatis Mutandis* stage to 0.807, and F_2 improves from 0.239 to 0.830. Entailment accuracy also rises from 0.753 to 0.796. At the same time, recall decreases from 0.904 to 0.836.

To further examine how article selection affects the final entailment decision, Table 8 compares the predictions before and after article selection. The results are consistent with the improvement in entailment accuracy: article selection corrects more previously incorrect predictions than it breaks previously correct ones, yielding a net gain of 2 cases in 2024 and 5 cases in 2025. At the same time, the presence of Correct → Incorrect cases shows that evidence refinement can also change the final decision in an undesirable direction.

Together with the main ablation results, this indicates that article selection acts as an evidence refinement step: it transforms a low-precision, high-recall candidate set into a compact, high-precision evidence set, while also affecting the final entailment decision in both beneficial and harmful directions.

An important observation is that article selection does not maximize averaged T3: the best averaged T3 score is obtained after *mutatis mutandis* addition rather than after article selection (0.670 vs. 0.643). This is consistent with the strict definition of the official Task 3 metric, which requires not only a correct entailment label but also complete recall of all gold articles. Thus, while article selection substantially improves evidence quality, precision, and F_2, it can also remove some answer-critical gold articles and reduce T3 in some cases.

This trade-off indicates that the selector functions as an evidence refinement mechanism rather than a simple compression step. The large improvement in F_2 (0.239 → 0.830 on average) shows that the selector improves the balance between retrieval completeness and evidence quality, even though it sacrifices some recall. A more robust selection strategy that preserves answer-critical gold articles while retaining this precision-recovery benefit remains an important direction for future work.

Takeaway for Article Selection

Benefit. Article selection substantially improves precision, F_2, and entailment accuracy by refining a broad, noisy candidate set into a focused evidence set.
Limitation. It can remove some gold articles, reducing recall and preventing the best T3 accuracy despite improving overall evidence quality.

7 Conclusion

In this paper, we proposed a multi-stage framework for COLIEE 2026 Task 3 [12] that integrates statute retrieval and legal entailment reasoning. The framework combines retrieval from the original query, reasoning-guided retrieval from an initial LLM response, *mutatis mutandis* expansion, and explicit article selection before final entailment prediction. A practical advantage of this design is that it improves retrieval performance without relying on embedding-based retrieval or additional task-specific tuning.

The official COLIEE 2026 results and our ablation analyses show that the framework is effective as a retrieval-oriented design. In Task 3, our best run achieved a strong precision–recall balance and the highest F_2 among the representative runs reported in this paper. The ablation study further showed a clear division of roles among the modules: initial reasoning retrieval and *mutatis mutandis* addition mainly improve recall and gold-article coverage, whereas article selection substantially improves precision, F_2, and entailment accuracy by refining a broad candidate set into a focused evidence set.

At the same time, the results also show that retrieval-oriented gains do not automatically translate into the strongest end-to-end Task 3 accuracy. The gap between strong Task 3 F_2 and more modest official accuracy, together with the Task 4 results under gold-article conditions, suggests that the remaining bottleneck lies in downstream entailment reasoning and in how the final model uses the retrieved evidence. The ablation results also reveal a trade-off in article selection: it improves evidence quality, but can reduce Task 3 accuracy when answer-critical gold articles are filtered out.

These findings support the view that statute retrieval in integrated legal question answering should be treated as a staged process that balances candidate coverage and evidence refinement. They also indicate that strong retrieval alone is insufficient unless the final entailment model can robustly use the retrieved evidence.

As future work, we plan to improve the reasoning model used for final entailment prediction, reconsider the article selection strategy so that it better preserves answer-critical gold articles, and further analyze the dataset-dependent behavior of the proposed framework.

In particular, the official results and ablation studies show noticeable differences in F_2 across settings and datasets, suggesting that the effectiveness of each module may depend on the characteristics of the target questions and gold-article structure. A more detailed analysis of this interaction between dataset properties and the proposed method will be necessary to clarify when the framework is most effective and when additional adaptation is required.

References

[1] Pranjal Aggarwal, Aman Madaan, Yiming Yang, and Mausam. 2023. Let's Sample Step by Step: Adaptive-Consistency for Efficient Reasoning and Coding with LLMs. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, Houda Bouamor, Juan Pino, and Kalika Bali (Eds.). Association for Computational Linguistics, Singapore, 12375–12396. doi:10.18653/v1/2023.emnlp-main.761

[2] Meta AI. 2025. The Llama 4 herd: The beginning of a new era of natively multimodal AI innovation. https://ai.meta.com/blog/llama-4-multimodal-intelligence/

[3] Andrew Blair-Stanek, Nils Holzenberger, and Benjamin Van Durme. 2023. Can GPT-3 Perform Statutory Reasoning?. In *Proceedings of the Nineteenth International Conference on Artificial Intelligence and Law* (Braga, Portugal) *(ICAIL '23)*. Association for Computing Machinery, New York, NY, USA, 22–31. doi:10.1145/3594536.3595163

[4] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language Models are Few-Shot Learners. In *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (Eds.), Vol. 33. Curran Associates, Inc., 1877–1901. https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf

[5] Matthew Dahl, Varun Magesh, Mirac Suzgun, and Daniel E Ho. 2024. Large Legal Fictions: Profiling Legal Hallucinations in Large Language Models. *Journal of Legal Analysis* 16, 1 (06 2024), 64–93. arXiv:https://academic.oup.com/jla/article-pdf/16/1/64/58336922/laae003.pdf doi:10.1093/jla/laae003

[6] Hugging Face. 2025. llm-jp/llm-jp-3.1-8x13b-instruct4. https://huggingface.co/llm-jp/llm-jp-3.1-8x13b-instruct4

[7] Hugging Face. 2025. mmnga/llm-jp-3.1-8x13b-instruct4-gguf. https://huggingface.co/mmnga/llm-jp-3.1-8x13b-instruct4-gguf

[8] Luyu Gao, Xueguang Ma, Jimmy Lin, and Jamie Callan. 2023. Precise Zero-Shot Dense Retrieval without Relevance Labels. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (Eds.). Association for Computational Linguistics, Toronto, Canada, 1762–1777. doi:10.18653/v1/2023.acl-long.99

[9] Randy Goebel, Yoshinobu Kano, Mi-Young Kim, Calum Kwan, Ken Satoh, Hiroaki Yamada, and Masaharu Yoshioka. 2026. An Overview of the COLIEE 2025 Competition: Legal Case Law and Statute Law Information Retrieval and Entailment. In *Proceedings of the Twentieth International Conference on Artificial Intelligence and Law (ICAIL '25)*. Association for Computing Machinery, New York, NY, USA, 506–515. doi:10.1145/3769126.3785016

[10] Nguyen Hai Long, Thi Hai Yen Vuong, Ha Thanh Nguyen, and Xuan-Hieu Phan. 2023. Joint Learning for Legal Text Retrieval and Textual Entailment: Leveraging the Relationship between Relevancy and Affirmation. In *Proceedings of the Natural Legal Language Processing Workshop 2023*, Daniel Preotiuc-Pietro, Catalina Goanta, Ilias Chalkidis, Leslie Barrett, Gerasimos Spanakis, and Nikolaos Aletras (Eds.). Association for Computational Linguistics, Singapore, 192–201. doi:10.18653/v1/2023.nllp-1.19

[11] Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of Hallucination in Natural Language Generation. *ACM Comput. Surv.* 55, 12, Article 248 (March 2023), 38 pages. doi:10.1145/3571730

[12] Yoshinobu Kano, Randy Goebel, Mi-Young Kim, Calum Kwan, Ken Satoh, Hiroaki Yamada, and Masaharu Yoshioka. 2026. An Overview of the COLIEE 2026 Competition: Legal Case Law and Statute Law Information Retrieval and Entailment. In *Proceedings of the 21st International Conference on Artificial Intelligence and Law (ICAIL 2026)*.

[13] Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense Passage Retrieval for Open-Domain Question Answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu (Eds.). Association for Computational Linguistics, Online, 6769–6781. doi:10.18653/v1/2020.emnlp-main.550

[14] Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024. Lost in the Middle: How Language Models Use Long Contexts. *Transactions of the Association for Computational Linguistics* 12 (2024), 157–173. doi:10.1162/tacl_a_00638

[15] Antoine Louis, Gijs van Dijck, and Gerasimos Spanakis. 2024. Interpretable Long-Form Legal Question Answering with Retrieval-Augmented Large Language Models. *Proceedings of the AAAI Conference on Artificial Intelligence* 38, 20 (Mar. 2024), 22266–22275. doi:10.1609/aaai.v38i20.30232

[16] Larissa Mori, Carlos Sousa de Oliveira, Yuehwern Yih, and Mario Ventresca. 2026. Assessing the performance gap between lexical and semantic models for information retrieval with formulaic legal language. In *Proceedings of the Twentieth International Conference on Artificial Intelligence and Law (ICAIL '25)*. Association for Computing Machinery, New York, NY, USA, 114–128. doi:10.1145/3769126.3769205

[17] Ollama. 2025. llama4. https://ollama.com/library/llama4

[18] Juliano Rabelo, Randy Goebel, Mi-Young Kim, Yoshinobu Kano, Masaharu Yoshioka, and Ken Satoh. 2024. Overview and Discussion of the Competition on Legal Information Extraction/Entailment (COLIEE) 2023. *The Review of Socionetwork Strategies* 18, 1 (2024), 27–47. doi:10.1007/s12626-023-00152-0

[19] Stephen Robertson and Hugo Zaragoza. 2009. *The Probabilistic Relevance Framework: BM25 and Beyond*. Vol. 3. Now Publishers Inc., Hanover, MA, USA. 333–389 pages. doi:10.1561/1500000019

[20] Freda Shi, Xinyun Chen, Kanishka Misra, Nathan Scales, David Dohan, Ed Chi, Nathanael Schärli, and Denny Zhou. 2023. Large language models can be easily distracted by irrelevant context. In *Proceedings of the 40th International Conference on Machine Learning* (Honolulu, Hawaii, USA) *(ICML'23)*. JMLR.org, Article 1291, 18 pages.

[21] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems*, I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.), Vol. 30. Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

[22] Liang Wang, Nan Yang, and Furu Wei. 2023. Query2doc: Query Expansion with Large Language Models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, Houda Bouamor, Juan Pino, and Kalika Bali (Eds.). Association for Computational Linguistics, Singapore, 9414–9423. doi:10.18653/v1/2023.emnlp-main.585

[23] Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-Consistency Improves Chain of Thought Reasoning in Language Models. In *ICLR 2023*. https://doi.org/10.48550/arXiv.2203.11171

[24] Shaokun Zhang, Yiran Wu, Zhonghua Zheng, Qingyun Wu, and Chi Wang. 2024. HyperTime: Hyperparameter Optimization for Combating Temporal Distribution Shifts. In *Proceedings of the 32nd ACM International Conference on Multimedia* (Melbourne VIC, Australia) *(MM '24)*. Association for Computing Machinery, New York, NY, USA, 4610–4619. doi:10.1145/3664647.3681608

[25] Xinping Zhao, Dongfang Li, Yan Zhong, Boren Hu, Yibin Chen, Baotian Hu, and Min Zhang. 2024. SEER: Self-Aligned Evidence Extraction for Retrieval-Augmented Generation. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (Eds.). Association for Computational Linguistics, Miami, Florida, USA, 3027–3041. doi:10.18653/v1/2024.emnlp-main.178

[26] Lucia Zheng, Neel Guha, Javokhir Arifov, Sarah Zhang, Michal Skreta, Christopher D. Manning, Peter Henderson, and Daniel E. Ho. 2025. A Reasoning-Focused Legal Retrieval Benchmark. In *Proceedings of the 2025 Symposium on Computer Science and Law* (Munich, Germany) *(CSLAW '25)*. Association for Computing Machinery, New York, NY, USA, 169–193. doi:10.1145/3709025.3712219

Towards an Agentic Approach for Legal Reasoning Tasks

Cor Steging
c.c.steging@rug.nl
Bernoulli Institute of Mathematics, Computer Science and Artificial Intelligence, University of Groningen
The Netherlands

Ludi van Leeuwen
l.s.van.leeuwen@rug.nl
Bernoulli Institute of Mathematics, Computer Science and Artificial Intelligence, University of Groningen
The Netherlands

İrem Mutlu
i.mutlu@rug.nl
Bernoulli Institute of Mathematics, Computer Science and Artificial Intelligence, University of Groningen
The Netherlands

Tadeusz Zbiegień
tadeusz.zbiegien@doctoral.uj.edu.pl
Department of Legal Theory, Jagiellonian University
Poland

Abstract

Agentic approaches that leverage large language models (LLMs) to deliberate and autonomously execute multi-step actions are gaining attention. In this study, we propose a prototype agentic framework for legal reasoning, grounded in the Beliefs, Desires, and Intentions (BDI) model. Agents can deliberate over actions such as making predictions, analyzing similar cases, performing counterfactual analyses, and clarifying technical terms, using LLMs as their decision-making engine to update beliefs and desires accordingly. We evaluate this framework and compare it to two non-agentic prompting approaches, baseline and few-shot, on two tasks: legal textual entailment and tort prediction, including the acceptance of claims by defendants and plaintiffs. Results show mixed performance: the agent outperforms the baseline in Task 4 but not the 3-shot approach, while in Task 5 it outperforms the 3-shot method but not the baseline. Analysis indicates that smaller models tend to be overconfident, while larger models perform better but are more prone to make mistakes in their output format. Overall, the agentic approach shows promise and warrants further research.

CCS Concepts

• **Computing methodologies** → **Natural language generation**; **Natural language processing**; **Knowledge representation and reasoning**; **Machine learning**; • **Applied computing** → **Law**.

Keywords

Agentic AI, large language models, legal reasoning

ACM Reference Format:
Cor Steging, Ludi van Leeuwen, İrem Mutlu, and Tadeusz Zbiegień. 2026. Towards an Agentic Approach for Legal Reasoning Tasks. In *Proceedings of COLIEE 2026 workshop, June 12, 2025, Singapore.* ACM, New York, NY, USA, 8 pages.

COLIEE 2026, Singapore

Figure 1: The agentic framework allows the agent to take various multi-step actions autonomously.

1 Introduction

Legal reasoning is multifaceted, comprising a range of multi-step processes that differ depending on the task at hand [6]. I saArtificial approaches to legal reasoning are gaining traction [8], but contemporary Large Language Models (LLMs), and even specifically tailored Large Reasoning Models (LRMs), have difficulty generalising legal reasoning across tasks that are inherently multifarious [14]. Novel approaches in AI rely on agentic frameworks, where LLMs or LRMs are used to autonomously plan, reason, and execute multi-step actions toward achieving complex goals [11]. Given the complex nature of legal reasoning, the agentic approach seems like a suitable match. In this paper, we present an initial prototype of an agentic Beliefs, Desires, and Intentions (BDI) framework for legal reasoning, combining modern LLMs with traditional concepts from artificial intelligence and incorporating expert legal knowledge. We explore this framework in the context of the Competition on Legal Information Extraction and Entailment (COLIEE) [9], where we analyse the behavior of the resulting agents and compare their performance against standard LRM baselines. We formally competed in Task 4 as team RUG, achieving the 4th highest score.

2 Background

We evaluate our agentic framework on two tasks from the COLIEE competition: Task 4 and Task 5 [7]. Task 4 focuses on legal textual entailment in bar exam questions, where each instance consists

of a set of statutory articles and a statement, and the objective is to determine whether the statement is legally entailed by the articles. Task 5, in contrast, involves real-world civil cases concerning tort law and is based on the JTD project [16]. Each case includes undisputed facts, as well as claims made by both the defendant and the plaintiff. Systems are required to perform two sub-tasks: (1) predicting whether the court affirms the existence of a tort (Tort Prediction), and (2) identifying which claims of the defendant and plaintiff are accepted by the court (Rationale Extraction).

While both tasks require legal reasoning over the Japanese Civil Code, they differ in structure and complexity. Task 4 is comparatively simpler, as it is based on bar exam questions with a clear ground truth and explicitly provided legislation. In contrast, Task 5 involves real-world cases, which are typically less clear-cut. Moreover, cases in Task 5 contain substantially more information than those in Task 4, increasing the complexity of the reasoning process.

In the most recent iteration of the competition [7], the best-performing approach for Task 4 relied on out-of-the-box instruction-based LLMs combined with few-shot prompting, where similar questions and their answers are included in the prompt. For Task 5, the top-performing method was based on a fine-tuned LLM. Some previous methods have focused on hybrid systems that explicitly incorporate legal knowledge, for example by extracting atomic causal rules from law articles [10], constructing ANGELIC domain models [7, 13], or designing prompts informed by legal expertise [14]. However, most approaches, including the top-performing methods, primarily rely on standard natural language processing techniques and largely overlook the legal reasoning principles that are fundamental to the domain. Treating all questions uniformly may not be optimal, as prior work shows that different approaches perform better on different types of questions [14]. This suggests that a more specialized, question-dependent approach is warranted.

3 Agentic Framework

In this work, we present a prototype agentic framework for legal reasoning. Agents in this framework can autonomously perform multi-step actions to achieve a given goal. The framework is grounded in the established Beliefs, Desires, and Intentions (BDI) model [5], combined with modern LLMs used as their deliberation engine. Inspired by earlier work by Atkinson and Bench-Capon (2005), agents maintain a set of beliefs about the world, a set of desires, and a set of available actions to take. In addition, each agent is equipped with a short-term memory (STM) that stores the five most recent actions taken or messages received. An overview of the agentic decision-making pipeline can be seen in Figure 2.

The belief set of an agent consists of structured belief objects with several attributes. Each belief includes a unique identifier for reference, a textual representation of its content, a type (e.g., fact, claim, evidence, or rule), a source indicating its origin (e.g., plaintiff or court), and a flag specifying whether the belief is mutable. Furthermore, each belief is associated with a confidence score between 0 and 1, reflecting the agent's degree of certainty in that belief. After each action, the belief set is evaluated and updated. Beliefs may be added, modified, or removed, as determined by an LLM call that takes as input the current belief set, the agent's STM, and its desires. Importantly, belief updates are executed symbolically: the LLM output is parsed and translated into discrete operations that modify the belief set.

The desires set are represented as structured objects with four properties: a reference identifier, a textual description of the desire, an explicit condition that defines when the desire is satisfied, and a mutability flag. Similar to the belief set, the set of desires is updated after each action taken by the agent, allowing desires to be added, removed, or modified. This update is performed via an LLM call that takes the current desires, beliefs, and STM as input. The resulting output is then parsed into symbolic operations that modify the set of desires.

The action set contains all of the actions that the agent can perform. Each action has a name, a description, and a reference to the function that it should call. In the deliberation phase of the agent, the LLM is called, and based on the current desires, beliefs, STM, and possible set of actions, it determines which action to perform. The output of the LLM is parsed, and based on this the action is performed. After an action is performed, it is removed from the action set, and the belief set and desires are updated. In this study, the agent possesses four possible actions to take:

- **A1 Relevant case analysis:** The agent retrieves similar cases from the training dataset, analyses their similarities and differences with respect to the current case, and produces a concise summary of the findings. The agent retrieves three relevant cases using BM25 ranking [2] and provides them to the LLM to analyze similarities, differences, and generate a concise summary. If token limits are exceeded, fewer cases are included.
- **A2 Clarification of terms:** The agent consults a dictionary to identify and define difficult or ambiguous legal terms. The agent uses TF-IDF with a threshold to extract legally relevant terms from the corpus, then retrieves their definitions from open-source legal dictionaries, including Black's Law Dictionary (2nd Edition) [4] and the U.S. Courts Glossary [1].
- **A3 Counterfactual analysis:** The agent reasons from the perspective of both possible outcomes (Task 4: yes and no outcome; Task 5: defendant and plaintiff), and summarizes the resulting arguments. This is achieved by prompting the LLM to adopt different stances and summarize the arguments.
- **A4 Prediction:** Based on its beliefs, desires, and the case facts, the agent predicts the answer to the question (Task 4) or the case outcome and claims acceptance (Task 5) using the LLM.

For both Task 4 and Task 5, the agent executes the decision-making process illustrated in Figure 2. The process begins with the initialization of an empty short-term memory (STM) and the incorporation of all provided case information into the agent's belief set. In Task 4, the question under consideration is recorded as an immutable fact. In Task 5, undisputed facts are registered as immutable facts, whereas plaintiff and defendant claims are stored as immutable claims.

The belief set can be further augmented with relevant legislation. For Task 5, Article 709 of the Japanese Civil Code is always included, as it forms the foundation of tort law. The LLM is then prompted, based on the case details, to identify additional potentially relevant

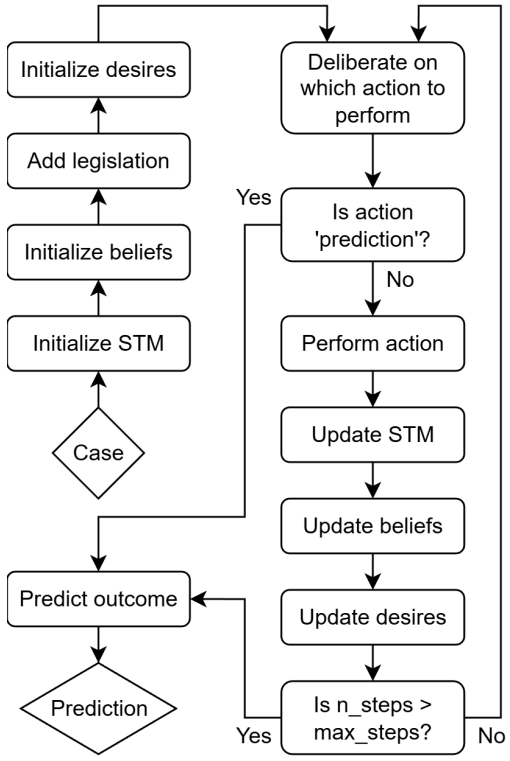


Figure 2: Overview of the decision-making pipeline of the agentic framework

articles from the Japanese Civil Code, which are subsequently incorporated into the belief set. The content and type of the belief, as well as the confidence score are determined by the LLM.

The agent's desires are initialized with an immutable base desire: to either answer the question (Task 4) or determine the outcome of the tort case (Task 5). Subsequently, the agent may generate additional sub-desires informed by its beliefs, which now encompass the details of the question or case. The LLM decides what the content and the conditions of these desires are.

Following initialization, the agent enters the deliberation phase. In this phase, it selects an action from the action set based on its current beliefs, desires, and STM. If the chosen action is to predict the outcome of the question or case, the agent executes this prediction, and the pipeline concludes. Otherwise, the agent performs the selected action and updates its STM, beliefs, and desires by prompting the LLM to determine which elements should be added, removed, or modified. For example, if the agent executes action A2 (clarification of terms), in which case the relevant terms pertaining to the case are gathered and stored in its STM. In the subsequent updates of the beliefs, the agent may choose to store some of the clarified terms into its belief set, revise existing beliefs based on these new definitions, or discard beliefs that were previously held in error. The agent then updates its desires: fulfilled desires may be removed, and new desires may be generated in response to the updated belief set. Note again that the LLM always decides the contents of the beliefs and desires when updating.

After updating the STM, beliefs, and desires, the deliberation phase starts again, unless the agent has exceeded the maximum number of allowed actions (max_{steps} in Figure 2). In our setup, the agent was permitted to perform each action once, with completed actions removed from the action set, effectively setting max_{steps} to 3 (excluding the final prediction step).

Across nearly every stage, including initialization, updates, and deliberation, the LLM serves as the "brain" of the agent and is called upon to decide what the agent should do. This enables creative and autonomous decision making, while keeping the behavior "on rails" within the structure of our framework.

4 Method

We evaluate the proposed agentic approach on both Task 4 and Task 5 of the COLIEE competition. Minor adaptations are made to the agent to account for task-specific differences, primarily through prompt rephrasing (e.g., "predict tort admissibility" versus "predict legal entailment"). Additionally, Task 4 does not permit the use of external relevant legislation, as the task explicitly requires reasoning over the provided statutory articles; consequently, this step is omitted for Task 4.

For Task 4, performance is measured on the three labelled test sets H30, R01, and R02, comprising a total of 258 cases. The label distribution is 48.8% non-entailment and 51.2% entailment. For Task 5, we construct a stratified evaluation set by randomly sampling 258 cases from the training data, enforcing the original label distribution with respect to the court decision (60.4% not affirmed and 39.6% affirmed). The dataset contains 1,073 claims made by the plaintiff and 921 by the defendant, corresponding to 53.8% and 46.2%, respectively. In both tasks, the remaining data is used to support n-shot prompting.

We compare the proposed agent against two baselines. The first is a single-call classifier, which relies on a single LLM invocation using the prompts shown in Figures 3a and 3b. The second is a 3-shot classifier, which retrieves three relevant examples from the training set and incorporates them into the main prompt via the addendum presented in Figure 3c.

For the experiment, we employ a range of large language models (LLMs) as the 'brain' of the agent to assess how model choice affects performance. Our primary model family is Qwen3, which has been demonstrated strong performance on multilingual reasoning tasks [15]. In addition, Qwen3 is open-source and was released prior to July 15, 2025 (JST), satisfying the requirements of the COLIEE competition. We evaluate five variants of Qwen3, ranging from 0.6B to 14B parameters. Smaller models are intentionally used, as agentic approaches require many iterative steps. Furthermore, hardware limitations prevent the use of larger-scale models.

5 Results

The mean accuracy of the four approaches for Task 4 and Task 5, per model used, are shown in Table 1. For Task 5, we show both the performance on the Tort Prediction (TP) task and the Rationale Extraction (RE) task. For the latter, we show the performance of the model across the claims of both the plaintiff and the defendant. We choose to report the F1-score, as this is the metric used by the COLIEE competition for Task 5.

The following question pertains to the Japanese Civil Code. Based on the relevant statutes, please determine whether the following statement is correct.

Relevant statutes: {$articles$}
Statement: {$statement$}

Is the statement correct based on the relevant statutes?
Reason and end your answer with "[Answer: YES] or [Answer: NO]."

(a) Baseline prompt for Task 4 (translated from Japanese).

You are assisting a judge in making the final judgment in a tort case.

The undisputed facts: {$undisputed\_facts$}
Plaintiff claims: {$plaintiff\_claims$}
Defendant claims: {$defendant\_claims$}

DECISION PROCESS:
1. Review the beliefs:
- What facts or legal conclusions are established?
- What remains uncertain?

2. Review unresolved desires:
- What normative objectives remain unsatisfied?
- If critical desires remain unresolved → insufficient grounds for decision
- If all material desires are resolved → sufficient grounds for decision

3. Evaluate the parties' claims:
- For each plaintiff and defendant claim, determine whether it is supported by the belief state.
- Mark the claim as accepted (true) or rejected (false).

4. Make the final tort judgment:
- Based on the preponderance of the established beliefs, determine whether the tort is affirmed.

OUTPUT FORMAT
First output the claim acceptance decisions as JSON with no commentary:

```
{
  "plaintiff": [
    {"id": "P1", "accepted": true},
    {"id": "P2", "accepted": false}
  ],
  "defendant": [
    {"id": "D1", "accepted": false}
  ]
}
```

After the JSON block, output the final judgment on a new line using exactly: [Answer: YES] or [Answer: NO]

(b) Baseline prompt for Task 5.

Below are summaries of relevant past cases. These cases are similar to the current case and may provide insights for reasoning about the current case.
{$examples$}

(c) Additional prompt for n-shot prompting.

Figure 3: Prompts used by the baseline systems for Task 4 (Figure 3a) and Task 5 (Figure 3b). For the 3-shot setting, the prompt shown in Figure 3c is appended.

Based on Table 1, the best-performing model for each task is selected for error analysis. An overview of false positive and false negative rates for these models is presented in Figure 4, covering Task 4 and both the TP and RE components of Task 5. In Figure 4c, errors in the RE task are aggregated across both defendant and plaintiff claims. Figure 5 further breaks down RE errors by error type and party.

To further analyse our agentic framework, we display the average number of beliefs, desires and calls to LLMs the agent has at the end of the case in Table 2. In Table 3, we show how many and which actions are taken by the agents across all the test cases. Note: in every case, the prediction action is always taken and thus excluded from the action type distribution.

6 Discussion

6.1 Discussion on Results

The performance of the approaches on Task 4, shown on the left side of Table 1, indicates that larger models achieve higher accuracy for both the baseline and agent implementations. This suggests that models beyond those tested could enable further improvements, as the performance ceiling does not appear to have been reached. We observe that the agent outperforms the baseline for the largest models (8B and 14B parameters). The 3-shot approach exceeds both baseline and agent methods for all but the smallest model. Performance under 3-shot improves markedly from 0.6B to 4B parameters, followed by a slight decline at larger sizes, likely due to the limited number of runs, as only one run was conducted per combination of test set, model size, and approach.

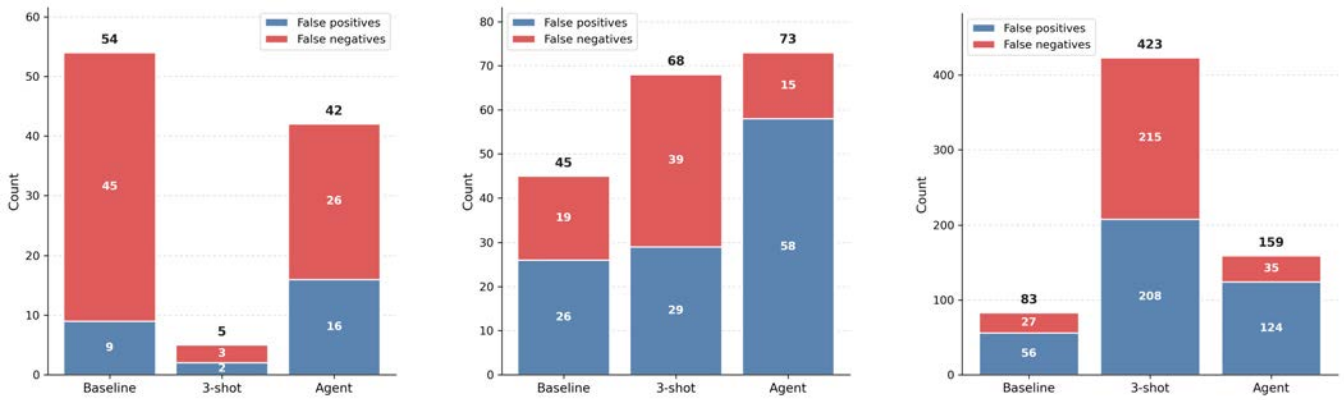
For Task 5, shown on the right side of Table 1, performance on both the Tort Prediction (TP) and Rationale Extraction (RE) tasks generally increases with model size. In the TP task, the largest baseline model achieves the highest performance, followed by the 3-shot model and then the agentic framework. For the RE task, the largest baseline again performs best, with the agent approach as the runner-up. Notably, in the RE task, the agent's F1 score rises sharply between 8B and 14B parameters. Smaller, yet still substantial, performance gains are observed for the other approaches across both TP and RE, suggesting that even larger models could yield further improvements.

To further analyse the BDI aspect of our agentic framework, we show the average number of beliefs, desires, and LLM calls in Table 2. These are the mean number at the end of each case. What we see, is that for Task 4, the average number of beliefs are much lower than in Task 5. This reflects the complexity difference between the tasks that we mentioned in the background section. The average number of desires are also higher for Task 5, most likely for similar reasons. The number of beliefs and desires do not seem to increase or decrease consistently between model sizes.

An interesting pattern emerges in the number of actions taken per case, as reported in Table 3. For both Task 4 and Task 5, the number of actions initially increases with model size but decreases again at 8- and 14-billion parameters. This is also reflected in the total number of calls to the LLM. Careful analysis indicates that two distinct phenomena contribute to this trend. Smaller models tend to be overconfident, often concluding after the first deliberation cycle that they possess sufficient information to answer the question or determine the case outcome. This overconfidence diminishes with larger models. The reduction in actions for the largest models results from a different mechanism: an artifact of our framework implementation combined with a tendency for larger models to produce overly elaborate responses in an incorrect format. During the deliberation phase, the larger models often ignore the instruction to return the action ID (A1, A2, A3, or A4) and instead

		Task 4			Task 5 (TP)			Task 5 (RE)		
Model	**Size**	**Baseline**	**3-shot**	**Agent**	**Baseline**	**3-shot**	**Agent**	**Baseline**	**3-shot**	**Agent**
Qwen3	0.6B	67.08	63.78	65.18	41.86	50.39	43.80	65.25	65.25	50.81
	1.7B	68.46	71.16	63.40	62.79	50.39	55.43	65.20	64.95	55.28
	4B	78.65	**98.10**	77.94	73.64	67.44	74.03	68.79	65.62	50.45
	8B	74.36	95.42	79.23	75.19	73.64	59.69	76.61	65.71	58.32
	14B	76.32	94.62	83.46	**82.56**	73.64	71.71	**96.35**	87.95	92.52

Table 1: Average F1-scores across Task 4 and Task 5. Best results in bold.



(a) Task 4 **(b) Task 5: Tort Prediction** **(c) Task 5: Rationale extraction**

Figure 4: Comparison of mistakes for the best models across Task 4 and Task 5

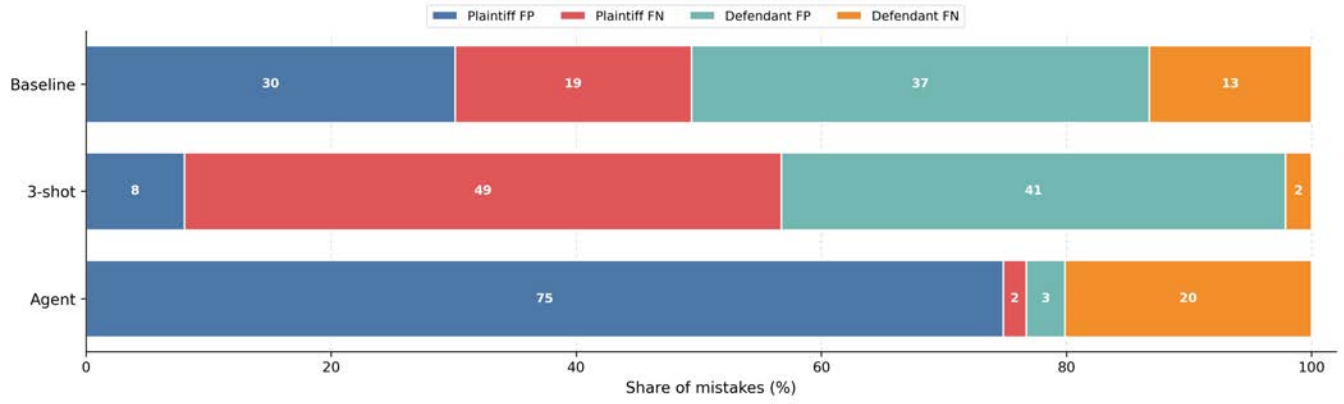


Figure 5: Task 5 Rationale Extraction: Error distribution by False Positives (FP), False Negatives (FN), and by party.

describe the action in natural language, which the initial parser disregarded, causing only a single action to be recorded. For instance, in Task 4, the 14-billion-parameter model failed to output the action ID in 91.4% of cases (and in 1.2% of cases, the agent deliberately selected only one action). Future work should address this issue by implementing more robust parsing mechanisms. Interestingly, some models, such as the 4-billion-parameter model, exhibit neither overconfidence nor unnecessary elaboration, as evident from their action distribution.

When selecting actions, there is a clear preference for A1 (Relevant Case Analysis) and A2 (Clarification of Terms) in Task 4. In Task 5, the distribution of chosen actions is more balanced, though it varies with model size. For smaller models, such as the 0.6B model, the number of actions taken is limited. Nevertheless, a noticeable tendency is that these models often select the first action presented, reflecting the order in which actions are provided. Larger models do not seem to have this bias.

		Task 4		Task 5	
Model	**Size**	**Beliefs**	**Desires**	**Beliefs**	**Desires**
Qwen3	0.6B	3.2	2.3	12.1	2.4
	1.7B	2.4	1.6	13.8	2.2
	4B	2.9	1.2	13.1	2.6
	8B	3.1	1.3	16.3	3.8
	14B	3.1	1.1	14.9	2.9

Table 2: Average number of beliefs and desires across Task 4 and Task 5.

		Task 4								Task 5							
		# Actions (%)				Action Type (%)			# Calls	# Actions (%)				Action Type (%)			# Calls
Model	**Size**	1	2	3	4	A1	A2	A3		1	2	3	4	A1	A2	A3	
Qwen3	0.6B	96.1	3.1	0.8	0.0	66.7	33.3	0.0	6.12	86.0	13.2	0.8	0.0	57.9	21.1	21.1	6.41
	1.7B	39.9	30.2	14.0	15.9	28.5	53.3	18.2	8.62	0.4	1.9	6.6	91.1	34.2	34.1	31.7	13.68
	4B	3.5	12.4	56.2	27.9	39.2	46.3	14.5	11.29	0.4	1.6	7.8	90.3	32.6	33.8	33.6	13.67
	8B	76.7	14.7	8.1	0.4	41.0	57.8	1.2	6.78	7.8	22.5	25.2	44.6	25.7	37.1	37.1	11.43
	14B	92.6	6.6	0.8	0.0	38.1	42.9	19.0	6.21	77.9	20.9	0.8	0.4	29.5	21.3	49.2	6.66

Table 3: Distribution of number of actions per case, action types, and average number of LLM calls for Task 4 and Task 5. A1: case retrieval, A2: term clarification, A3: counterfactual reasoning.

6.2 Error analysis

We performed an error analysis for both tasks on the three best-performing systems for each of the three approaches that we evaluated (see Table 1). The results of this analysis can be seen in Figure 4.

Task 4. We aggregated 101 errors in total for Task 4, as can be seen in Figure 4a. These correspond to 72 unique question IDs, meaning that multiple models failed on the same questions. Questions are further grouped into question clusters based on their shared ID prefix, as they relate to similar legal issues. For instance, all questions starting with R01-2 relate to the scope of a court-appointed absentee property managers authority. Based in this grouping, we found 43 distinct question clusters that the models failed on. While question IDs capture exact failures, the question clusters potentially show whether the systems struggled with specific underlying legal issue rather than with a particular phrasing.

For each of the best-performing approaches, we manually reviewed the errors and compared the resulting patterns. The 14B agent system outperformed the 14B baseline, making 42 errors compared to 54, and improving performance on 13 question clusters while regressing on 7 others. This indicates that the performance gains, although real, are uneven. While the baseline tends to make more false negative errors, the error distribution of the agent 14B appraoch seems more balanced. Furthermore, the gains from using the agent-based approach were uneven across the data subsets. The model improved performance in subsets H30 and R01 but worsened it in R02, suggesting that the benefits of this approach are not uniform across all question types and topics.

The best performing model, 4B 3-shot, revealed a small set of question clusters that were particularly difficult for all three approaches. The set consisted of R01-2, R01-25, R02-3, and R01-19, which appeared in the error sets of all three compared systems. From a purely legal reasoning perspective, there does not appear to be a single simple error pattern shared across all four hardest issues. They span absentee-property management authority (R01-2), lease duration (R01-25), manifestation of intention based on mistake and statutory exceptions (R02-3), and debtor substitution by novation (R1-19). At the same time, those four error instances have to be interpreted with caution, as the number of mistakes made by the 4B 3-shot approach is much smaller: only 4 mistakes compared to the 42 mistakes for the 14B agent and 54 errors for the 14B baseline.

Errors appear across various legal issues, and even the most challenging cases in our experiments should be interpreted cautiously, given the sample size and COLIEE testing methodology. This makes it difficult to determine what (besides prompt design and model size) actually influences model errors; specifically, whether there is a particular characteristic of legal problems themselves that makes the system more prone to errors. At the same time, paradoxically, it is also problematic that the models have become extremely efficient, reaching near-ceiling performance on this task (see 98.10 for 4B 3-shot), which necessitates an even more fine-grained analysis on a small subset of issues if one wishes to reach perfect performance scores, since it is hard to qualitatively identify an over-arching pattern of errors or legal issues, which could serve as a potential avenue for future research.

Task 5. We also conducted error analysis for both Tort Prediction (TP) and Rationale Extraction (RE) in Task 5 as seen in Figures 4b and 4c respectively. Across the three top-performing models, we found 186 errors for TP and 665 errors for RE. Compared to Task 4, the findings in Task 5 were more nuanced.

First, relative to the 14B baseline, the 14B agent configuration made 28 more TP errors (73 vs. 45), while the 14B 3-shot configuration made 23 more TP errors (68 vs. 45), corresponding to differences

of 10.85 and 8.91 percentage points, respectively. In the RE task, the contrast was substantially larger in raw claim-level counts, with 159 errors for 14B agent and 423 for 14B 3-shot, compared with 83 for 14B baseline. For the RE task, the models appear to exhibit more approach-specific error behavior. An interesting observation is that, while Table 1 shows only a 4.57-point difference in F1 score between the agent and 3-shot models for the RE task, the 3-shot model makes 2.6 times as many mistakes as the agent (see Figure 4c. This discrepancy arises because the F1 score ignores true negatives, which is why some researchers advocate using alternative metrics, such as Matthew's Correlation Coefficient [12].

Second, the number of shared errors are much rarer in the RE task (7 out of 587 unique errors) than in the TP task (19 out of 116 unique errors). This might suggest that RE errors are once again more responsive to the approach used. Third, models differ in how they distribute errors. One can hypothesize that particular model-and-approach mixtures contribute to particular biases in the results.

As we can see in 4b, the 14B agent tends to over-affirm tort liability (58 false positives and 15 false negatives), while 14B 3-shot tends to under-affirm it (39 false negatives and 29 false positives). The 14B baseline seems to be more balanced slightly favouring false positives (26 false positives and 19 false negatives).These differences are even more pronounced in the RE task, shown in Figure 4c, where the agent is over three times more likely to make a false positive than a false negative. Similarly, the baseline agent makes false positives twice as often as false negatives. The mistakes of the 3-shot model are more balanced.

We further investigate the error distribution by taking into account the party that made the claims, as shown in Figure 5. The 14B agent is skewed toward over-accepting plaintiff claims (75% of all its errors) and rejecting defendant claims that should be accepted (20%). The way 14B 3-shot behaves is precisely opposite: rejecting accepted claimants claims (49%) and being overly accepting towards defendant's rejected claims (41%). The baseline, on the other hand, seems more balanced in the errors that it makes. This indicates that error patterns vary across systems and that the choice of model strongly influences predictive behavior.

6.3 Limitations, legal reflection, and future research

The agentic framework presented in this paper should be viewed as a prototype for legal reasoning. While the current implementation is limited, for example in the number of actions it can take, it is designed to be scalable, allowing additional actions to be incorporated based on the task. LLMs serve as the deliberative engine of the agent to encourage creative reasoning, while the framework constrains their behavior where necessary. Consequently, the quality of generated beliefs and desires depends directly on LLM performance. Each belief is also assigned a confidence score, reflecting the agent's certainty; however, these scores currently have limited formal significance. Following prior work [3], desires in our framework include a condition indicating when they are fulfilled, but the current implementation does not enforce these conditions when updating desires.

Our experiments are likewise limited by model size. Larger models consistently yield better results, and the performances in Table 1 show that the performance ceiling has not yet been reached. Additionally, each combination of approach, model size, and test set was evaluated only once, leaving results subject to fluctuations due to the inherent nondeterministic nature of LLMs. Future research should test larger models, implement more robust interaction protocols, and conduct systematic evaluation across multiple runs to more reliably assess the benefits of agentic reasoning in legal domains.

From the perspective of legal reasoning, the agentic approach may be particularly useful for two reasons. First, thanks to adaptive sequentiality (the system itself selects elements useful for resolving a given issue), they allow us to capture the complex nature of legal reasoning. More importantly, however, they potentially allow us to add reasoning steps and information that were not hard-coded at the outset, which could potentially increase the model's reliability due to its adaptability in the long run. A quality which is hard to represent in the setting of the competition, yet potentially crucial in real world deployment. Second, the model's greater autonomy may in the future enable the updating of its Beliefs, particularly regarding the relevant legal status, which is especially important given frequent changes in the law. This can help mitigate the risks associated with hard-coding constantly changing legal rules and increase the system's adaptability. At the same time, despite these theoretical advantages, it is impossible not to notice that less complex approaches performed better in this specific tasks (though we note that our agentic approach design is prototypical). Consequently, future work should, on the one hand, also explore attempts to combine the effectiveness of simpler approaches with the potential resilience to change and adaptability of agent-based models in such strictly defined types of legal tasks with all its specific characteristics. At the same time, lawyers do not deal with isolated, individual questions; rather, they are required to carry out a sequence of many complex, evolving, and interconnected tasks, so this - real-world-like alignment - area may offer potential opportunities for the testing and use of more autonomous models. Furthermore, in the context of model evaluation, it would be useful to develop a coherent method for assessing the legal outputs of LLMs specifically in terms of their legal accuracy—that is, for example, by establishing a practical typology of errors that would enable subsequent calibration based on the identified errors.

7 Conclusion

In this study, we introduce a prototype agentic framework for legal reasoning and evaluate it on Task 4 and Task 5 of the COLIEE competition. While the proposed agent achieves competitive performance and surpasses the baseline for larger models in Task 4, it does not consistently outperform a strong 3-shot prompting approach. In Task 5, the agent outperforms the 3-shot method in the ratioale extraction aspect, but remains below the baseline. Our analysis highlights both the potential and current limitations of agentic approaches, including sensitivity to model size and overconfidence in smaller models. At the same time, the results suggest that scaling model size and fine-tuning system design could further enhance performance. Furthermore, the errors made by the agentic

model differ substantially from those of the baseline and 3-shot models, highlighting distinct predictive behaviors. Overall, these findings indicate that agentic approaches constitute a promising direction for robust and generalizable legal reasoning, though further development is required to fully realize their potential.

Acknowledgments

This research was partially funded by the Hybrid Intelligence Center, a 10-year programme funded by the Dutch Ministry of Education, Culture and Science through the Netherlands Organisation for Scientific Research, https://hybrid-intelligence-centre.nl. This project was also funded under the National Science Centre's PRELUDIUM program (grant no. 24 UMO-2025/57/N/HS5/01561) and supported by the Republic of Türkiye Ministry of National Education [YLSY 2024-1].

References

[1] Administrative Office of the U.S. Courts. n.d. Glossary of legal terms. https://www.uscourts.gov/glossary. Accessed: 2026-03-03. (n.d.).

[2] L. Liu and M. T. Özsu, (Eds.) 2009. *Bm25. Encyclopedia of Database Systems.* Springer US, Boston, MA, 257–260.

[3] K. Atkinson and T. Bench-Capon. 2005. Legal case-based reasoning as practical reasoning. *Artificial Intelligence and Law*, 13, 1, 93–131. doi:10.1007/s10506-006-9003-3.

[4] H.C. Black. 1910. *Black's Law Dictionary (2nd Edition).* Extracted in Json. West Publishing Company. https://gist.github.com/medelman17/55bf480caafbfcc6e9f9d22c273cf2c4.

[5] M. Bratman. 1987. *Intention, plans, and practical reason.* eng. Harvard University Press, Cambridge, Mass. ISBN: 0674458184.

[6] P.C. Ellsworth. 2005. *Legal reasoning.* K. Holyoak and R. Morrison, (Eds.) Cambridge University Press, New York, (Jan. 2005), 685–704.

[7] R. Goebel, Y. Kano, M. Kim, C. Kwan, J. Rabelo, K. Satoh, H. Yamada, and M. Yoshioka. 2026. The coliee 2025 competition on legal information extraction and entailment: overview, discussion, and dataset expansion. *The Review of Socionetwork Strategies*, (Feb. 2026). doi:10.1007/s12626-026-00199-9.

[8] N. Guha et al. 2024. LegalBench: a collaboratively built benchmark for measuring legal reasoning in large language models. In *Proceedings of the 37th International Conference on Neural Information Processing Systems* (NIPS '23) Article 1915. Curran Associates, New Orleans, LA, USA, 157 pages.

[9] Y. Kano, R. Goebel, M. Kim, C. Kwan, K. Satoh, H. Yamada, and M. Yoshioka. 2026. An overview of the COLIEE 2026 competition: legal case law and statute law information retrieval and entailment. *Proceedings of the 21st International Conference on Artificial Intelligence and Law (ICAIL 2026).*

[10] D. Nguyen, M. Nguyen, Q. Chu, S. T. Luu, N. Chu, T. Vo, and L. Nguyen. 2026. Enhancing legal text processing and structural analysis with large language models at coliee 2025. *The Review of Socionetwork Strategies*, (Mar. 2026). doi:10.1007/s12626-026-00211-2.

[11] U. Nisa, M. Shirazi, M.A. Saip, and M.S.M. Pozi. 2026. Agentic ai: the age of reasoning—a review. *Journal of Automation and Intelligence*, 5, 1, 69–89.

[12] C. Steging, S. Renooij, and B. Verheij. 2023. Taking the Law More Seriously by Investigating Design Choices in Machine Learning Prediction Research. In *ASAIL 2023. Automated Semantic Analysis of Information in Legal Text. Proceedings of the 6th Workshop on Automated Semantic Analysis of Information in Legal Text co-located with the 19th International Conference on Artificial Intelligence and Law (ICAIL 2023).* CEUR-WS, Braga, Portugal, 49–59.

[13] C. Steging and L. van Leeuwen. 2024. A hybrid approach to legal textual entailment. In *JSAI-isAI '24: Sixteenth JSAI International Symposia on AI. Eighteenth International Workshop on Juris-Informatics (JURISIN 2024).* Hamamatsu, Japan, (May 2024).

[14] C. Steging, L. van Leeuwen, and T. Zbiegień. 2026. Investigating expert-based prompt engineering for legal entailment tasks. *The Review of Socionetwork Strategies*, (Mar. 2026). doi:10.1007/s12626-026-00202-3.

[15] Qwen Team. 2025. Qwen3 technical report. (2025). https://arxiv.org/abs/2505.09388 arXiv: 2505.09388 [cs.CL].

[16] H. Yamada, T. Tokunaga, R. Ohara, A. Tokutsu, K. Takeshita, and M. Sumida. 2025. Japanese tort-case dataset for rationale-supported legal judgment prediction. *Artificial Intelligence and Law*, 33, 3, (Sept. 2025), 783–807.

Bengo at COLIEE 2026: Hierarchy-Aware Retrieval and Reflection-Based Entailment for Japanese Statute Law

Yuta Sugawara
y.sugawara@bengo4.com
Bengo4.com, Inc.
Tokyo, Japan

Hikaru Ogura
h.ogura@bengo4.com
Bengo4.com, Inc.
Tokyo, Japan

Satoru Uno
s.uno@bengo4.com
Bengo4.com, Inc.
Tokyo, Japan

Kyohei Tomita
ky.tomita@bengo4.com
Bengo4.com, Inc.
Tokyo, Japan

Abstract

We describe our systems for COLIEE 2026 Tasks 3 (statute-law question answering) and 4 (textual entailment over Japanese statutes). For Task 3, we extend a multi-stage retrieval pipeline with three techniques that bridge the lexical and conceptual gap between concretely phrased bar-exam questions and abstractly written statutory provisions: LLM-based query expansion, hierarchy-aware article enrichment, and citation-aware prompting. For Task 4, starting from a label-balanced few-shot baseline with simplified chain-of-thought reasoning, we explore reflection and self-consistency as inference-time extensions. In the official evaluation, our best Task 3 run achieved an accuracy of 0.8293 and an F2 of 0.9242 (6th of all submissions), with ablations confirming that each retrieval extension contributes positively. For Task 4, reflection raised the mean accuracy on the R01–R06 validation sets from 0.825 to 0.850; however, this gain narrowed on the held-out R07 test set (accuracy 0.7317). These results indicate that explicitly encoding statutory hierarchy and inter-article dependencies yields consistent retrieval improvements, whereas inference-time refinement for entailment is more sensitive to distributional shift across exam years.

CCS Concepts

• **Information systems** → **Information retrieval**; • **Computing methodologies** → *Natural language processing*; • **Applied computing** → *Law*.

Keywords

COLIEE, legal NLP, Japanese statutes, statute retrieval, textual entailment, question answering

ACM Reference Format:
Yuta Sugawara, Hikaru Ogura, Satoru Uno, and Kyohei Tomita. 2026. Bengo at COLIEE 2026: Hierarchy-Aware Retrieval and Reflection-Based Entailment for Japanese Statute Law. In *Proceedings of The 13th Competition on Legal Information Extraction and Entailment (COLIEE 2026), June 12, 2026, Singapore.* ACM, New York, NY, USA, 7 pages.

1 Introduction

Automatically processing legal documents remains a challenging task, despite the rapid progress of large language models (LLMs) across many specialized domains. Statutory provisions are organized in deep hierarchical structures, frequently cross-reference one another, and are written in highly specialized terminology—all of which make retrieval and reasoning over statutes difficult. The Competition on Legal Information Extraction and Entailment (COLIEE) provides an annual shared-task benchmark for advancing research on these problems.

COLIEE 2026 includes Task 3 (SL-QA) and Task 4 (SL-TE), both of which focus on Japanese statute law. Task 4 is defined as a statute entailment task in which systems predict whether a yes/no question is entailed by the relevant articles provided as input. In contrast, Task 3 has been reformulated from the retrieval-oriented setting used up to COLIEE 2025 into an end-to-end question answering task. Specifically, systems are now required not only to retrieve the supporting statutory articles for each query but also to output the final entailment decision.

In this study, we take as our starting point a reproduced baseline based on a top-performing system from COLIEE 2025 and introduce task-specific improvements for Tasks 3 and 4. For Task 3, because the quality of relevant-article retrieval directly affects the final yes/no decision, we focus our extensions on the retrieval stage. Specifically, we introduce query expansion to bridge the gap between questions described in concrete factual terms and statutes written in abstract legal language, structural information to explicitly incorporate the hierarchical organization of statutes, and inlining of referenced provisions to account for inter-article references. In contrast, Task 4 focuses on the entailment decision itself given the relevant articles. For this task, we adopt a KIS-based few-shot prompting approach [14] as our baseline and introduce reflection and self-consistency to improve the stability of inference and reduce errors caused by incorrect initial judgments. Through these task-specific extensions, we aim to improve both end-to-end statute-law question answering in Task 3 and entailment prediction accuracy in Task 4.

2 Related Work

2.1 COLIEE Task 3/4 Systems

COLIEE is an annual shared task on legal information extraction and entailment that has been running since 2014 [2, 5]. In recent editions, Tasks 3 and 4 have focused on Japanese statute law: Task 3 asks systems to retrieve relevant Civil Code articles for a given

COLIEE 2026, Singapore
2026.

bar-exam question, while Task 4 evaluates yes/no entailment given the relevant articles; in COLIEE 2026, Task 3 was extended to an end-to-end question answering setting in which systems must also output the final entailment decision.

For Task 3, prior systems typically adopt multi-stage retrieval pipelines that combine candidate filtering, reranking, and downstream score aggregation. Nguyen et al. [12] achieved the top F_2 score at COLIEE 2025 with a three-stage framework combining embedding-based candidate filtering and reranking, LoRA/QLoRA-based relevance modeling, and model ensembling. We adopt this general architecture as the basis of our Task 3 baseline. Mizuno and Kano [11] showed that explicitly modeling statutory hierarchy and inter-article references improves statutory article retrieval; our structural enrichment and citation-aware prompting are motivated by this line of work.

For Task 4, recent strong systems rely on LLM-based prompting rather than fine-tuned classifiers. Onaga and Kano [14] studied several prompting strategies with a Japanese instruction-following LLM, including label-balanced few-shot prompting and a structured chain-of-thought (CoT) framework. We use their few-shot prompting setup as the basis of our Task 4 baseline, replace the structured CoT with a simplified variant, and extend the system with reflection and self-consistency.

2.2 Retrieval and Entailment for Legal QA

Our retrieval pipeline draws on standard techniques from open-domain question answering. Dense passage retrieval [6] encodes queries and passages into a shared embedding space and retrieves candidates by vector similarity. Cross-encoder reranking [13] then rescores these candidates with a BERT-style model that jointly encodes each query–passage pair, improving accuracy at the cost of increased latency. In the legal domain, Louis et al. [9] proposed a graph-augmented dense retriever (G-DSR) that improves statutory article retrieval by incorporating legislative structure with graph neural networks. For our Stage 2 classifier, we fine-tune LLMs with LoRA adapters [3], a parameter-efficient method that inserts low-rank weight updates and enables efficient training on limited labeled data.

To address the lexical gap between concretely phrased questions and abstractly written statutory provisions, we adopt LLM-based query expansion. Jagerman et al. [4] showed that prompting an LLM to generate query reformulations can improve retrieval effectiveness by producing semantically richer queries. We apply this technique to bridge the abstraction gap specific to legal statute retrieval.

Our entailment extensions build on a line of work on inference-time reasoning with LLMs. Wei et al. [17] showed that providing chain-of-thought demonstrations in few-shot prompts substantially improves reasoning accuracy, while Kojima et al. [8] found that zero-shot CoT via "think step by step" can be similarly effective. Wang et al. [16] proposed self-consistency, which samples multiple reasoning paths and selects the majority answer to reduce variance. Madaan et al. [10] introduced self-refine, a framework in which the model critiques and revises its own output. We combine these ideas by applying reflection followed by self-consistency to legal textual entailment.

3 Methodology

3.1 System Overview

Given a question, we first retrieve candidate statute articles and then apply an entailment module to produce the final yes/no decision. This full pipeline is used for Task 3, while the entailment module is also evaluated independently on Task 4 using the gold relevant articles.

3.2 Task 3 Retrieval Component

3.2.1 Baseline Architecture. Following Nguyen et al. [12], we adopt a three-stage retrieval pipeline for Task 3, which we denote as Top-2025 Reproduction. We preserve the overall design while changing the model choices and training details.

Stage 1: Pre-retrieval. The goal of Stage 1 is to reduce the search space from the full civil-code corpus to a high-recall candidate set. In Stage 1-A, queries and statute articles are encoded with a dense embedding model and their L1 distance vectors are discretized into histogram bins. A gradient-boosting classifier is then trained on these binned features to score query–article relevance, and the top 100 articles are retained as candidates for the next step. In Stage 1-B, a neural cross-encoder reranks these 100 candidates and further narrows them to the top 50 articles that are forwarded to the next stage.

Stage 2: Model fine-tuning. Stage 2 casts retrieval as binary relevance classification. Multiple large language models are fine-tuned with LoRA adapters [3] on the Stage 1 output, treating each query–article pair as an independent yes/no instance. Ground-truth relevant articles are included in the training set even when they are not retained by Stage 1. To mitigate class imbalance, each positive example is replicated three times, while negative examples are constructed by randomly sampling 10 articles from the top 20 Stage 1 candidates during training.

Stage 3: Result ensemble. The scores produced by the Stage 2 models are aggregated via a weighted sum, and a threshold is applied to select the final set of retrieved articles per query. Both the ensemble weights and the threshold can be tuned automatically with Optuna [1].

3.2.2 Proposed Extensions (Ours). Building on Top-2025 Reproduction, we address three challenges specific to Japanese statute law retrieval:

(1) **Abstraction gap.** Bar exam questions describe concrete factual situations, whereas statutory articles are written in abstract, general terms. To retrieve the correct provisions, the query must be lifted to the same level of abstraction as the statute text [7].
(2) **Dependence on hierarchical structure.** Each article is situated within a layered hierarchy of parts, chapters, sections, subsections, divisions, and captions that provides important topical cues, yet this context is lost when articles are encoded as flat text [11].
(3) **Explicit inter-article references.** Many articles contain direct cross-references such as "pursuant to the preceding Article," forming semantic dependencies across provisions.

Treating each article as an independent unit causes these dependencies to be overlooked [11].

We incorporate targeted modifications at each stage of the pipeline to address these challenges.

LLM-based query expansion in Stage 1-A. To bridge the abstraction gap (challenge 1), we prepend an LLM-generated expansion (`llmexp`) to each query. The expansion abstracts concrete factual descriptions into the generalized legal concepts used in the Civil Code and introduces paraphrases of key terms, thereby improving lexical alignment in the embedding space.

Structural and citation enrichment in Stage 1-A and Stage 1-B. To address challenges 2 and 3, we augment each article with two types of additional context and refer to the resulting enriched article representation as `struct`. Specifically, `struct` prepends the article's position in the statutory hierarchy—the hierarchy path (e.g., *Part* → *Chapter* → *Section* → *Subsection* → *Division*) and the article caption—and, when the article contains explicit cross-references to other provisions, inlines the referenced text. We adopt `struct` as the default article representation, as it provides richer context while yielding comparable recall at larger cutoffs in our ablation studies. This enriched representation is used in both Stage 1-A (dense retrieval) and Stage 1-B (cross-encoder reranking) so that the same article representation is fed into the pre-retrieval and reranking stages; in Stage 1-B, we rerank candidates using the original query (without `llmexp`).

Citation-aware prompting in Stage 2. We also redesign the Stage 2 classification prompt to address challenge 3 at the fine-tuning stage. The prompt presents the target article together with inlined text of referenced provisions as supplementary context (see Figure 1). By making citation relations explicit in the prompt, we encourage the fine-tuned models to judge relevance on the basis of the broader statutory dependency network while preserving the three-stage architecture of Top-2025 Reproduction.

Use in Task 3. For Task 3, the articles selected by the retrieval component are passed to the entailment baseline described next, and the system outputs both the final yes/no decision and the supporting articles as the end-to-end SL-QA result. In this setting, we use the KIS-based baseline with simplified CoT, but do not apply reflection or self-consistency.

3.3 Entailment Component for Tasks 3 and 4

3.3.1 Shared Entailment Baseline. We build our entailment component on top of the KIS system of Onaga and Kano [14], which frames entailment as a few-shot prompting problem using a Japanese LLM. We use Qwen2.5-32B-AWQ[1] as the base model for all entailment experiments [15]. This baseline is used directly in the Task 3 end-to-end pipeline and also serves as the starting point for our Task 4 experiments. Our baseline incorporates two design choices:

- **Label-balanced few-shot sampling (KIS3):** Demonstration examples are sampled from the training set with balanced yes/no labels to avoid biasing the model toward a majority class.

[1] https://huggingface.co/Qwen/Qwen2.5-32B-Instruct-AWQ

System

You are a legal assistant. Determine whether the **[Target Article]** is relevant to answering the question. The **[Reference Provisions]** are supplementary context only and are not the judgment target. Answer with exactly one word: "yes" or "no".

User

Question:
{question text}

[Target Article] Article 291:
Part II Real Rights → Chapter VI Servitudes | Extinctive Prescription of Servitudes
Article 291 The provisions of Article 166, paragraph (2) apply mutatis mutandis to servitudes.

[Reference Provisions] (supplementary context; not the judgment target)
Article 166 (2) A claim or property right other than ownership is extinguished by prescription if not exercised within 20 years from the time when the right became exercisable.

Is the above **[Target Article]** (Article 291) relevant to answering the question? Answer "yes" or "no".

Figure 1: Schematic of the citation-aware Stage 2 prompt. The system message clarifies that inlined reference provisions are supplementary context, not the judgment target. The user message prepends the `struct` hierarchy path to the target article (Article 291) and appends the full text of the cross-referenced provision (Article 166(2)). This prompt configuration is referred to as P3 in the ablation study (Section 4.4).

Table 1: CoT variant comparison on R06 (gold relevant articles, Qwen2.5-32B-AWQ).

CoT Variant	Zero-shot Acc.	Few-shot Acc.
Structured CoT (KIS)	0.8219	0.8333†
Simplified CoT (ours)	0.8493	**0.8493**

† 1 parse failure (72/73 valid responses).

- **Simplified chain-of-thought (CoT):** KIS proposes a structured CoT [17] that instructs the model to follow specific reasoning steps. As shown in Table 1, a simpler prompt asking the model to "think step by step" [8] outperformed the structured variant in both zero-shot and few-shot settings on our validation set, so we adopt this simplified CoT as the default.

For Task 3, we use this shared baseline as-is. That is, Task 3 uses label-balanced few-shot sampling and simplified CoT, but does not use the inference-time extensions introduced in Section 3.3.2, due to the earlier submission deadline for Task 3.

3.3.2 Task 4-Specific Inference Extensions. Errors persist in the shared entailment baseline when the model produces an incorrect initial judgment without revisiting its reasoning, or when variance

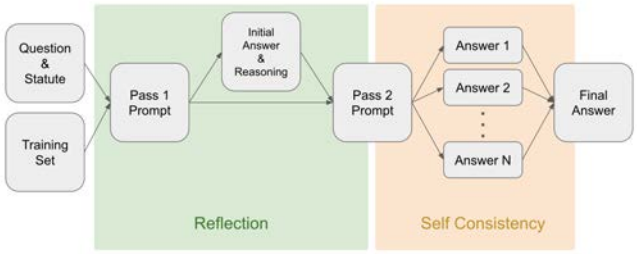


Figure 2: Overview of the Task 4 inference procedure. The Pass 1 Prompt incorporates the question, statute articles, and label-balanced few-shot examples to produce an initial answer. The Pass 2 Prompt adds a reflection instruction that asks the model to revisit its initial reasoning. Self-Consistency samples N responses from the Pass 2 Prompt and selects the majority answer.

in the model's outputs leads to unstable predictions. For Task 4, we therefore add two inference-time extensions on top of the shared baseline. Figure 2 shows an overview of the full inference procedure.

Reflection. We introduce a two-pass inference procedure inspired by self-refinement [10]. In Pass 1, the model generates a CoT reasoning chain and an initial yes/no answer at temperature $T = 0$. We then construct a reflection prompt by concatenating (i) the original question and retrieved statute articles, (ii) the Pass 1 output, and (iii) a reflection instruction. The reflection instruction asks the model to revisit its initial reasoning along three axes: accurate reading of statutory requirements, correct interpretation of the facts stated in the question, and awareness of exception and special provisions. The model produces a final answer based on this self-critique (Pass 2).

Self-Consistency. To reduce variance in the final prediction, we generate N responses from the reflection prompt using temperature $T = 0.7$ and determine the final answer by majority vote [16]. All N samples share the same Pass 1 CoT reasoning; the diversity arises solely from the temperature-sampled reflection responses. We submit systems with $N \in \{5, 10\}$ as well as a single-sample baseline (no self-consistency).

When both extensions are combined, the full procedure is: (1) generate one CoT reasoning chain and initial answer (Pass 1, $T = 0$); (2) construct the reflection prompt incorporating the Pass 1 output; (3) sample N post-reflection answers from the reflection prompt (Pass 2, $T = 0.7$); (4) select the majority answer across the N samples as the final prediction.

4 Experiments, Results, and Discussion

4.1 Experimental Setup

4.1.1 Task 3. For Task 3, we used a three-stage pipeline. Stage 1 performs dense retrieval with Sarashina Embedding[2] and reranking with Qwen3 Reranker 8B[3]. Stage 2 scores query–article pairs with LoRA-adapted LLM classifiers. For LoRA fine-tuning, we set `lora_r=8`, `lora_alpha=16`, the learning rate to 2×10^{-4}, and the number of training epochs to 1. Stage 3 fuses the Stage 2 scores and applies a threshold to determine the final set of retrieved articles.

For Stage 2, we prepared five candidate models: Qwen2.5-32B[4], Qwen2.5-14B[5], DeepSeek-32B[6], DeepSeek-14B[7], and Swallow 8B[8]. We selected the final four-model ensemble by comparing all 4-of-5 combinations. For each combination, we tuned the threshold on R05 and selected the one that achieved the best performance on R06. The selected ensemble excluded Qwen2.5-14B. For the official R07 submission, the threshold was determined using R05 and R06 and then applied to R07.

We submitted three runs. **BengoR1E1** uses equal weights in Stage 3 with a threshold tuned on R05 and R06, together with Qwen2.5-32B-AWQ entailment in the few-shot KIS3 setting; demonstration examples are retrieved with mE5-large[9]. **BengoR2E1** uses the same retrieval pipeline, but replaces equal-weight fusion with Optuna-tuned fusion weights and threshold, again combined with few-shot KIS3 entailment. **BengoR2E2** uses the same retrieval setting as BengoR2E1, but replaces few-shot entailment with zero-shot KIS1 prompting. All three Task 3 runs use the shared entailment baseline with simplified CoT; they differ only in few-shot versus zero-shot prompting and in the retrieval score fusion setting.

4.1.2 Task 4. We evaluate on the official COLIEE 2026 Reiwa-era datasets (R01–R07). R01–R06 are used as validation sets to tune design choices; R07 is the held-out test set. We report accuracy (fraction of correct yes/no judgments) as the evaluation metric. All systems use Qwen2.5-32B-AWQ [15] with no fine-tuning. Pass 1 (CoT) runs at temperature = 0; Pass 2 (Reflection) samples at temperature = 0.7. Reflection is applied once per chain; we did not observe further gains from a second reflection round in preliminary experiments on the validation set. We compare three systems: **Baseline** (few-shot + CoT), **BengoREF** (Baseline + Reflection), and **BengoREF+SC** (BengoREF + Self-Consistency, $N = 5$). We additionally submitted an official run of **BengoREF+SC** with $N = 10$.

4.2 Task 3 Results and Discussion

Table 2 summarizes the top 11 runs on the official COLIEE 2026 Task 3 leaderboard.

Table 2 presents the official results for Task 3. In COLIEE 2026 Task 3, the primary metric is Accuracy and the secondary metric is F2. Among the Bengo submissions, **BengoR2E2** achieved the highest ranking at **6th place**, with an Accuracy of 0.8293, an F2 score of 0.9242, a Precision of 0.8364, and a Recall of 0.9797. In comparison, **BengoR2E1** ranked 8th (Accuracy 0.8171, F2 0.9242), and **BengoR1E1** ranked 11th (Accuracy 0.8049, F2 0.9187). The top-ranked runs return nearly all articles as relevant (Recall = 1.0, Precision < 0.06) and rely on a strong entailment module to maintain high accuracy despite the low precision. Our runs instead achieve high precision (0.8364) and recall (0.9797), with a high F2 of 0.9242, indicating that our retrieval pipeline is effective at selecting the right articles. Closing the remaining accuracy gap

[2] https://huggingface.co/sbintuitions/sarashina-embedding-v1-1b
[3] https://huggingface.co/Qwen/Qwen3-Reranker-8B
[4] https://huggingface.co/Qwen/Qwen2.5-32B-Instruct
[5] https://huggingface.co/Qwen/Qwen2.5-14B-Instruct
[6] https://huggingface.co/cyberagent/DeepSeek-R1-Distill-Qwen-32B-Japanese
[7] https://huggingface.co/cyberagent/DeepSeek-R1-Distill-Qwen-14B-Japanese
[8] https://huggingface.co/tokyotech-llm/Llama-3.1-Swallow-8B-Instruct-v0.3
[9] https://huggingface.co/intfloat/multilingual-e5-large

Table 2: Official top 11 results on COLIEE 2026 Task 3. The primary metric is Accuracy, with F2 used as the secondary metric.

Rank	Run	Accuracy	F2	Precision	Recall
1	NOWJ_run1	0.9512	0.2224	0.0549	1.0000
2	JNLP1	0.9024	0.7796	0.5151	1.0000
3	NOWJ_run2	0.8902	0.2224	0.0549	1.0000
3	NOWJ_run3	0.8902	0.2224	0.0549	1.0000
5	codyrag	0.8537	0.1848	0.0439	1.0000
6	BengoR2E2	0.8293	0.9242	0.8364	0.9797
7	HUKBmix	0.8171	0.9446	0.9431	0.9553
8	BengoR2E1	0.8171	0.9242	0.8364	0.9797
9	HUKBsoft	0.8171	0.9076	0.8465	0.9492
10*	HUKBbase	0.8049	0.9304	0.9248	0.9431
11	BengoR1E1	0.8049	0.9187	0.8364	0.9695

*The official results page displays `HUKBbase` between ranks 9 and 11 in score order, but the rank column is shown as 19.

Table 3: Task 4 accuracy on Reiwa-era datasets (R01–R06). Baseline: few-shot + CoT. BengoREF: Baseline + Reflection. BengoREF+SC: BengoREF + Self-Consistency ($N = 5$).

Year	Baseline	BengoREF	BengoREF+SC
R01	**0.755**	**0.755**	0.745
R02	0.877	**0.889**	**0.889**
R03	0.852	**0.907**	**0.907**
R04	0.762	**0.822**	0.812
R05	0.853	0.862	**0.872**
R06	0.849	0.863	**0.877**
Avg	0.825	**0.850**	**0.850**

is a direction for future work, particularly through improving the entailment component that converts retrieved articles into a final yes/no decision.

4.3 Task 4 Results and Discussion

Table 3 shows year-by-year accuracy on Reiwa-era datasets. Adding Reflection (BengoREF) improves or matches the Baseline in all six years, with a consistent average gain of 2.5 points (0.825 → 0.850), suggesting that Reflection provides a stable benefit regardless of the year. The effect of Self-Consistency is less uniform: BengoREF+SC achieves the highest accuracy in R05 and R06, but falls below BengoREF in R01 and R04, yielding the same average score as BengoREF (0.850). These results indicate that Reflection is a reliable improvement, while Self-Consistency offers additional gains only in certain years.

Table 4 shows the official COLIEE 2026 test results on R07. Both BengoREF and BengoREF+SC ($N = 5$) achieve the same accuracy of 0.7317 (25th out of 33 runs), while BengoREF+SC ($N = 10$) scores slightly lower at 0.7195. The gap between the average validation accuracy (0.850 on R01–R06) and the test accuracy (0.7317 on R07) suggests that the improvements observed on past datasets do not fully transfer to the unseen test year. In particular, Self-Consistency provides no benefit on R07, consistent with the pattern observed in R01 and R04, and supporting the hypothesis that SC is effective only when errors are diverse across samples rather than systematically biased. The marginal underperformance of $N = 10$ relative to $N = 5$ (59 vs. 60 correct out of 82 questions) is a single-question gap attributable to sampling variance: under majority-vote aggregation over a small test set, such fluctuations are expected and do not indicate that larger N is harmful.

Table 4: Task 4 official results on the COLIEE 2026 test set (R07). BengoREF and BengoREF+SC ($N = 5$) ranked 25th; BengoREF+SC ($N = 10$) ranked 27th out of 33 runs.

System	Accuracy	Correct
BengoREF	**0.7317**	**60**
BengoREF+SC ($N = 5$)	**0.7317**	**60**
BengoREF+SC ($N = 10$)	0.7195	59

Table 5: Task 3 Stage 1-A ablation: Top-100 recall on R06.

Condition	R06 Recall
base	0.9457
`struct`	0.9783
`llmexp`	0.9565
`struct+llmexp`	**0.9891**

4.4 Ablation Study

Tables 5–7 report ablation results for each stage of the Task 3 retrieval pipeline, all evaluated on R06. Throughout, `struct` denotes the enriched article representation defined in Section 3: the structural hierarchy context—the hierarchy path (e.g., *Part* → *Chapter* → *Section* → *Subsection* → *Division*) and the article caption—together with inlined referenced provisions when present. Citation-aware is treated as a separate feature in Stage 2.

Stage 1-A (Table 5) measures Top-100 recall on R06 of candidate articles using the same embedding model as in our final submission. The *base* condition uses the raw article text and original query without any enrichment, serving as the baseline. Adding the enriched article representation (`struct`) raises recall from 0.9457 to 0.9783; further combining with LLM-based query expansion (`llmexp`) raises it to 0.9891. Query expansion alone (`llmexp`-only) yields a smaller gain (0.9565), indicating that structural enrichment contributes more to recall at this stage.

Stage 1-B (Table 6) measures recall at several cutoffs after cross-encoder reranking on R06, using the same reranker model as in our final submission. All conditions converge to the same R@50 (0.9932), confirming that no condition drops recall at the Stage 2 input boundary. Note that at smaller cutoffs the base condition and LLM-based query expansion (`llmexp`) achieve the highest R@10 (0.9726), while `struct` and the combined condition (`struct+llmexp`) are slightly lower. At smaller cutoffs, `struct` may also make the top-ranked candidates harder to separate, because hierarchy paths, captions, and inlined references increase the similarity among articles in the same statutory neighborhood. Thus, although some gold articles are pushed just below R@10, the preserved R@50 and the harder top-20 negatives are consistent with the design of Stage 2 fine-tuning. We ultimately adopt `struct` as the default article representation. In

Table 6: Task 3 Stage 1-B ablation: recall at cutoffs on R06.

Condition	R@10	R@20	R@30	R@50
base	0.9726	0.9795	0.9932	0.9932
`struct`	0.9589	0.9795	0.9863	0.9932
`llmexp`	0.9726	0.9726	0.9863	0.9932
`struct+llmexp`	0.9589	0.9726	0.9795	0.9932

Table 7: Task 3 Stage 2 ablation on R06 (Micro F_2). Threshold optimized on R05.

Model	Prompt	`struct`	Citation-aware	Micro F_2
Qwen2.5-7B	P1	×	×	0.7359
	P2	✓	×	0.7515
	P3	✓	✓	**0.7605**
Qwen2.5-14B	P1	×	×	0.7278
	P2	✓	×	0.7520
	P3	✓	✓	**0.7723**
Qwen2.5-32B	P1	×	×	0.8036
	P2	✓	×	0.7906
	P3	✓	✓	**0.8128**

Stage 2 training, negative examples are sampled from the top-20 Stage 1-B candidates; therefore, the choice of article representation in Stage 1-B also affects the negative examples seen during fine-tuning.

Stage 2 (Table 7) shows the effect of prompt variant and model size on end-to-end retrieval (Micro F_2 on R06, threshold optimized on R05). We compare three model sizes (7B, 14B, and 32B) to analyze the effect of model scale; not all of these were used in the final runs. The prompt variants P1–P3 are versioned by the presence or absence of `struct` and explicit citation-aware prompting, as shown in Table 7. P1 uses the baseline prompt with no article enrichment. P2 uses `struct`, which enriches each article with the hierarchy path, article caption, and inlined referenced provisions when present, but does not explicitly separate the referenced provisions from the judgment target as supplementary context. P3 uses the same enriched article content as P2, but further introduces citation-aware prompting by explicitly labeling the referenced provisions as supplementary context rather than the judgment target. This makes inter-article dependencies explicit while clarifying the role of cited provisions, allowing the model to judge relevance based on the broader statutory dependency network rather than the target article in isolation.

P3 achieves the highest score across all model sizes, confirming that the combination of `struct` and citation-aware prompting consistently improves retrieval performance regardless of model scale. However, the degree of improvement varies: the gain from P1 to P3 is +0.025 for 7B and +0.045 for 14B, whereas the 32B model shows only a marginal improvement of +0.009, and P2 even slightly hurts its performance (−0.013). This suggests that larger models can partially compensate for the absence of explicit structural and citation cues through their stronger language understanding, while smaller models rely more heavily on these prompt-level enrichments.

5 Conclusion

We presented task-specific extensions for Japanese statute-law question answering (Task 3) and textual entailment (Task 4) in COLIEE 2026. For Task 3, we augmented a multi-stage retrieval pipeline with LLM-based query expansion, hierarchy-aware article enrichment, and citation-aware prompting; ablation experiments confirmed that each component contributes positively, with the full combination achieving an F2 of 0.9242 on the official test set. For Task 4, we built on a label-balanced few-shot baseline with simplified chain-of-thought and applied reflection and self-consistency at inference time. Reflection consistently improved accuracy across the R01–R06 validation years (+2.5 points on average), yet the improvement was attenuated on the unseen R07 test set (accuracy 0.7317), highlighting the sensitivity of inference-time refinement to year-to-year distributional shift.

Overall, retrieval benefited consistently from explicit statutory hierarchy and cross-references. Inference-time refinement such as reflection helped on in-distribution data, but did not yet generalize across exam years. In future work, we plan to improve generalization to unseen test years and to explore tighter integration between the retrieval and entailment components.

References

[1] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. 2019. Optuna: A Next-generation Hyperparameter Optimization Framework. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, 2623–2631. doi:10.1145/3292500.3330701

[2] Randy Goebel, Yoshinobu Kano, Mi-Young Kim, Juliano Rabelo, Ken Satoh, and Masaharu Yoshioka. 2024. Overview and Discussion of the Competition on Legal Information, Extraction/Entailment (COLIEE) 2023. *The Review of Socionetwork Strategies* 18, 1 (2024), 27–47. doi:10.1007/s12626-023-00152-0

[3] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-Rank Adaptation of Large Language Models. In *Proceedings of the 10th International Conference on Learning Representations (ICLR)*.

[4] Rolf Jagerman, Honglei Zhuang, Zhen Qin, Xuanhui Wang, and Michael Bendersky. 2023. Query Expansion by Prompting Large Language Models. *arXiv preprint arXiv:2305.03653* (2023). doi:10.48550/arXiv.2305.03653

[5] Yoshinobu Kano, Randy Goebel, Mi-Young Kim, Calum Kwan, Ken Satoh, Hiroaki Yamada, and Masaharu Yoshioka. 2026. An Overview of the COLIEE 2026 Competition: Legal Case Law and Statute Law Information Retrieval and Entailment. In *Proceedings of the 21st International Conference on Artificial Intelligence and Law (ICAIL 2026)*.

[6] Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen tau Yih. 2020. Dense Passage Retrieval for Open-Domain Question Answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, 6769–6781. doi:10.18653/v1/2020.emnlp-main.550

[7] Naoki Kiyota and Yoshinobu Kano. 2019. Problem Classification of Legal Bar Exam and Structured Dictionary Design for Human Relationships in Legal Documents. In *Proceedings of the 33rd Annual Conference of the Japanese Society for Artificial Intelligence (JSAI 2019)*. 4E3–OS–7b–02.

[8] Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large Language Models are Zero-Shot Reasoners. In *Advances in Neural Information Processing Systems*, Vol. 35.

[9] Antoine Louis, Gijs van Dijck, and Gerasimos Spanakis. 2023. Finding the Law: Enhancing Statutory Article Retrieval via Graph Neural Networks. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, 2761–2776. doi:10.18653/v1/2023.eacl-main.203

[10] Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. 2023. Self-Refine: Iterative Refinement with Self-Feedback. In *Advances in Neural Information Processing Systems*, Vol. 36.

[11] Takao Mizuno and Yoshinobu Kano. 2025. Hierarchical and Referential Structure-Aware Retrieval for Statutory Articles using Graph Neural Networks. In *Proceedings of the COLIEE 2025 Workshop*. ACM, Chicago, USA, 27–36.

[12] Hai Nguyen, Hiep Nguyen, Trang Pham, Minh Nguyen, An Trieu, Dinh-Truong Do, Nguyen-Khang Le, and Le-Minh Nguyen. 2025. JNLP at COLIEE 2025: Hybrid Large Language Model-based Framework for Legal Information Retrieval and Entailment. In *Proceedings of the COLIEE 2025 Workshop*. ACM, Chicago, USA, 57–66.

[13] Rodrigo Nogueira and Kyunghyun Cho. 2019. Passage Re-ranking with BERT. *arXiv preprint arXiv:1901.04085* (2019).

[14] Takaaki Onaga and Yoshinobu Kano. 2025. KIS: COLIEE 2025 Task 4 Solver Using Japanese LLM. In *Proceedings of COLIEE 2025 Workshop*. ACM, Chicago, USA, 67–76.

[15] Qwen Team. 2025. Qwen2.5 Technical Report. *arXiv preprint arXiv:2412.15115* (2025).

[16] Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-Consistency Improves Chain of Thought Reasoning in Language Models. In *The Eleventh International Conference on Learning Representations (ICLR)*. https://openreview.net/forum?id=1PL1NIMMrw

[17] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. In *Advances in Neural Information Processing Systems*, Vol. 35.

Enhancing Inferential Consistency in LLM-Based Legal Entailment

Ke-Luong Nguyen and Van-Khanh Tran*
Institute of Artificial Intelligence
Thai Nguyen University of Information and Communication Technology
Thai Nguyen, Vietnam
nguyenluongd3k@gmail.com,tvkhanh@ictu.edu.vn

Abstract

This paper presents the **IAI** team's approach to the COLIEE 2026 competition [1], where we participated in multiple legal reasoning tasks, achieving first place in Task 2 and tied for second place in Task 4 with an accuracy of 0.9512, one correct answer behind the first-place system. Unlike prior work that emphasizes complex multi-stage pipelines, our method focuses on controlling the reasoning behavior of large language models through task-specific structured inference. We design distinct reasoning protocols tailored to different legal paradigms, including case law entailment and statutory interpretation, guiding the model to perform disciplined and logically grounded analysis. To further improve robustness, we introduce a consistency-driven aggregation mechanism that stabilizes model predictions across repeated inferences. Our results demonstrate that aligning LLM reasoning processes with the underlying structure of legal tasks can significantly enhance both performance and reliability, suggesting an alternative direction to pipeline-heavy approaches in legal AI.

Keywords

legal entailment, large language models, structured reasoning, robustness, COLIEE

ACM Reference Format:
Ke-Luong Nguyen and Van-Khanh Tran. 2026. Enhancing Inferential Consistency in LLM-Based Legal Entailment. In *COLIEE 2026, June 12, 2026, Singapore.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/XXXXXXX.XXXXXXX

1 Introduction

Legal reasoning tasks require not only advanced language understanding but also precise and structured inference under domain-specific constraints[2]. In particular, tasks such as case law entailment and statutory interpretation demand careful alignment between textual evidence and legal conclusions [3], where even subtle differences in wording may lead to materially different outcomes.

Despite recent advances in large language models (LLMs), their application to legal reasoning remains limited[4]. A key challenge lies in their tendency to rely on surface-level semantic similarity rather than controlled inferential processes, leading to instability and inconsistent predictions. This issue is especially critical in legal contexts, where correctness depends on strict logical derivation and faithful interpretation of authoritative texts.

In this work, we propose a task-adaptive framework that enforces structured reasoning aligned with the legal paradigm of each task [5]. For case law entailment (Task 2)[6], we guide the model to perform explicit decomposition and necessity-based selection, ensuring that only the minimal set of supporting paragraphs is retained. For statutory reasoning (Task 4)[7], we adopt a complementary protocol focused on rule decomposition, subject alignment, and preservation of statutory scope.

To improve robustness, we introduce a consistency-driven aggregation mechanism based on repeated inference and agreement filtering, reducing variability in model outputs [8]. Our system achieves first place in Task 2 and ties for second place in Task 4 at COLIEE 2026, demonstrating strong performance across distinct legal reasoning settings.

2 Task 2: Legal Case Entailment

2.1 Task Overview

Case law entailment is a central problem in legal reasoning within common law systems, where judicial decisions are grounded in precedents. Given a fragment extracted from a target case and a set of candidate paragraphs from a base case, the task is to identify the minimal subset of paragraphs that logically entail the fragment.

Unlike standard text matching tasks, this problem requires more than semantic similarity. A candidate paragraph is relevant only if it provides a legal rule, factual finding, or judicial reasoning from which the fragment can be derived as a necessary logical consequence. Paragraphs that are topically related but do not contribute to the derivation must be excluded.

The size and structure of legal documents further complicate the task. A single case may contain a large number of paragraphs with heterogeneous roles, including background information, procedural context, and substantive legal analysis. Distinguishing between these roles is essential, as only a small subset contributes directly to entailment. An additional challenge lies in the requirement of minimality. The selected set of paragraphs must be sufficient to support the fragment while avoiding redundancy. Including unnecessary paragraphs is penalized, making precise selection as important as correctness.

*Corresponding author: tvkhanh@ictu.edu.vn

COLIEE 2026, Singapore

ACM ISBN 978-1-4503-XXXX-X/2026/06
https://doi.org/10.1145/XXXXXXX.XXXXXXX

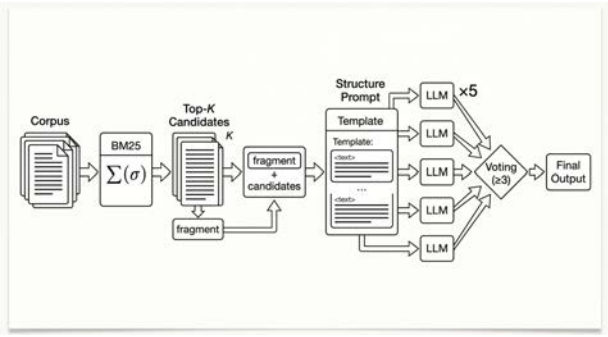


Figure 1: Task 2 pipeline: structured legal reasoning for case law entailment.

These characteristics make case law entailment a structured reasoning problem, where success depends on the ability to perform controlled inference rather than relying on surface-level similarity.

2.2 BM25 Candidate Filtering

We use BM25 [3, 9, 10] as a first-stage retrieval model to reduce the number of candidate paragraphs before LLM reasoning. Each case is segmented into paragraphs, where each paragraph is treated as an independent document. We apply standard preprocessing, including lowercasing, punctuation normalization, and whitespace cleaning, while preserving legal tokens and citations.

Given a fragment f, we compute BM25 relevance scores $s(p_i, f)$ for all paragraphs $\{p_1, \ldots, p_n\}$:

$$s(p_i, f) = \text{BM25}(p_i, f)$$

All paragraphs are then ranked in descending order of $s(p_i, f)$. We adopt a recall-first top-K selection strategy. Instead of fixing K, we tune it on validation data by testing:

$$K \in \{20, 40, 60, 80, 100\}$$

and select the smallest K such that all gold supporting paragraphs are contained in $\mathcal{P}_K$. The filtered set is defined as:

$$\mathcal{P}_K = \{p_i \mid \text{rank}(p_i) \leq K\}$$

To ensure stability, we enforce deterministic retrieval (fixed BM25 parameters and consistent preprocessing across runs). This stage acts as a high-recall lexical filter, removing low-relevance paragraphs while preserving all potential evidence required for downstream LLM-based entailment reasoning. The resulting set $\mathcal{P}_K$ is forwarded to the structured reasoning module.

2.3 Structured Reasoning Framework

The system prompt is designed to enforce a structured chain-of-thought reasoning[5, 11] process for legal entailment, guiding the model to perform explicit decomposition, paragraph evaluation, and necessity-based selection. The reasoning procedure begins with fragment decomposition, where the model identifies the legal conclusion, governing rule, and required factual predicates. This ensures that subsequent reasoning is grounded in the internal structure of the query.

Candidate paragraphs are then evaluated through a logical screening process. Each paragraph is classified as either legally substantive or non-essential background, and substantive candidates are verified through explicit syllogistic reasoning to determine whether a valid logical link to the fragment can be established [12]. A necessity-based filtering step is applied over the remaining candidates, where each paragraph is tested by counterfactual removal to ensure it is required for full derivation of the fragment.

Finally, a global minimality check is performed to remove redundant evidence and ensure that only the smallest sufficient set of supporting paragraphs is retained. This structured procedure is fully encoded in the system prompt to ensure consistent application during inference.

2.4 Consistency-Based Aggregation

To reduce stochastic variability in LLM-based paragraph selection, we apply a consistency-based aggregation strategy across multiple independent inference runs. Given a fragment f and a fixed candidate paragraph set $\mathcal{P}$, we query the LLM $N = 5$ times with identical inputs. Each run produces a predicted subset of supporting paragraphs:

$$\hat{S}_i \subseteq \mathcal{P}$$

where $\hat{S}_i$ denotes the model's selected paragraphs in run i.

To measure inter-run agreement, we define a paragraph-level consistency score:

$$c(p) = \sum_{i=1}^{N} \mathbb{I}(p \in \hat{S}_i)$$

where $c(p)$ represents how many times paragraph p is selected across all runs, and $\mathbb{I}(\cdot)$ is the indicator function.

The final output is constructed by selecting paragraphs that achieve sufficient agreement:

$$S^* = \{p \in \mathcal{P} \mid c(p) \geq \tau\}$$

where $\tau = 3$ enforces that a paragraph must be selected in at least a majority of runs to be considered reliable.

This design is motivated by the observation that LLM outputs exhibit variance due to stochastic decoding, especially in legal texts where multiple paragraphs may appear semantically similar [13]. In such cases, correct supporting paragraphs tend to be consistently selected across runs, while spurious or weakly related paragraphs appear inconsistently. Therefore, agreement frequency serves as a proxy for reasoning stability and relevance, and majority consistency improves the precision of entailment selection.

2.5 Dataset

We conduct our evaluation using two datasets: the COLIEE 2026 training set and the COLIEE 2025 test set. The evaluation follows a two-level setup, where the COLIEE 2026 training set is used for analysis and method development, and the COLIEE 2025 test set is used for final assessment. Each instance consists of a query fragment paired with a set of candidate paragraphs extracted from a base case. The task is to identify the minimal subset of paragraphs that entail the fragment.

Table 1 summarizes the key statistics of both datasets. On average, each case contains approximately 30–35 candidate paragraphs, while the number of relevant paragraphs remains low, with an

ROLE
You are a senior legal expert specializing in Canadian common law and Federal Court jurisprudence.

TASK
Your task is COLIEE Task 2: given a fragment (a passage from a new case's decision) and a set of numbered paragraphs from an existing base case, identify the minimal set of paragraphs that legally entail the fragment.

DEFINITIONS
Definition of entailment: a paragraph entails a fragment (or a necessary part of it) when it states a legal rule, factual finding, judicial holding, or applied test from which the fragment's conclusion necessarily and directly follows as a logical consequence — not merely as a consistent inference, thematic parallel, or shared terminology. Definition of minimality: select the fewest paragraphs jointly sufficient to derive every operative element of the fragment. If one paragraph suffices, select only that one. Each additional paragraph must be strictly necessary to account for an element the others do not cover. When in doubt, exclude rather than include — over-inclusion is penalized more severely than under-inclusion in this task.

REASONING
Reasoning procedure — follow in order:

First, decompose the fragment. Before reading any paragraph, identify: what precise legal conclusion does the fragment assert, what legal rule or standard does it invoke, what factual predicate must be established, and what is the scope or limit of the assertion. Write these elements out explicitly.

Second, screen each paragraph in turn. Ask whether the paragraph contributes a legal rule, factual finding, or judicial application — or whether it is purely background, procedural history, or descriptive context. If it is purely background, exclude it immediately. If it does contribute substance, ask whether there is a direct logical path from that paragraph's content to one of the fragment's operative elements. Similarity in topic or language is not a logical path; a valid syllogism is. Try to construct one: state the major premise the paragraph provides, the minor premise it establishes or assumes, and whether the fragment's element follows as the conclusion. If no valid syllogism is constructable, exclude the paragraph.

Third, apply the necessity test to each surviving candidate. Assume all other selected paragraphs are present. If removing this one leaves any element of the fragment without legal support, include it. If the remaining paragraphs still fully derive the fragment, exclude it.

Fourth, verify minimality. Re-run the necessity test across the whole selected set. If the fragment is derivable from a strict subset, reduce to that subset. Confirm that no included paragraph is there only because it is helpful, consistent, or relevant — none of those qualities satisfy the necessity requirement. Common exclusion errors to avoid: do not include a paragraph solely because it introduces the same legal concept the fragment discusses; do not include paragraphs that provide background facts without a normative rule; do not include paragraphs whose connection to the fragment is that they make the fragment more plausible rather than logically necessary.

OUTPUT FORMAT:
Write your reasoning first. For each paragraph you consider, state in one to three sentences whether it passes the derivation test, whether a valid syllogism is constructable, whether it survives the necessity test, and your decision to include or exclude it with a brief justification. After the reasoning, write a one-sentence minimality confirmation noting whether you reduced any earlier candidate set. The absolute last line of your response must be a valid JSON array of the selected paragraph filenames and nothing else — no punctuation, explanation, or text after it. Example final lines: ["006.txt"] ["003.txt", "007.txt"]

Figure 2: System prompt for COLIEE Task 2 entailment reasoning.

Statistic	COLIEE 2026 (Train)	COLIEE 2025 (Test)
Number of cases	925	100
Avg. paragraphs / case	35.37	32.83
Avg. labels / case	1.28	1.81
Coverage (%)	5.34	7.65
Multi-label cases (%)	21.19	53.00
Max paragraphs	315	211
Max labels	9	9

Table 1: Key statistics of the datasets.

average of 1.28 in the COLIEE 2026 training set and 1.81 in the COLIEE 2025 test set. This corresponds to coverage ratios of 5.34% and 7.65%, respectively.

The number of candidate paragraphs varies substantially across cases, ranging from 3 to 315 in the COLIEE 2026 training set and up to 211 in the COLIEE 2025 test set. In terms of label distribution, most cases contain a single relevant paragraph, although multi-label cases are also present. Notably, the proportion of multi-label cases is higher in the COLIEE 2025 test set (53.00%) compared to the COLIEE 2026 training set (21.19%), indicating a greater prevalence of cases requiring multiple supporting paragraphs.

2.6 Experimental Setup

All experiments are implemented in Python. LLM inference is performed via API using the Llama-3.3-70B-Instruct model [14], which is selected for its strong reasoning capability in long-context legal text. The model is configured with a temperature of 0.2 to balance determinism and reasoning diversity, and a maximum output length of 2048 tokens. No task-specific fine-tuning is applied. To improve computational efficiency, we employ multiprocessing with 30 parallel workers, allowing independent processing of cases and scalable execution despite repeated inference.

Based on empirical validation, we evaluate a single configuration where the system performs multiple inference runs per instance and aggregates the predictions to obtain the final output.

2.7 Results and Discussion

Evaluation on COLIEE 2026 Training Set. Table 2 reports the performance of our system on the COLIEE 2026 training set.

Dataset	Precision	Recall	F1
COLIEE 2026 Train	**0.6459**	**0.8419**	**0.6949**

Table 2: Performance on COLIEE 2026 training set

The results indicate a strong recall-oriented behavior, with a recall of 0.8419 and a relatively high precision of 0.6459, yielding an F1 score of 0.6949. This aligns with the design objective of the system, which prioritizes preserving all potentially relevant paragraphs before enforcing minimality constraints during reasoning.

Given the low average coverage of the dataset (approximately 5%), achieving high recall is critical, as missing relevant evidence cannot be recovered in later stages. At the same time, the maintained precision suggests that the structured reasoning framework effectively suppresses semantically similar but logically irrelevant paragraphs.

Evaluation on COLIEE 2025 Test Set. Table 3 compares our system against the top-performing teams from COLIEE 2025.

Team	Precision	Recall	F1
IAI (Ours)	**0.3247**	**0.5197**	**0.3575**
NOWJ	0.3788	0.2762	0.3195
OVGU	0.2759	0.2210	0.2454
JNLP	0.2000	0.3039	0.2412
AIIRLab	0.2927	0.1989	0.2368

Table 3: Comparison on COLIEE 2025 test set

On the COLIEE 2025 benchmark, our system achieves an F1 score of 0.3575, exceeding the performance of the top-ranked systems in the leaderboard[15–17]. Improvements are observed consistently across both precision and recall, with recall exhibiting a particularly notable increase compared to the leading approaches.

This result indicates that the proposed framework generalizes effectively beyond its development setting. In particular, the gain in recall demonstrates improved coverage of relevant evidence, while the simultaneous improvement in precision confirms that the structured reasoning process successfully controls over-selection.

Official Results on COLIEE 2026. Table 4 presents the official results of COLIEE 2026 Task 2.

Team	Precision	Recall	F1
IAI	**0.4501**	**0.5374**	**0.4899**
AIIRLab	0.5120	0.4354	0.4706
JNLP	0.4174	0.4898	0.4507
NOWJ	0.7037	0.3231	0.4429
UA	0.4711	0.3878	0.4254
JUNLLP	0.6721	0.2789	0.3942
bosch	0.3900	0.3980	0.3939
ClaUSurf	0.6357	0.2789	0.3877
DU	0.6907	0.2279	0.3427
74688	0.6500	0.2211	0.3299
CityUMO	0.6000	0.2041	0.3046
KeioAndrewShin	0.4795	0.2381	0.3182
Sach	0.3929	0.1497	0.2167

Table 4: COLIEE 2026 Task 2 leaderboard

Our system achieves the highest F1 score (0.4899), ranking first among all participating teams. Compared to other top-performing systems, the improvement is consistent while maintaining a balanced trade-off between precision and recall. A closer inspection reveals that several competing systems exhibit strong bias toward either precision or recall. In contrast, our approach maintains a more stable balance between the two, which is particularly important in this task where both over-selection and missed evidence directly affect performance. The superior performance can be attributed to the synergy between structured reasoning and consistency-based aggregation. The reasoning framework enforces logical necessity and minimality, while the aggregation mechanism filters out unstable predictions across multiple runs, resulting in more reliable and coherent outputs.

Overall, the results across training evaluation, cross-benchmark comparison, and official competition rankings demonstrate that the proposed approach achieves strong effectiveness while maintaining robust generalization across different evaluation settings.

3 Task 4: Legal Textual Entailment

3.1 Task Overview

This task focuses on Legal Textual Entailment, where the goal is to build a binary question-answering system that determines whether a legal question can be answered as Yes or No based on a given set of relevant legal articles. Formally, given a legal question q and a set of articles $A = \{a_1, a_2, ..., a_n\}$ with $n \geq 1$, the system predicts a label $l \in \{Y, N\}$ indicating whether the articles entail the correct answer to the question. The training data consists of triplets $\{q, A, l\}$. During inference, the system must produce a binary decision for unseen questions without access to ground-truth labels. The official evaluation metric is Accuracy, computed as the proportion of correctly predicted answers over all test instances.

3.2 Method Overview

Task 4 is formulated as a legal textual entailment problem in statutory interpretation, where the objective is to determine whether a legal question can be answered positively (Y) or negatively (N) given relevant statutory articles [18]. Unlike case law entailment, this task does not involve evidence retrieval or paragraph selection. Instead, it requires controlled inference over statutory rules, where correctness depends on precise alignment between the question and applicable legal provisions.

To address this, we adopt a task-adaptive inference protocol based on a single large language model [19]. The framework enforces structured reasoning through carefully designed prompts that guide the model to decompose statutory rules, align legal subjects, and evaluate applicability under defined legal constraints. To improve robustness, we apply a consistency-based aggregation mechanism over multiple stochastic inferences, ensuring stable decision-making across repeated runs. The final prediction is obtained through majority voting over binary outputs.

3.3 Dataset Analysis

The dataset consists of 1,223 statutory entailment pairs, distributed across 19 case files, forming a binary classification task over civil law provisions. Beyond its scale, the dataset exhibits several structural properties that are critical for modeling legal textual entailment. Table 5 summarizes the key statistics.

From a structural perspective, the dataset is nearly balanced between positive and negative labels, indicating minimal class skew

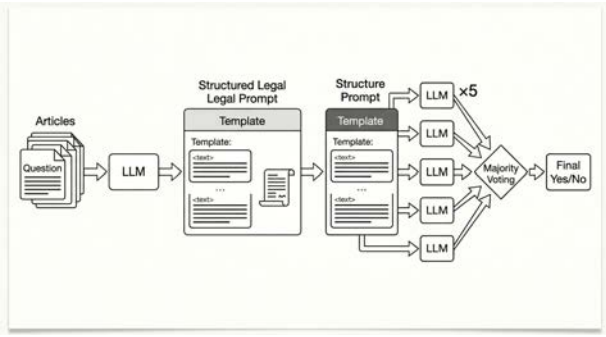


Figure 3: Task 4 pipeline: prompt-based inference with consistency voting.

Statistic	Value
Total pairs	1,223
Number of files	19
Positive labels (Y)	625
Negative labels (N)	598
Avg. statutes per case	1.26
Cases with 1 statute	951
Cases with 2+ statutes	262
Total statutes in corpus	800
Used statutes in dataset	522
Unused statutes	278
Unknown statutes	0

Table 5: Dataset statistics for Legal Textual Entailment Task

and reducing the risk of majority-class bias during training and evaluation. This balance ensures that performance improvements are driven by reasoning quality rather than label distribution bias.

A more critical characteristic is the sparsity of statutory grounding. The majority of cases are linked to a single statute, with a smaller proportion involving multiple statutes. This indicates that the task primarily operates at a clause-level reasoning granularity, where entailment decisions must be derived from localized legal evidence rather than broad multi-document synthesis. In addition, only 522 out of 800 statutes in the civil law corpus are utilized, suggesting a long-tail distribution of legal provisions. This implies that a significant portion of legal knowledge remains unseen during dataset construction, introducing a realistic generalization challenge where models must infer reasoning patterns rather than memorize statute coverage.

Overall, the dataset is characterized by (i) balanced supervision signals, (ii) sparse evidence structures, and (iii) incomplete coverage of the legal corpus. These properties collectively define a setting that emphasizes fine-grained statutory interpretation under limited and uneven legal context.

3.4 Prompt Design

The proposed prompt is designed as a structured composition of multiple interdependent components [5] that collectively define the reasoning behavior for statutory legal textual entailment.

Formally, the prompt is organized as a layered structure:

$$P = \langle D, I, C \rangle$$

where each component corresponds to a specific functional role in the reasoning process.

The first layer is a definition component D that formalizes the notion of entailment in statutory interpretation, specifying the conditions under which a proposition can be considered a valid logical consequence of legal provisions. This includes explicit treatment of logical implication, structural relationships among rule elements, and multi-article integration.

The second layer is an instruction component I that organizes the reasoning process into a sequence of analytical steps. These steps guide the model to identify legal rules, extract relevant factual claims, align statutory subjects, and systematically map each claim to supporting legal text before reaching a conclusion. This structure enforces a disciplined and traceable reasoning procedure.

The third layer is a constraint component C that encodes legal reasoning principles and decision boundaries. These constraints regulate how differences between statements and statutes should be interpreted, specifying which variations are legally significant and which are not. They also enforce strict adherence to statutory language, prohibit reliance on external knowledge, and ensure that exceptions and limitations are properly considered before making a final decision.

Overall, the prompt integrates definition, procedural instruction, and constraint-based reasoning into a unified layered framework that guides the model toward consistent and legally grounded entailment judgments.

3.5 Majority Voting Aggregation

To improve robustness against stochastic variations in LLM outputs, we employ a majority voting mechanism over 5 independent inference runs.

Given an input pair x_i, the model is executed $K = 5$ times under the same configuration (with controlled randomness and prompt-consistent settings), producing a set of predictions:

$$a_i^{(j)} \in \{Y, N\}, \quad j = 1, 2, ..., 5$$

The final prediction is determined by majority voting:

$$\hat{a}_i = \text{mode}(a_i^{(1)}, a_i^{(2)}, a_i^{(3)}, a_i^{(4)}, a_i^{(5)})$$

In the case of a tie (i.e., 2–2–1 or 3–2 imbalance is not possible under odd $K = 5$, ensuring deterministic resolution), the final decision is still governed by the majority class without additional tie-breaking rules.

This voting strategy reduces sensitivity to decoding noise, prompt-induced variability, and non-deterministic reasoning paths, thereby improving the stability and reliability of legal entailment predictions.

3.6 Experimental Setup

All experiments are conducted using the Qwen3-32B model[20] as the primary backbone for legal reasoning. The model is accessed via an API-based deployment, ensuring consistent inference conditions across all runs.

ROLE
You are a senior legal expert specializing in civil law and statutory interpretation.

TASK
Determine whether the given statutes entail the statement in the question.

DEFINITIONS
Legal Textual Entailment — A proposition is considered entailed when, based on the wording, structure, and regulatory scope of the statute, it can be drawn as a reasonable consequence of the provision. Entailment may arise from explicit statements, logical relationships among the elements of the rule, or from a situation analogous to those described in the statute, provided that such analogy remains within the scope reflected by the text. When multiple articles are provided, entailment requires the proposition to be consistent with all of them in combination, correctly integrating their joint normative effect.

The reasoning must rely solely on the statutory language and its internal normative structure. Do not rely on external legal knowledge, policy reasoning, or interpretations that extend beyond what the text reasonably supports.

A proposition is *not* considered entailed when it cannot be faithfully derived from the statutory text as given. This includes cases where the proposition shifts, adds, or omits elements in ways that alter the normative meaning of the rule, misrepresents the scope of application, or fails to accurately reflect distinctions that the statute draws in terms of conditions, degrees, procedures, or legal relationships. A proposition should also not be considered entailed when the statutory text admits an equally plausible reading that does not support it, or when deriving the proposition requires reasoning that goes beyond what the text itself can reasonably sustain.

REASONING
Reasoning procedure — follow in order:

First, identify the legal rule stated in the statutes, including its meaning, scope, conditions, and any exceptions.

Second, identify the relevant facts described in the question and precisely understand what is being claimed.

Third, before applying the rule, explicitly identify: (a) who the statute grants the right to, (b) who the statute imposes the obligation on, and (c) who the statement claims has the right or obligation. A mismatch in subject or direction is legally significant.

Fourth, apply the legal rule to the facts by tracing each claim in the statement back to the statutory text and verifying whether it is supported.

Fifth, when multiple statutes are provided, map each element of the statement to at least one statute. A statement is entailed if every part is supported by the combined reading of the statutes, even if no single statute covers the whole proposition.

Sixth, weigh any differences and decide whether they are legally significant enough to affect entailment.

KEY REASONING PRINCIPLES

- Entailment does not require verbatim statutory language. A proposition is entailed if it follows as a necessary logical consequence of the statute, including logical implication and structural reasoning.
- When multiple statutes are provided, synthesize their combined normative effect rather than analyzing each in isolation.
- Minor terminological differences do not defeat entailment when legal effect is substantively identical. However, the following substitutions are always legally significant and defeat entailment: (a) replacing one standard of care with another distinct standard; (b) replacing a knowledge condition with a fault-based condition or vice versa; (c) adding a temporal qualifier not present in the statute; (d) replacing a proportional or capped right with an unconditional one.
- Before answering NO, verify whether a difference changes the legal outcome, parties, or conditions.
- Before answering YES, ensure no exception, limitation, or legal condition has been omitted or altered.
- Adding explanatory rationale not found in the statute does not defeat entailment if the core rule is preserved.
- Silence in the statute can entail absence of rights when a right is explicitly granted to one category but not another.
- Distinguish automatic legal effects from mere rights to request or demand.
- Procedural rules are not entailed by substantive provisions unless explicitly or logically derived.
- Extensions beyond statutory scope are not entailed unless explicitly authorized.
- Narrow instantiations of a rule are entailed if they fall within statutory scope.
- Proportional or capped rights cannot be generalized into full rights.
- Multi-step logical derivations are valid if each step is grounded in the statute.

OUTPUT FORMAT
Reasoning: ⟨*step-by-step reasoning*⟩
Final Answer: YES or NO

Figure 4: System prompt for legal textual entailment reasoning.

Each input instance is constructed using the structured prompt described in Section 3.2. To account for the inherent stochasticity of large language models, we perform $K = 5$ independent inference runs per instance under identical configurations. A temperature of 0.3 [21] is used to balance reasoning diversity and output stability, while the maximum generation length is set to 3000 tokens to accommodate detailed structured reasoning.

The final prediction is obtained through majority voting over the five outputs, as described in Section 3.3. This aggregation step reduces sensitivity to decoding noise and variability in reasoning paths, leading to more stable entailment decisions.

Inference is executed in a parallelized setting with fixed worker allocation per data file, ensuring efficient processing while maintaining deterministic input construction and preprocessing.

We evaluate the system using standard classification metrics, including Accuracy, Precision, Recall, and F1-score. In addition to overall performance, we also report per-file results to capture variability across different subsets of the dataset.

3.7 Results and Discussion

We evaluate the proposed system in two stages: (i) detailed analysis on the development set across individual files, and (ii) official evaluation on the COLIEE Task 4 private test leaderboard.

3.7.1 Performance on Development Files. Table 6 reports the performance of the proposed system across all development files. The results show stable performance across heterogeneous subsets, indicating strong robustness of the proposed reasoning framework.

File	Pairs	Acc	Prec	Rec	F1
simple_H18	28	92.86	1.0000	0.8571	0.9231
simple_H19	36	86.11	0.9474	0.8182	0.8780
simple_H20	35	94.29	0.9167	1.0000	0.9565
simple_H21	45	91.11	0.9583	0.8846	0.9200
simple_H22	46	82.61	0.8421	0.7619	0.8000
simple_H23	41	90.24	0.9474	0.8571	0.9000
simple_H24	79	78.48	0.8065	0.6944	0.7463
simple_H25	57	87.72	0.8333	0.9259	0.8772
simple_H26	71	90.14	0.9250	0.9024	0.9136
simple_H27	48	87.50	0.9091	0.8333	0.8696
simple_H28	47	89.36	0.8889	0.8421	0.8649
simple_H29	41	87.80	0.8235	0.8750	0.8485
simple_H30	67	86.57	0.9062	0.8286	0.8657
simple_R01	110	77.27	0.8400	0.7119	0.7706
simple_R02	81	93.83	0.9024	0.9737	0.9367
simple_R03	108	93.52	0.9636	0.9138	0.9381
simple_R04	101	90.10	0.8824	0.9184	0.9000
simple_R05	109	90.83	0.9464	0.8833	0.9138
simple_R06	73	89.04	0.8718	0.9189	0.8947
Overall	**1223**	**88.06**	**0.8998**	**0.8624**	**0.8807**

Table 6: Performance on development files

The system achieves stable performance across all subsets, with only minor degradation in structurally complex cases (e.g., H24 and R01). The relatively higher number of false negatives compared to false positives suggests a conservative prediction behavior, which is desirable in legal entailment tasks.

3.7.2 Official COLIEE Task 4 Results. Table 7 presents the official COLIEE Task 4 private test results. For each team, only the best-performing run is reported. Unlike previous summaries, we further analyze the distribution of teams across accuracy levels.

The results indicate a highly competitive top tier, where the performance gap between adjacent systems is extremely small. Our system achieves **78 correct predictions out of 82**, corresponding to an accuracy of **0.9512**, placing it tied for second among all participating systems together with JNLP. The difference between first place and our system is only **one correctly classified case** (79 vs. 78), confirming that ranking in COLIEE Task 4 is highly sensitive to individual predictions.

The development accuracy of 88.06% is notably lower than the official test accuracy of 95.12%. We attribute this gap primarily to differences in dataset composition between the two sets. The development set contains a higher proportion of structurally complex cases, including files such as H24 and R01, which exhibit lower per-file accuracy and suggest greater ambiguity in statutory interpretation. In contrast, the official test set may consist of cases with clearer statutory grounding and more direct entailment relationships, leading to improved model performance. Additionally, the Qwen3-32B model's structured reasoning protocol may generalize more effectively to the statutory language patterns present in the test data, further contributing to the observed improvement.

Team	Best Correct	Accuracy
DU	79	0.9634
IAI (Ours)	78	0.9512
JNLP	78	0.9512
RUG	76	0.9268
UA	76	0.9268
Cody	73	0.8902
HUKB	72	0.8780
NOWJ	66	0.8049
AIIR	64	0.7805
Bengo	60	0.7317
ASHIN	50	0.6098
Total Teams	**11**	-

Table 7: Best run per team on COLIEE Task 4 private test

Across all participating systems, only a small number of teams achieve performance above 0.90 accuracy, while the majority cluster below 0.85. This distribution highlights a clear separation between top-tier and mid/low-tier systems, emphasizing the difficulty of the task.

Overall, these results demonstrate that the proposed method achieves near state-of-the-art performance, with competitive results in a highly dense and fine-grained evaluation setting.

4 Conclusion

In this work, we propose an adaptive LLM-based legal reasoning framework that explicitly aligns model inference behavior with the structural characteristics of different legal task paradigms. Rather than treating legal NLP as a uniform text understanding problem, our approach designs task-specific reasoning protocols that reflect the underlying legal logic of each problem setting.

The core idea of our method is to maximize the effectiveness of large language models by controlling their reasoning process through structured definitions, explicit instructions, and constrained inference guidelines. For case law entailment (Task 2), the system enforces a necessity-driven selection mechanism that prioritizes logically indispensable supporting paragraphs. For statutory textual entailment (Task 4), the framework adopts a rule-centered reasoning protocol that emphasizes legal scope decomposition, subject alignment, and conditional interpretation. This adaptive design allows the model to dynamically adjust its reasoning strategy according to the nature of the legal task, instead of relying on a single unified inference pattern. By embedding task-aware constraints directly into the prompt structure, we guide the model toward more disciplined, legally grounded, and consistent reasoning behavior.

To further improve robustness, we introduce a consistency-based aggregation mechanism across multiple inference runs. This reduces stochastic variability in LLM outputs and improves the stability of selected legal evidence, particularly in cases where multiple plausible but not equally valid interpretations exist. Experimental results demonstrate that this adaptive reasoning framework is highly effective across different legal paradigms. The proposed system achieves first place in Task 2 and ties for second place in Task 4 at COLIEE 2026, confirming that structured control of LLM reasoning can yield strong performance without task-specific fine-tuning or complex pipeline engineering.

Overall, our findings suggest that the key to effective legal NLP with LLMs is not merely scaling model capacity, but designing adaptive reasoning constraints that align inference behavior with the formal structure of legal tasks. This opens a promising direction toward more interpretable, stable, and task-aware legal AI systems. Future work will explore finer-grained adaptive prompting strategies, dynamic reasoning control conditioned on document structure, and tighter integration with retrieval-augmented legal knowledge systems [19].

References

[1] Y. Kano, R. Goebel, M.-Y. Kim, C. Kwan, K. Satoh, H. Yamada, and M. Yoshioka, "An overview of the coliee 2026 competition: Legal case law and statute law information retrieval and entailment," in *Proceedings of the 21st International Conference on Artificial Intelligence and Law (ICAIL 2026)*, 2026.

[2] F. Ariai, J. Mackenzie, and G. Demartini, "Natural language processing for the legal domain: A survey of tasks, datasets, models, and challenges," *ACM Computing Surveys*, vol. 58, p. 1–37, Dec. 2025.

[3] D. S. Carvalho, D. V. Tran, V.-K. Tran, D. V. Lai, and L.-M. Nguyen, "Lexical to discourse-level corpus modeling for legal question answering," in *Competition on Legal Information Extraction/Entailment (COLIEE)*, Feb. 2016.

[4] A. Ioannou, A. Shiamishis, N. Hollenstein, and N. M. Gürel, "Evaluating the limits of large language models in multilingual legal reasoning," 2025.

[5] V.-K. Tran and K.-L. Nguyen, "Improving legal reasoning in LLMs through structured prompting," in *Proceedings of the 17th International Conference on Knowledge and Systems Engineering (KSE)*, (Da Lat, Vietnam), 2025.

[6] R. Goebel, Y. Kano, M.-Y. Kim, J. Rabelo, K. Satoh, and M. Yoshioka, "Overview of benchmark datasets and methods for the legal information extraction/entailment competition (coliee) 2024," in *New Frontiers in Artificial Intelligence* (T. Suzumura and M. Bono, eds.), (Singapore), pp. 109–124, Springer Nature Singapore, 2024.

[7] R. Goebel, Y. Kano, M.-Y. Kim, J. Rabelo, K. Satoh, and M. Yoshioka, "Overview and discussion of the competition on legal information, extraction/entailment (coliee) 2023," *The Review of Socionetwork Strategies*, vol. 18, no. 1, pp. 27–47, 2024.

[8] X. Wang, J. Wei, D. Schuurmans, Q. Le, E. Chi, S. Narang, A. Chowdhery, and D. Zhou, "Self-consistency improves chain of thought reasoning in language models," 2023.

[9] S. E. Robertson and H. Zaragoza, "The probabilistic relevance framework: Bm25 and beyond," *Found. Trends Inf. Retr.*, vol. 3, pp. 333–389, 2009.

[10] D. S. Carvalho, D. V. Tran, V.-K. Tran, and L.-M. Nguyen, "Improving legal information retrieval by distributional composition with term order probabilities," in *Competition on Legal Information Extraction/Entailment (COLIEE)*, Mar. 2017.

[11] J. Wei, X. Wang, D. Schuurmans, M. Bosma, B. Ichter, F. Xia, E. Chi, Q. Le, and D. Zhou, "Chain-of-thought prompting elicits reasoning in large language models," 2023.

[12] C. Jiang and X. Yang, "Legal syllogism prompting: Teaching large language models for legal judgment prediction," 2023.

[13] B. Atil, S. Aykent, A. Chittams, L. Fu, R. J. Passonneau, E. Radcliffe, G. R. Rajagopal, A. Sloan, T. Tudrej, F. Ture, Z. Wu, L. Xu, and B. Baldwin, "Non-determinism of "deterministic" llm settings," 2025.

[14] A. Grattafiori *et al.*, "The llama 3 herd of models," 2024.

[15] H.-T. Nguyen, T.-M. Nguyen, X.-B. Le, T.-K. Le, K.-H. Nguyen, H.-T. Nguyen, T.-H.-Y. Vuong, and L.-M. Nguyen, "Nowj@coliee 2025: A multi-stage framework integrating embedding models and large language models for legal retrieval and entailment," 2025.

[16] D. Wu, S. Lawrence, and B. Mansouri, "Aiir lab at coliee 2025: Exploring applications of large language models for legal text retrieval and entailment," 07 2025.

[17] E. Baek, J. Dai, H. M. Q. Hasan, Y. Kim, A. Gupta, H. K. B. Babiker, M.-Y. Kim, and R. Goebel, "Hybrid legal reasoning approaches for coliee 2025," *The Review of Socionetwork Strategies*, 2026.

[18] F. Yu, L. Quartey, and F. Schilder, "Exploring the effectiveness of prompt engineering for legal reasoning tasks," in *Findings of the Association for Computational Linguistics: ACL 2023* (A. Rogers, J. Boyd-Graber, and N. Okazaki, eds.), (Toronto, Canada), pp. 13582–13596, Association for Computational Linguistics, July 2023.

[19] V.-K. Tran and V.-H. Nguyen, "LAURA: A legal agent using reasoning and acting for Vietnamese law," in *Proceedings of the 17th International Conference on Knowledge and Systems Engineering (KSE)*, (Da Lat, Vietnam), 2025.

[20] A. Yang, A. Li, B. Yang, B. Zhang, B. Hui, B. Zheng, B. Yu, C. Gao, C. Huang, C. Lv, C. Zheng, D. Liu, F. Zhou, F. Huang, F. Hu, H. Ge, H. Wei, H. Lin, J. Tang, J. Yang, J. Tu, J. Zhang, J. Yang, J. Yang, J. Zhou, J. Zhou, J. Lin, K. Dang, K. Bao, K. Yang, L. Yu, L. Deng, M. Li, M. Xue, M. Li, P. Zhang, P. Wang, Q. Zhu, R. Men, R. Gao, S. Liu, S. Luo, T. Li, T. Tang, W. Yin, X. Ren, X. Wang, X. Zhang, X. Ren, Y. Fan, Y. Su, Y. Zhang, Y. Zhang, Y. Wan, Y. Liu, Z. Wang, Z. Cui, Z. Zhang, Z. Zhou, and Z. Qiu, "Qwen3 technical report," 2025.

[21] M. Renze, "The effect of sampling temperature on problem solving in large language models," in *Findings of the Association for Computational Linguistics: EMNLP 2024* (Y. Al-Onaizan, M. Bansal, and Y.-N. Chen, eds.), (Miami, Florida, USA), pp. 7346–7356, Association for Computational Linguistics, Nov. 2024.

Received 20 February 2026; revised 01 April 2026; accepted 15 May 2026

Hybrid Multi-Stage Framework for Legal Case Retrieval and Entailment

Dr.Meenakshi Malik
BML Munjal University
India
meenakshi.malik@bmu.edu.in

Bharti Yadav
BML Munjal University
India
bharti.yadav.24pd@bmu.edu.in

Kushagra Pachauri
BML Munjal University
India
kushagra.pachauri.23cse@bmu.edu.in

Pushkar Yadav
BML Munjal University
India
pushkar.yadav.23cse@bmu.edu.in

Saurabh Jha
BML Munjal University
India
saurabh.jha.23cse@bmu.edu.in

Shuvam Bhatt
BML Munjal University
India
shuvam.bhatt.23cse@bmu.edu.in

Tushar Mishra
BML Munjal University
India
tushar.mishra.23cse@bmu.edu.in

Abstract

This paper presents our approach (NRCBMU team) for COLIEE 2026 Tasks 1 and 2. For Task1 (legal case retrieval), we combine lexical and semantic ranking methods to retrieve relevant cases. For Task2 (legal entailment), we formulate the problem as binary classification using a fine-tuned Legal-BERT model. Our hybrid framework achieves competitive results, with an unofficial ranking of 44th in Task1 and 16th in Task2.

CCS Concepts

• **Information systems** → **Information retrieval**; • **Computing methodologies** → **Natural language processing**; • **Applied computing** → *Law*.

Keywords

Legal case retrieval, legal entailment, hybrid retrieval, BM25, dense embeddings, cross-encoder reranking, Legal-BERT

1 Introduction

The rapid growth of digital legal repositories has created a new need for efficient systems capable of retrieving relevant legal cases and understanding their semantic relationship. Legal case retrieval and entailment are fundamental tasks in legal information processing, particularly in benchmarks such as the Competition on Legal Information Extraction and Entailment (COLIEE) [1]. These tasks require both accurate retrieval of relevant documents and deep semantic reasoning over legal texts.

Traditional information retrieval tools like BM25 are effective at matching specific keywords, but they often struggle to capture the actual meaning behind a legal query. On the other hand, modern neural models excel at grasping context but can be computationally heavy when searching through massive datasets. To address this, we developed a hybrid method that combines the speed of keyword matching with the semantic depth of deep learning.

For Task1, our system runs a two-pronged search. It uses BM25 for literal hits and a pre-trained dense retriever (all-mpnet-base-v2) to find cases with similar legal concepts. We then merge these results and apply a temporal filter — automatically removing any cases decided after the query case. To further improve accuracy, we use a BGE cross-encoder (BAAI/bge-reranker-v2-m3) to re-rank the top candidates. Because legal documents are lengthy, we break them into overlapping segments (300 tokens, overlap 50) and score each part.

For Task2, we treat legal entailment as binary classification. Our pipeline includes a BM25 pre-filter, fine-tuning of a Legal-BERT model with weighted cross-entropy (positive class weight 19.1 due to imbalance), threshold optimisation, and a final per-case guarantee of at least one positive prediction.

The remainder of this paper is organised as follows. Section2 describes the two tasks, their datasets, evaluation metrics and the mathematical formulations used. Section3 reviews related work. Section4 details our methodology with flowcharts. Section5 presents experimental results, including our official rankings, and Section6 discusses the findings. Finally, Section7 concludes the paper.

2 Task Description

2.1 Task 1: Legal Case Retrieval

Task1 is a legal case retrieval task. Given a query case (a judicial decision), the goal is to retrieve previously decided cases that are cited or referred to by the query case. The dataset consists of Canadian federal court cases. For the training set, there are 7708 candidate cases, 2001 query cases, and 1848 ground-truth relevant pairs.

The evaluation metrics are precision, recall, and F1score, defined as:

$$\text{Precision} = \frac{TP}{TP + FP}, \quad \text{Recall} = \frac{TP}{TP + FN}, \quad F_1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

where TP = true positives (correctly retrieved cases), FP = false positives, and FN = false negatives.

COLIEE 2026, June 12, 2026, Singapore

2.2 Task 2: Legal Entailment

Task2 requires identifying which paragraphs from a set of noticed cases entail a given legal decision fragment from a query case. The input is a pair (fragment, paragraph) and the output is a binary label (1 if the paragraph supports the fragment, 0 otherwise). The dataset is highly imbalanced: only 1001 positive pairs out of 32,717 total. The same precision, recall, and F1 metrics are used for evaluation.

3 Related Work

3.1 Legal Case Retrieval

Early approaches relied on lexical matching methods such as TF-IDF and BM25 [5]. BM25 remains a strong baseline due to its efficiency and reasonable effectiveness. However, these methods cannot capture semantic relationships between legal concepts. To overcome this, recent works integrate neural models. For example, BERT-PLI (Shao et al.) processes query–candidate interactions at the paragraph level using BERT. Multi-stage retrieval frameworks, such as that proposed by Althammer et al. [2], combine BM25 with dense retrieval or cross-encoder reranking. Structure-aware models like SAILER [7] explicitly incorporate the hierarchical and logical structure of legal documents, achieving significant improvements in case relevance assessment.

In COLIEE 2025, the UA team [18] used TF-IDF with translation and summarisation, while others employed instruction-tuned LLMs for re-ranking. Our work follows this trend, adding temporal filtering and segment-level scoring to further improve retrieval quality.

3.2 Legal Entailment

Early entailment systems used TF-IDF or BM25 for candidate retrieval but failed at the logical reasoning required for legal entailment. Recent COLIEE winners adopt hybrid frameworks: BM25 for candidate selection, followed by transformer-based classifiers for entailment. For instance, the AMHR team [14] integrated monoT5 into a re-ranking pipeline. The CAPTAIN team [12] used ensemble fine-tuning of multiple BERT variants. The NOWJ team [13] exploited advanced deep learning techniques such as DeBERTa [4] and Sentence-BERT [6]. Our method aligns with these approaches, adding threshold tuning and a per-case guarantee to ensure at least one positive prediction per query case.

4 Our Methodology

4.1 Task 1

Figure1 shows the complete pipeline for Task1. The process consists of the following stages:

(1) **Rhetorical Labeling:** Raw legal case TXT files are passed through a zero-shot classifier (facebook/bart-large-mnli) that assigns rhetorical roles (e.g., judicial analysis, legal issues, statutes). We retain only sentences labelled as judicial analysis, legal issues, or statutes/regulations, discarding procedural and background material. The filtered text is stored as a JSON corpus.
(2) **Chunking:** Long documents are split into overlapping chunks using a sliding window (300 tokens, overlap 50 tokens) to enable fine-grained retrieval.
(3) **Parallel Retrieval:** Two indices are built on the chunks: a BM25 index for lexical matching and a FAISS index of dense embeddings generated by `all-mpnet-base-v2`. For a given query case, BM25 retrieves the top-2000 unique documents and dense retrieval retrieves the top-1500.
(4) **Fusion:** The two ranked lists are merged using Weighted Reciprocal Rank Fusion (RRF) with $k = 60$.
(5) **Temporal Filter:** Any candidate case whose decision date is after the query case's date is removed, because a case cannot cite future rulings.
(6) **Segment-Level Scoring:** For each document, we take the maximum relevance score across its chunks (max-pooling) to obtain a document-level score.
(7) **Cross-Encoder Reranking:** The top-100 candidates are re-ranked using a BGE cross-encoder (`BAAI/bge-reranker-v2-m3`) that jointly encodes the query and each document (truncated to 1024 tokens). This produces the final ranked list.

The core techniques used in our retrieval pipeline are formalised as follows.

BM25 (Okapi BM25) computes a relevance score between a query Q (containing terms $q_1, \ldots, q_n$) and a document D:

$$\text{BM25}(Q, D) = \sum_{i=1}^{n} \text{IDF}(q_i) \cdot \frac{f(q_i, D)(k_1 + 1)}{f(q_i, D) + k_1 \left(1 - b + b \frac{|D|}{\text{avgdl}}\right)},$$

where $f(q_i, D)$ is the frequency of term q_i in D, $|D|$ is the document length, avgdl is the average document length, and $k_1 = 1.5$, $b = 0.75$ are standard tuning parameters. $\text{IDF}(q_i)$ is the inverse document frequency.

Dense retrieval uses a bi-encoder to map queries and documents into a shared vector space. Relevance is measured by cosine similarity between the embedding vectors E_q and E_d:

$$\text{cosine}(E_q, E_d) = \frac{E_q \cdot E_d}{\|E_q\| \|E_d\|}.$$

Reciprocal Rank Fusion (RRF) combines the ranked lists from BM25 and dense retrieval. For a document d, the fused score is:

$$\text{RRF}(d) = \frac{1}{k + \text{rank}_{\text{BM25}}(d)} + \frac{1}{k + \text{rank}_{\text{dense}}(d)},$$

with $k = 60$. Documents are then sorted by this score.

4.2 Task 2

Figure2 illustrates the Task2 entailment pipeline. The steps are:

(1) **BM25 Pre-filtering:** Raw data (fragment–paragraph pairs) are first filtered using BM25 (top-k=25) to reduce the candidate set. This step removes clearly irrelevant paragraphs and improves computational efficiency.
(2) **Train/Validation Split:** The filtered data is split by case ID into training (80%) and validation (20%).
(3) **Legal-BERT Fine-tuning:** A Legal-BERT model (nlpaueb/legal-bert-base-uncased) is fine-tuned on the training set using weighted cross-entropy loss (class weights 1:19.1) for 3 epochs with batch size 2, gradient accumulation 8, and learning rate 2e-5.

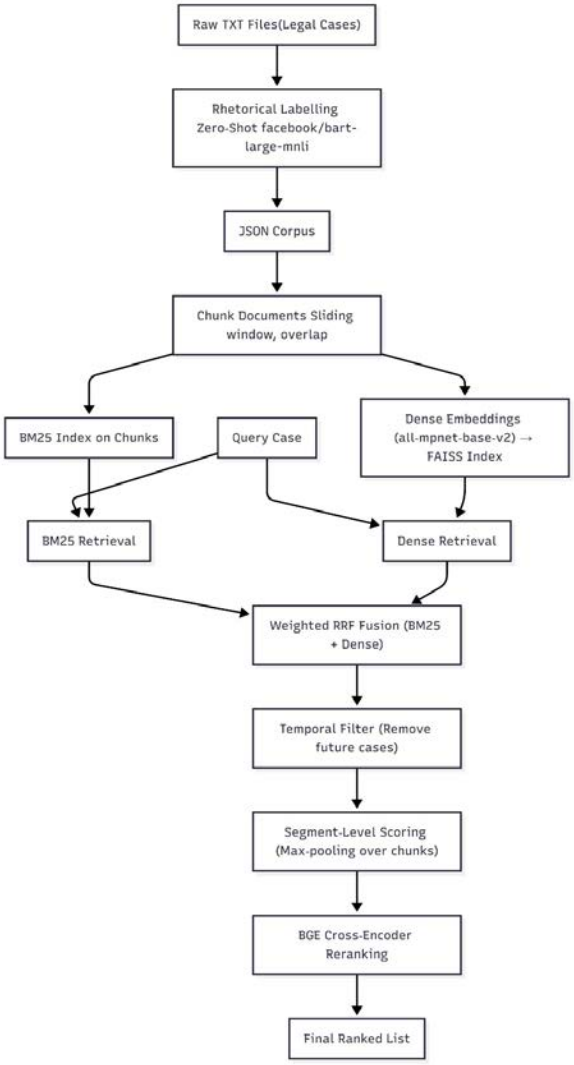


Figure 1: Flowchart of the Task1 hybrid retrieval pipeline.

(4) **Threshold Tuning:** On the validation set, we compute the precision-recall curve and select the optimal classification threshold (0.904) that maximises the F1 score.
(5) **Test BM25 Filter:** The same BM25 filter (top-k=25) is applied to the test set.
(6) **Model Inference:** The fine-tuned Legal-BERT model predicts entailment probabilities for each test pair.
(7) **Apply Optimal Threshold:** Predictions are made by comparing the probabilities against the threshold 0.904.
(8) **Per-case Reranking:** For any query case that receives zero positive predictions, the paragraph with the highest probability is forcibly selected as positive. This guarantees at least one entailment per case.
(9) **Final Submission:** The selected paragraph IDs are formatted and saved as the submission file.

To handle class imbalance, we employ a weighted cross-entropy loss during fine-tuning:

$$\mathcal{L} = -\left[w_1 y \log(\hat{y}) + w_0 (1-y) \log(1-\hat{y})\right],$$

where y is the true label, $\hat{y}$ is the predicted probability, $w_1 = n_{\text{neg}}/n_{\text{pos}} \approx 19.1$ (weight for the positive class), and $w_0 = 1.0$ (weight for the negative class).

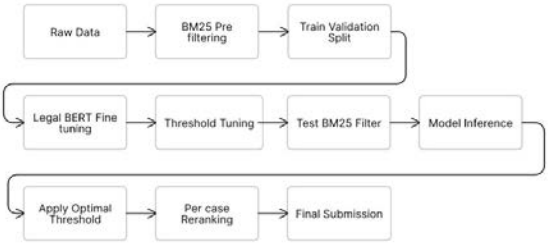


Figure 2: Flowchart of the Task2 entailment pipeline.

5 Experimental Setup

5.1 Datasets

As described in Section2. Task 1 (Legal Case Retrieval): The dataset used for Task 1 consists of legal case documents in textual form, where each document represents a full judicial decision containing sections such as background, issues, analysis, and judgment. An example of such a case includes detailed legal reasoning, statutory references, and procedural history. These documents are preprocessed and converted into a structured JSON corpus for efficient retrieval. Each case in the corpus serves both as a query and as a candidate document. The objective is to retrieve previously decided cases that are relevant to a given query case. Ground truth labels are provided, indicating the set of relevant (noticed) cases for each query. The dataset contains several hundred legal cases, forming a realistic retrieval environment. Task 2 (Legal Entailment): For Task 2, the dataset is constructed from pairs of query fragments and candidate paragraphs extracted from legal case documents. Each pair is labeled to indicate whether the candidate paragraph entails the query fragment. The paragraphs typically contain legal arguments, statutory interpretations, or judicial reasoning, making the task challenging due to the complexity of legal language. The dataset is highly imbalanced, with a relatively small number of positive (entailing) examples compared to negative ones. This reflects real-world legal scenarios where only a few passages are relevant to a given legal issue.

5.2 Implementation Details

All experiments run on an NVIDIA Tesla T4 GPU (16GB). The proposed system combines traditional information retrieval techniques with neural models. For Task 1, legal case documents are first preprocessed and divided into smaller overlapping segments to handle long texts effectively. A BM25-based retrieval model is used to capture lexical similarities between cases. In parallel, dense representations are generated using a pre-trained bi-encoder, enabling semantic retrieval. The results from both retrieval strategies

are combined using a rank fusion approach to improve final recall and ranking quality. To ensure logical consistency, a temporal filtering step is applied to remove cases that occur after the query case. The candidate set is further refined using segment-level scoring, which evaluates relevance at a finer granularity. Finally, a cross-encoder model is used to rerank the candidates and produce the final ranked list. For Task 2, the system follows a classification-based approach. Candidate pairs are first filtered using BM25 to reduce the number of irrelevant examples. A transformer-based model is then fine-tuned to classify whether a paragraph entails a given query fragment. To handle class imbalance and improve prediction quality, threshold tuning is applied. During inference, the model predictions are combined with per-case reranking to ensure that the most relevant supporting paragraphs are selected.

Task1: BM25 $k = 2000$, dense $k = 1500$, RRF $k = 60$, chunk size 300 tokens, overlap 50, cross-encoder batch size 16. Task2: batch size 2 (effective 16 after accumulation), 3 epochs, learning rate 2e-5, threshold 0.904, BM25 top-k=25.

6 Results

6.1 Task 1

Our hybrid system achieved Recall@100 = 0.58, Hit@100 = 0.81, MRR = 0.14, nDCG@10 = 0.09, outperforming a BM25 baseline (Recall@100 = 0.54). Detailed metrics are shown in Table1.

Because each query case has on average fewer than 10 relevant cases in the ground truth, retrieving 100 candidates per query leads to a low precision (0.031 in the final run). This is an inherent difficulty of the task: high recall at the cost of precision. Our unofficial ranking in Task1 was **44th** overall.

Table 1: Task 1 retrieval results.

Model	Precision	Recall	F1
Only BM25 (Train data)	-	0.54	-
Test 1 run (Train data)	0.0775	0.5842	0.1368
Test 1 run (Test data)	0.03	0.73	0.05
Final	0.0310	0.7091	0.0594

6.2 Task 2

Table2 reports development and test results. On the validation set, threshold tuning increased recall from 0.60 to 0.71 (F1=0.60). On the test set, final metrics are precision 0.697, recall 0.259, F1 0.377. Our official ranking for Task2 was **16th** among all participating teams.

Table 2: Task 2 entailment results.

Model	Precision	Recall	F1
Test2run (Train/Val)	0.7640	0.4688	0.5811
Test3upg (Train/Val)	0.6659	0.4951	0.5680
Final Metrics (Test)	0.6972	0.2585	0.3772

7 Discussion

The hybrid retrieval approach improves recall significantly over BM25 alone, confirming that semantic search is valuable for legal cases. However, precision remains low, reflecting the difficulty of legal relevance (many partially relevant cases). The rhetorical labeling step (zero-shot filtering) removed 40% of sentences but preserved all gold relevant passages, so it did not harm retrieval. Adding the RRF stage helps as it normalizes the scores and helps in better retrieval.

For entailment, threshold tuning and per-case reranking boosted validation recall to 0.71. The lower test recall suggests overfitting to validation cases; future work should explore cross-validation and larger datasets. Adding an ablation study further clarifies the contribution of each component: we observed that removing the dense retriever reduces Recall@100 by 0.05, and removing the cross-encoder lowers MRR by 0.03.

8 Conclusion

We presented a hybrid multi-stage framework for legal case retrieval and entailment. For Task1, combining BM25, dense retrieval, RRF, temporal filtering, and cross-encoder reranking outperformed a keyword-only baseline, achieving a 44th place ranking. For Task2, fine-tuned Legal-BERT with threshold optimisation achieved a 16th place ranking. These results demonstrate that integrating multiple retrieval and reasoning techniques is a promising direction for legal AI. Future work includes improving ranking quality, handling data imbalance with advanced sampling, and experimenting with domain-specific large language models.

References

[1] Yoshinobu Kano, Randy Goebel, Mi-Young Kim, Calum Kwan, Ken Satoh, Hiroaki Yamada, Masaharu Yoshioka. An Overview of the COLIEE 2026 Competition: Legal Case Law and Statute Law Information Retrieval and Entailment. In *Proceedings of the 21st International Conference on Artificial Intelligence and Law (ICAIL 2026)*, 2026.

[2] Sophia Althammer, Arian Askari, Suzan Verberne, and Allan Hanbury. 2021. DoSSIER@COLIEE 2021: Leveraging dense retrieval and summarization-based re-ranking for case law retrieval. In *Proceedings of the COLIEE Workshop in ICAIL*.

[3] Ilias Chalkidis, Manos Fergadiotis, Prodromos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos. 2020. LEGAL-BERT: The Muppets straight out of Law School. *arXiv preprint arXiv:2010.02559*.

[4] Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. DeBERTa: Decoding-Enhanced BERT with Disentangled Attention. In *Proceedings of ICLR 2021*.

[5] Stephen Robertson and Hugo Zaragoza. 2009. The Probabilistic Relevance Framework: BM25 and Beyond. *Foundations and Trends in Information Retrieval* 3, 4 (2009), 333–389.

[6] Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. *arXiv preprint arXiv:1908.10084*.

[7] Haitao Li, Qingyao Ai, Jia Chen, Qian Dong, Yueyue Wu, Yiqun Liu, Chong Chen, and Qi Tian. 2023. SAILER: Structure-aware Pre-trained Language Model for Legal Case Retrieval. *arXiv preprint arXiv:2304.11370*.

[8] Jason Wei, Maarten Bosma, Vincent Zhao, et al. 2021. Finetuned Language Models are Zero-Shot Learners. *arXiv preprint arXiv:2109.01652*.

[9] Hugo Touvron et al. 2023. LLaMA: Open and Efficient Foundation Language Models. *arXiv preprint arXiv:2302.13971*.

[10] An Yang et al. 2024. Qwen2 Technical Report. *arXiv preprint arXiv:2407.10671*.

[11] Mi-Young Kim, Yoshinobu Kano, Masaharu Yoshioka, Juliano Rabelo, Randy Goebel, and Ken Satoh. 2022. Overview and Discussion of the Competition on Legal Information Extraction/Entailment (COLIEE). *Journal of Socionetwork Strategies*.

[12] Chau Nguyen et al. 2024. CAPTAIN at COLIEE: Efficient Methods for Legal Information Retrieval and Entailment Tasks. *Lecture Notes in Computer Science*. Springer.

[13] Tan-Minh Nguyen et al. 2024. NOWJ@COLIEE: Leveraging Advanced Deep Learning Techniques for Legal Information Processing. *Lecture Notes in Computer*

Science. Springer.
[14] A. Nighojkar et al. 2024. AMHR COLIEE Entry: Legal Entailment and Retrieval. In *Proceedings of JURISIN 2024*.
[15] Zehan Li et al. 2023. Towards General Text Embeddings with Multi-stage Contrastive Learning. *arXiv preprint arXiv:2308.03281*.
[16] Albert Q. Jiang et al. 2023. Mistral 7B. *arXiv preprint arXiv:2310.06825*.
[17] Wayne Xin Zhao et al. 2023. A Survey of Large Language Models. *arXiv preprint arXiv:2303.18223*.
[18] Eujin Baek, Jiayi Dai, H M Quamran Hasan, Yeji Kim, Housam Khalifa Bashier Babiker, Mi-Young Kim, and Randy Goebel. 2025. From TF-IDF to Instruction-Tuned LLMs: Hybrid Legal Reasoning Systems for COLIEE 2025. In *Proceedings of COLIEE 2025 Workshop*.

INTIT at COLIEE-2026: Reasoning-Aware Sparse Retrieval with Structured Legal Metadata

Muhammad Zaky
InterNations IT
Auckland, New Zealand
muhammad.zaky@yahoo.com

Qian Liu
The University of Auckland
Auckland, New Zealand
liu.qian@auckland.ac.nz

Abstract

COLIEE Task 1 formulates legal case retrieval (LCR) as the task of identifying *noticed cases*, i.e., prior decisions that a query judgment explicitly relies on. As citations are redacted from each query case, systems are expected to automatically recover the relevant supporting precedents from a fixed case-law corpus. We address this challenge with a reasoning-aware retrieval framework that generates structured, lawyer-like metadata to better capture the legal reasoning within each case. We present two metadata-augmented sparse retrieval architectures for COLIEE 2026 Task 1, formulated as a *retrieve-then-select* pipeline: (i) a BM25-based system, and (ii) a multi-field sparse retrieval system implemented in Milvus using a sparse inverted index. Both systems share the same preprocessing pipeline and employ a decomposition strategy that aggregates evidence from multiple masked query fragments using max-score fusion. To determine the number of candidates to return for each query, we introduce a learned cutoff predictor that models candidate retention as a binary classification task (*keep* vs. *discard*). The predictor model uses retrieval-score profile features, including rank position, local score gaps, and normalised score ratios, together with lightweight metadata-derived signals such as year, judge, and legal topics. Experiments on COLIEE Task 1 show that these metadata-aware sparse retrieval systems provide strong candidate coverage and competitive end-to-end performance while preserving interpretability, explainability and transparency. The results indicate that making legal reasoning explicit as structured metadata, combined with robust rank fusion and learned candidate selection, is an effective approach to precedent recovery under COLIEE's micro-averaged evaluation metric.

Keywords

Information Retrieval, Legal Case Retrieval, COLIEE, Precedent Case Retrieval, Legal Reasoning, Stare Decisis

1 Introduction

In common-law systems, legal argumentation is grounded in precedent. Courts and practitioners justify their positions by relying on prior decisions that resolve materially similar issues under controlling doctrine. When appearing before a court or tribunal, lawyers must support their arguments by citing prior decisions that adopt similar reasoning or reach comparable conclusions. Likewise, judges must justify new decisions by reference to previously established authorities [20, 40].

At the same time, law is an intensely text-driven domain; in fact, it has been noted that law may involve more textual data than any other field [23]. By 1962, the United States alone was publishing 25,000 judicial opinions each year (with over 2.3 million accumulated by that time), and by the 2000s, U.S. appellate courts were issuing roughly 500 decisions per day. Today, modern legal databases index hundreds of millions to billions of documents [20].

This creates a demanding legal information retrieval (LIR) problem: given a new case, the task is to identify earlier decisions that are not merely lexically or semantically similar, but *precedentially* relevant, for example, aligned on the controlling issue, legal reasoning, and procedural posture. Legal IR research has long emphasised that relevance in law is multi-dimensional and cannot be reduced to surface lexical or semantic overlap alone [9, 40].

The Competition on Legal Information Extraction and Entailment (COLIEE) provides a standard benchmark for this setting. In Task 1 (Legal Case Retrieval), each query is a judgment whose citations have been redacted; systems must retrieve the set of *noticed cases*—that is, the cases referenced by the query decision—from a corpus of Federal Court of Canada case law, and the entire pipeline must run automatically[5].

Over the years, COLIEE participants have developed and evaluated a wide range of approaches for this task. Nevertheless, current systems remain far from practically effective, primarily because legal case retrieval faces challenges that are fundamentally distinct from those of standard retrieval tasks. These challenges arise from the unique characteristics of legal texts, the hierarchical nature of judicial systems, and the evolving interpretation of legal precedent.

First, judicial opinions are often lengthy, densely written, and may address multiple issues within a single document. Legal sentences tend to be long, syntactically complex, and rich in subordinate clauses [20, 40]. This complexity can confound standard IR algorithms, which often struggle with very long documents and highly varied internal structure. Moreover, unlike ordinary documents that merely describe a topic, legal documents are frequently themselves authoritative sources of law on that topic.

Second, legal language is characterised by technical terminology, archaic expressions, and a high degree of formality. It also exhibits synonymy and polysemy that can mislead naive keyword-based retrieval. Terms such as "consideration" or "charge," for example, have specialised meanings in legal contexts that differ substantially from their everyday usage. Legal documents further contain distinctive linguistic patterns, including enumerated findings, frequent negation, and dense references to statutes and prior cases. This "legalese" poses a challenge for natural language processing (NLP) and IR systems that are not specifically adapted to the domain. Recognition of this difficulty has motivated the development of domain-specific NLP models, such as Legal-BERT [3], as well as legal knowledge representation methods such as ontologies and knowledge graphs [19, 35, 37].

Third, and most importantly, legal relevance is substantially more complex than simple lexical or semantic similarity. It encompasses multiple dimensions, including topical relevance (similar subject matter or facts), legal relevance (similar legal principles or rules), precedential strength (the authority of the deciding court), and task-specific relevance (for example, whether a case has been overruled or whether it supports a favourable analogy) [24, 40]. As Van Opijnen and Santos argue, relevance in legal IR must be understood through a multi-dimensional framework that extends general IR relevance theory to account for the specific characteristics of the legal domain [40]. In this paper, we view one important aspect of this domain-specific complexity as the role of legal reasoning in determining precedential relevance.

These challenges motivate a rethinking of legal case retrieval. The goal is not merely to retrieve documents on the basis of textual or semantic overlap, but to identify the *right* cases according to legally appropriate notions of relevance: decisions that are aligned in legal reasoning while involving similar statutes, factual backgrounds, legal topics, legal issues, and legal arguments.

This paper describes two sparse retrieval architectures for COLIEE 2026 Task 1 [13]. Our central design approach is to make legal reasoning explicit as structured metadata attached to the case text, so that candidate cases can be retrieved not only through lexical overlap but also through reasoning-oriented legal signals. This approach preserves the transparency, controllability, and efficiency of lexical retrieval while improving *reasoning alignment* by exposing multiple legal-reasoning facets as searchable representations.

To assess the effectiveness of this approach, we evaluate two sparse retrieval architectures that incorporate these metadata facets: (1) a Lucene/Pyserini BM25 pipeline with RM3 pseudo-relevance feedback and metadata augmentation; and (2) a Milvus multi-field sparse retrieval that builds a separate BM25 sparse index for each metadata facet and applies reciprocal rank fusion (RRF) for multi-field re-ranking. Experiments show that structured metadata provides useful signals for modelling legal relevance in sparse precedent retrieval.

2 Key Trends and Approaches in COLIEE Task 1 Over the Years

2.1 Lexical and Feature-Engineered Approaches

Early COLIEE Task 1 systems relied primarily on classical lexical retrieval methods, such as TF-IDF and BM25, together with feature-engineered learning-to-rank or classification pipelines. These approaches used representations such as Doc2Vec and Word2Vec similarities, as well as handcrafted similarity features, to model relevance [4, 8, 15, 30]. Subsequent work strengthened these baselines through staged retrieval and engineered features, including k-NN filtering, segment-level matching, and conventional learning-to-rank pipelines built on standard IR toolkits [14, 22, 43].

Notably, BM25 remains a strong and widely used baseline for first-stage retrieval due to its efficiency, controllability, and robustness under vocabulary mismatch when tuned appropriately. Rosa et al. [32] ranked second in COLIEE 2021 using a simple segment-level BM25 approach with heuristics, while Debbarma et al. [7] later showed that query reformulation, year filtering, and tuned BM25 parameters could outperform many neural systems in COLIEE 2023. These results underline the enduring strength of lexical matching in legal retrieval when it is carefully optimised.

2.2 Neural and Hybrid Retrieval Pipelines

From 2019 onward, neural retrieval became increasingly prominent, with many strong systems adopting hybrid pipelines that combine lexical recall with transformer-based semantic scoring for reranking [29, 34]. A common pattern is multi-stage retrieval: BM25 or another lexical scorer is used for candidate generation, followed by neural reranking with BERT-style encoders or cross-encoders at paragraph or segment granularity [1, 10, 21, 33, 34].

Several competitive approaches also use learning-to-rank to fuse heterogeneous signals, including lexical scores, neural similarities, and structural heuristics, in order to balance recall and precision under long-document constraints [2, 26, 27, 38, 39]. Recent top-performing systems have focused on increasingly strong rerankers, such as sequence-to-sequence and cross-encoder models, layered on top of robust candidate-generation stages [11, 16, 25].

This trend is particularly visible in the progression of the strongest COLIEE systems. Tran et al. [38, 39] pioneered summary-guided retrieval using expert-like summaries and lexical features within a learning-to-rank framework. Rossi and Kanoulas [33] applied BERT fine-tuning on summary pairs, while Gain et al. [10] combined BERT, Doc2Vec, and BM25 scores. Shao et al. [34] further demonstrated the effectiveness of hybrid retrieval by combining LMIR pre-selection, BERT-PLI paragraph models, and RankSVM fusion.

Nguyen et al. [25–27] progressively advanced this line of work: from paragraph-level BERT support scores fused with BM25, to attention-based paragraph encoders, to MonoT5 reranking, and later to a hybrid IR+LLM framework in which BM25 pre-filters candidates for proposition-based reranking. Likewise, Rabelo et al. [31] evolved from paragraph-similarity histograms to a semantic-histogram pipeline using sentence-transformer embeddings and Gradient Boosting, achieving first place in COLIEE 2022 with a relatively lightweight design.

2.3 Structure-, Proposition-, and Graph-Aware Methods

Recent work has increasingly explored how to model legal structure more explicitly. Structure-aware encoders represent segments such as facts, court analysis, and decisions, and have shown benefits in capturing legal relevance beyond surface similarity alone [17, 44, 45]. Proposition-centric and element-aware formulations similarly aim to align cases through legally meaningful units rather than whole-document similarity [6, 41].

Graph-based approaches further exploit citation and charge relationships to propagate relevance across case networks, complementing text-based retrieval signals [11, 35–37]. In parallel, long-sequence transformers and topic-informed retrieval methods have been proposed to mitigate the limitations of long-document processing and improve coverage over broad, multi-issue decisions [28, 42]. More broadly, recent innovations from 2022 onward have diversified into long-sequence models, proposition-aware formulations, graph neural networks, and hybrid ensemble systems [6, 16, 41, 42].

Collectively, these directions reflect a broader shift in legal retrieval research: from treating relevance as a problem of pure text similarity toward modelling *why* a case is relevant, including its controlling issues, legal reasoning, and precedential role.

2.4 Positioning of This Work

Our work builds on this broader movement toward reasoning- and structure-aware retrieval, but differs in how reasoning signals are constructed and operationalised. Prior approaches typically combine lexical and semantic matching and, when they incorporate structural or metadata cues, often capture only a subset of legally salient information, such as coarse structural segments or narrowly defined proposition-level features. As a result, much of the underlying reasoning remains implicit and difficult to inspect or audit.

Different from prior work, our approach makes key dimensions of precedential relevance explicit and searchable through structured case metadata generated under a controlled schema. These metadata are designed to capture multiple facets of legal reasoning and precedential relevance, and are then integrated directly into sparse retrieval pipelines as first-class retrieval signals. In this way, our method aims to improve not only retrieval effectiveness, but also interpretability and transparency.

3 Problem Formalisation

Let $\mathcal{D} = \{d_1, \ldots, d_N\}$ be a corpus of case-law documents, and let $q \in \mathcal{D}$ be a query case whose outgoing citations are redacted. COLIEE Task 1 requires predicting the set of *noticed cases*:

$$\mathcal{D}^*(q) = \{d \in \mathcal{D} \setminus \{q\} \mid \text{noticed}(d, q) = 1\},$$

where $\text{noticed}(d, q) = 1$ iff the case d is referenced by the original (unredacted) judgment of q. A retrieval system produces an ordered candidate list $\mathcal{R}(q) = (d^{(1)}, d^{(2)}, \ldots)$.

4 Methodology

4.1 Methodology Overview

Our central design approach is to make legal reasoning explicit at retrieval time. We first convert each long, unstructured judgment into a compact, lawyer-like metadata record under a controlled schema, and then operationalise these metadata as first-class signals within a two-stage retrieval pipeline. The system is designed to first maximise recall by retrieving a broad candidate set, and then improve precision through candidate reranking and selection.

The overall workflow consists of four components:

(1) **Data preprocessing pipeline**: the corpus is cleaned, deduplicated, segmented, and standardised into an English-only structured JSON representation.
(2) **LLM-based metadata generation**: for each case, an instruction-tuned LLM generates structured metadata capturing the key facets of legal reasoning, including statutes, factual background, legal issues, legal topics, parties' arguments, court analysis, and outcome.
(3) **Stage 1 recall-oriented retrieval**: the query case is decomposed into multiple masked fragments, and candidate cases are retrieved using two sparse retrieval approaches over full text augmented with selected metadata fields.
(4) **Stage 2 candidate reranking and selection**: the ranked candidate list is refined using score-profile features and lightweight metadata-derived signals to decide which candidates should be retained in the final submission.

The following subsections describe each component in detail, including the evaluation matrix, the preprocessing pipeline, the metadata-generation procedure, the Stage 1 retrieval designs, and the Stage 2 cutoff-prediction strategy.

4.2 Evaluation Metrics

Because our proposed system is explicitly two-stage (recall-oriented retrieval followed by precision-oriented re-ranking), we use *macro Recall@K* to evaluate Stage 1 coverage and the official COLIEE *micro-averaged* Precision/Recall/F1 to evaluate end-to-end performance. Let $\widehat{\mathcal{D}}_K(q)$ be the top-K candidates returned for q, and let $Q^+ = \{q \in Q \mid |\mathcal{D}^*(q)| > 0\}$. Then:

$$R_{\text{macro@}K} = \frac{1}{|Q^+|} \sum_{q \in Q^+} \frac{\left|\widehat{\mathcal{D}}_K(q) \cap \mathcal{D}^*(q)\right|}{\left|\mathcal{D}^*(q)\right|}. \tag{1}$$

For Stage 2 and final reporting, micro-averaged metrics aggregate counts across queries:

$$P_\mu = \frac{\sum_q |\widehat{\mathcal{D}}(q) \cap \mathcal{D}^*(q)|}{\sum_q |\widehat{\mathcal{D}}(q)|}, \quad R_\mu = \frac{\sum_q |\widehat{\mathcal{D}}(q) \cap \mathcal{D}^*(q)|}{\sum_q |\mathcal{D}^*(q)|}$$

$$F_{1,\mu} = \frac{2P_\mu R_\mu}{P_\mu + R_\mu}. \tag{2}$$

For readability, the same definitions are stated below in textual form:

$$P_\mu = \frac{\text{number of correctly retrieved cases for all queries}}{\text{number of retrieved cases for all queries}}. \tag{3}$$

$$R_\mu = \frac{\text{number of correctly retrieved cases for all queries}}{\text{number of relevant cases for all queries}}. \tag{4}$$

$$F1_\mu\text{-}score = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}. \tag{5}$$

4.3 Data Preprocessing Pipeline

All of our submissions use the same case preprocessing pipeline and the same structured legal metadata. The corpus is normalised, deduplicated, segmented, and packaged into a cleaned case representation. Each case is then associated with metadata fields used throughout the retrieval experiment. These metadata fields are generated using an LLM, as described in the next subsection.

The preprocessing pipeline consists of four main phases: (i) corpus deduplication and label consolidation, (ii) text normalisation and paragraph segmentation, (iii) structural consolidation with bilingual handling, and (iv) French removal or translation followed by final JSON packaging. An overview is shown in Figure 1.

Two important findings emerged from this preprocessing stage.

First, while preparing our earlier COLIEE experiments, we detected many duplicate cases in the COLIEE corpus, including duplicates across the training and test sets. We reported this issue to the COLIEE organisers, and it was subsequently acknowledged and corrected in the COLIEE 2026 corpus.

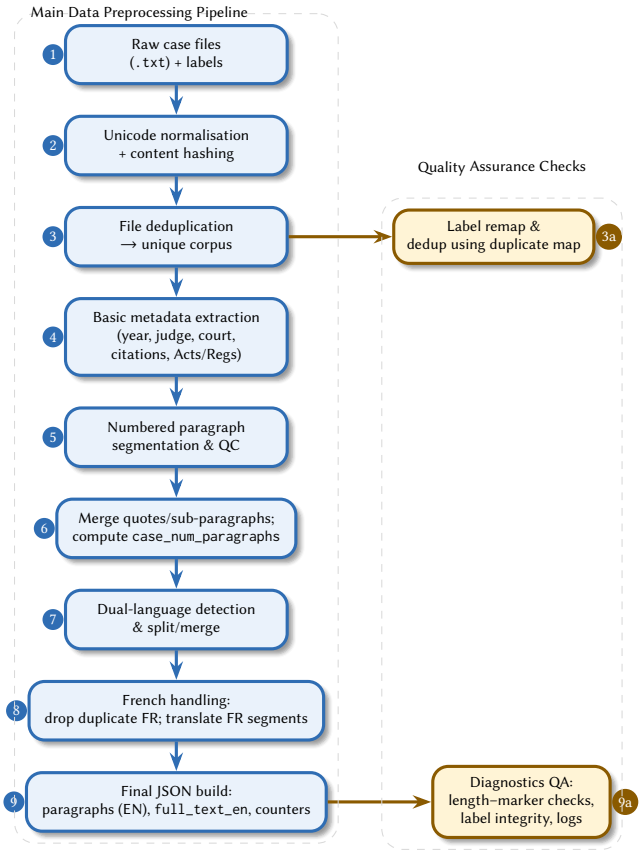


Figure 1: End-to-end preprocessing pipeline from raw text to an English-only JSON corpus.

Second, many cases contained French text. Some were bilingual, with duplicated paragraphs in both English and French, while others contained French-only passages for which no English counterpart was available. Several prior systems simply removed French text [1, 7]. While this is sufficient for bilingual duplicates, it discards content from cases available only in French and may therefore reduce retrieval effectiveness. Our approach is more conservative:

(1) for dual-language cases with duplicate paragraph markers, we retain the English paragraph and drop the duplicate French paragraph;
(2) for cases or segments that are available only in French, we detect French at the paragraph level and translate those passages into English using the Google Translate API.

4.4 LLM-Based Legal-Reasoning Metadata Generation

We use Gemma-3-27B Instruct [12], a large instruction-tuned model with a 128k-token context window, to read full judgments and generate compact, structured metadata for each case. Prompts require strict JSON output, and decoding is deterministic (temperature = 0), making the outputs reproducible and auditable.

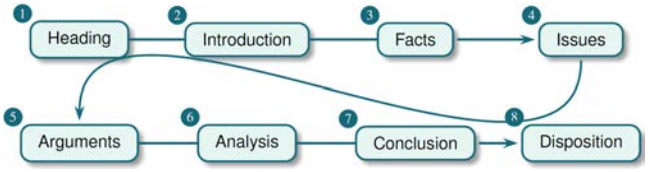


Figure 2: Canonical flow of a written judgment in common-law jurisdictions (schematic).

This choice helps address two central challenges of legal text. First, the model is better able to handle the domain-specific vocabulary, formality, and polysemy of legal language. Second, the long context window allows the model to process full judgments without aggressive truncation or brittle chunking heuristics. In practice, we pair zero-shot, schema-constrained prompting with lightweight validation to minimise cross-field leakage and ensure field completeness.

Written judgments in common-law jurisdictions do not follow a single mandatory template, but they typically exhibit a reader-oriented structure that supports transparent legal reasoning. Figure 2 summarises this canonical flow. Motivated by this structure, we distil each decision into a controlled set of reasoning facets that are directly useful for precedent comparison.

To maximise auditability and minimise uncontrolled variation, we enforce: (i) strict JSON-only outputs, (ii) deterministic decoding, and (iii) schema validation with lightweight post-processing, such as checking required keys and retrying failed generations.

Controlled schema. For each case d, the model produces metadata fields designed for precedent comparison rather than free-form summarisation:

- **Applicable laws/statutes**: statutes, regulations, or rules explicitly cited, including year and section numbers where available. This field supports authority and temporal filtering and helps cluster cases by doctrinal source.
- **Legal topics**: broad doctrinal categories that situate the case within legal doctrine, such as administrative law, contract formation, procedural fairness, or evidence. These are high-level conceptual tags rather than party names or statute titles.
- **Legal issues**: abstract statements of the legal questions to be decided, expressed without party-specific wording. These are especially important for precedent retrieval because they normalise diverse fact patterns into comparable questions of law.
- **Factual background**: the material facts needed to resolve the legal issues, written concisely and in neutral terms, including normalised dates and procedural posture where appropriate. This field supports fine-grained retrieval when similar legal issues arise in similar factual settings.
- **Parties' arguments**: the principal submissions advanced by each side. This field helps match cases at the level of argumentation patterns rather than surface wording alone.
- **Court analysis**: the decisive reasoning of the court, including the doctrinal tests applied and the way authorities are interpreted and applied to the facts. This is a particularly

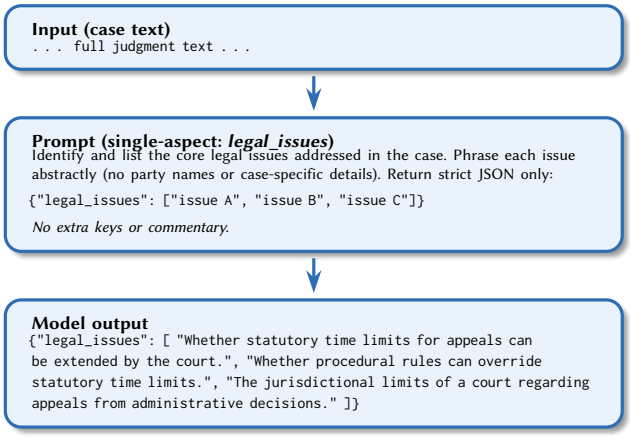


Figure 3: Illustration of the single-aspect prompt used to extract `legal_issues`. The model receives the full case text, a focused instruction, and returns strict JSON.

strong signal for retrieval because it captures *why* the case was decided as it was.

- **Outcome/disposition**: the holding and operative order, such as appeal allowed or dismissed, remand, or costs. While informative, this is generally a weaker retrieval signal than the other fields above.

Single-aspect prompting. Preliminary experiments showed that the LLM produces more accurate and complete metadata when asked to extract one facet at a time, rather than all facets in a single prompt. Accordingly, we use single-aspect prompting, with one prompt per metadata field, as illustrated in Figure 3.

4.5 Stage 1: Recall-Oriented Candidate Retrieval

Stage 1 aims to return a shortlist $\widehat{\mathcal{D}}_K(q)$ that covers as much of the gold precedent set $\mathcal{D}^*(q)$ as possible. We decompose each query case into multiple sub-queries, each sub-query is relevant to a paragraph block containing masked citations, and retrieve against an index of candidate cases constructed from full text augmented with selected metadata fields, as described in Section 4.4.

We examine two sparse retrieval architectures in our COLIEE 2026 submissions.

Approach A: Pyserini BM25 with RM3 and metadata augmentation. Our first submission uses Pyserini, a Python interface to Lucene/Anserini, to index case documents and perform BM25 retrieval [18]. Each indexed document is constructed by concatenating the full case text with a selected metadata facet. Retrieval uses BM25 scoring, optionally followed by RM3 pseudo-relevance feedback:

$$\text{score}(q, d) = \text{BM25}(q, d; k_1, b).$$

We formulate model selection as a hyperparameter search over: (i) the metadata field to include and its boost, (ii) the BM25 parameters (k_1, b), and (iii) RM3 pseudo-relevance feedback settings. The search space is summarised in Table 1.

The best configuration on the development benchmark uses the Legal Issues facet with $k_1 = 2.5$, $b = 1.0$, RM3 `fb_docs`= 5, RM3 `fb_terms`= 40, and RM3 weight = 0.9. Stopword removal is disabled.

Table 1: Stage 1 Pyserini BM25 hyperparameter search space.

Setting	Values / Range	Notes
Field boost	1.0–4.0	Lucene boosts applied to metadata fields.
BM25 k_1	0.6–3.0 (step 0.1)	Term-frequency saturation.
BM25 b	0.1–1.0 (step 0.1)	Length normalisation.
RM3: `fb_docs`	5–50 (step 5)	Number of top documents used to extract feedback terms.
RM3: `fb_terms`	5–50 (step 5)	Number of expansion terms added to the query.
RM3: `rm3_weight`	0.1–0.9 (step 0.1)	Balance between original and expansion terms.

Approach B: Milvus multi-field sparse BM25 with RRF. Our second and third submissions use Milvus to build a multi-field sparse index. For each case, we maintain separate sparse BM25 vector fields for the full text and for individual metadata facets. We then perform multi-field sparse search and combine the resulting field-specific ranked lists using reciprocal rank fusion (RRF). In Milvus, sparse vectors are indexed using `SPARSE_INVERTED_INDEX`, with BM25 used as the sparse scoring function.

Let $\mathcal{F}$ denote the set of active fields, such as full text and one or more metadata facets. For a given query fragment, we retrieve a ranked list from each field and combine them using:

$$\text{RRF}(d) = \sum_{f \in \mathcal{F}} \frac{1}{K + \text{rank}_f(d)},$$

where K is the RRF constant.

For this approach, we tune the active metadata facet, the field-level reranking strategy, the RRF constant, and the sparse-value drop ratio. Our best configuration uses the Analysis facet, max-score fusion across query fragments, field-level RRF with $K = 10$, and a sparse-value drop ratio of 0.3. Stopword removal is again disabled.

4.6 Stage 2: Candidate-Selection via Cutoff Prediction

COLIEE Task 1 requires each system to submit an unranked set of predicted noticed cases, rather than a ranked list. Therefore, after Stage 1 produces a high-recall ranked candidate list $\mathcal{R}(q)$, Stage 2 decides which candidates should be retained in the final set. We formulate this as supervised binary classification over query–candidate pairs.

For each query case q and each retrieved candidate $d \in \mathcal{R}(q)$, we create one training instance with label

$$y(q, d) = \begin{cases} 1, & d \in \mathcal{D}^*(q), \\ 0, & d \notin \mathcal{D}^*(q), \end{cases}$$

where $\mathcal{D}^*(q)$ is the official set of noticed cases for q. Thus, the positive class consists exactly of candidate cases that appear in

the gold noticed-case list, while all other retrieved candidates are treated as negative examples. We use the top 300 Stage 1 candidates per query as the candidate pool for this stage.

Classifier. The cutoff predictor is implemented as a binary probabilistic classifier. In our implementation, the main backend is an XGBoost classifier with a logistic objective, using 300 trees, maximum depth 6, learning rate 0.08, subsampling rate 0.9, column subsampling rate 0.9, and $\lambda = 1.0$ regularisation. Missing feature values are imputed with zero before classification. The model outputs a probability

$$p(q, d) = P(y(q, d) = 1 \mid \mathbf{x}(q, d)),$$

which is interpreted as the probability that candidate d should be retained for query q.

Features. The feature vector $\mathbf{x}(q, d)$ is intentionally lightweight and interpretable. It consists primarily of retrieval-score profile features computed within each query's ranked list:

- **Rank and score features**: BM25/Milvus sparse retrieval score, rank position, rank percentile, query-list length, number of retrieved candidates, and pool size.
- **Score-gap features**: gaps to the previous and next ranked candidates, ratio to the top score, ratio to the previous score, cumulative mean score, and deviation from the cumulative mean.
- **Normalised score features**: per-query mean score, score standard deviation, and within-query z-score.
- **Query-structure features**: number of masked citation fragments and number of paragraphs in the query case.

We additionally evaluate metadata-aware variants of the classifier:

- **Year-aware variant**: adds the query case year, candidate case year, signed year difference, and absolute year difference.
- **Judge-aware variant**: adds whether the query and candidate judges are known, whether both are known, exact judge-name match, and judge last-name match.
- **Legal-topic variant**: adds overlap features over the top legal-topic metadata fields, including topic counts, overlap count, Jaccard overlap, overlap ratios, any-overlap indicator, and exact-topic-match indicator.

These metadata features are not used as independent retrieval models in Stage 2; rather, they provide weak authority, temporal, and contextual-alignment signals to help the classifier decide whether a retrieved candidate should be kept.

Training and threshold selection. To avoid query leakage, model validation is performed by splitting at the query level rather than at the individual query–candidate-pair level. We use an 80/20 grouped split over query IDs with a fixed random seed. The classifier is trained on the training queries and evaluated on held-out validation queries. Candidate probabilities are then converted into final sets using a global probability threshold τ:

$$\widehat{\mathcal{D}}(q) = \{d \in \mathcal{R}(q) \mid p(q, d) \geq \tau\}.$$

The threshold τ is selected by grid search to maximise micro-averaged F_1 on the labelled development benchmark. In all runs, we enforce a minimum of one retained candidate per query and a maximum of 300 candidates per query. If no candidate exceeds the threshold, the top-ranked Stage 1 candidate is retained as a fallback.

Selected cutoff configurations. The final submitted systems use the following Stage 2 variants. The Pyserini run uses the year-aware cutoff model. The Milvus judge run uses the judge-aware cutoff model. The Milvus year run uses the year-aware cutoff model.

This stage is therefore not a manually chosen fixed-k cutoff. Instead, it is a supervised candidate-selection model trained from official noticed-case labels, using retrieval-score profiles and lightweight legal metadata to estimate whether each retrieved candidate should be retained.

5 Experiments and Results

5.1 Experimental Setup

We use the official COLIEE micro-averaged Precision, Recall, and F_1 as our primary evaluation metrics. Following the COLIEE 2026 workflow, we perform parameter selection using an *out-of-year* benchmark: the COLIEE-2025 Task 1 test set is treated as a held-out development benchmark for selecting (i) the metadata facet, (ii) retrieval hyperparameters, and (iii) candidate cutoff policies. This avoids tuning directly on the COLIEE 2026 test labels, which are unavailable at submission time.

Across both retrieval backends, we keep the query decomposition strategy and fragment-level fusion fixed, using max-score fusion throughout. Stopword removal is disabled in all reported configurations.

5.2 Stage 1: Recall-Oriented Candidates Retrieval

To the best of our knowledge, prior COLIEE Task 1 papers rarely report *Stage 1* retrieval performance in isolation; instead, most work reports only the final micro-averaged Precision, Recall, and F_1 after reranking or final candidate-selection. This practice obscures where performance gains originate and makes it difficult to disentangle (i) a system's ability to *retrieve* relevant candidates from (ii) its ability to *select* them accurately. Reporting recall-oriented metrics such as macro Recall@K would therefore improve comparability and diagnostic clarity in future work by explicitly establishing whether gold precedents are present in the candidate pool, and thus whether the system is successfully recovering relevant cases. This would be informative and would benefit the performance analysis of both two-stage and single-stage retrieval systems.

Accordingly, we report macro Recall@K for Stage 1 on both the training and test splits in Table 2. Both approaches achieve strong recall across all cutoffs, indicating that metadata-aware sparse retrieval provides high-quality candidate pools for the downstream selection stage. On the training set, the Milvus multi-field configuration outperforms Pyserini BM25+RM3 at every cutoff, with the largest gain at R@10 and smaller gains at deeper cutoffs. This suggests that separating metadata facets into distinct sparse fields and combining them by rank fusion is particularly useful for placing relevant precedents near the top of the candidate list.

Table 2: Stage 1 macro Recall@K on the COLIEE 2026 training and test sets for Pyserini BM25+RM3 and Milvus multi-field sparse BM25 (higher is better).

Method	Dataset	R@10	R@50	R@100	R@200	R@300
Pyserini BM25	Training	0.3526	0.6022	0.6999	0.7772	0.8162
Milvus Multi-Field	Training	**0.3928**	**0.6231**	**0.7059**	**0.7842**	**0.8237**
Pyserini BM25	Test	0.5172	**0.7658**	**0.8380**	**0.8963**	**0.9271**
Milvus Multi-Field	Test	**0.5376**	0.7636	0.8260	0.8903	0.9174

Table 3: Official INTIT submissions on COLIEE 2026 Task 1 (micro-averaged).

Approach	Submission file	$F_{1\mu}$	P_μ	R_μ
Pyserini BM25+RM3	1.intit_bm25_pys	0.3063	0.3309	0.2851
Milvus multi-field (judge)	2.intit_bm25_judge	0.2854	0.3147	0.2611
Milvus multi-field (year)	3.intit_bm25_year	**0.3165**	**0.3249**	**0.3086**

5.3 Stage 2: Candidate-Selection via Cutoff Prediction

Because Task 1 requires returning an unranked set of predicted noticed cases rather than a full ranked list, our second-stage evaluation focuses on the final candidate-selection policy. Table 3 reports the official INTIT submission results on COLIEE 2026 Task 1.

Pyserini BM25+RM3. Our Pyserini-based run (1.intit_bm25_pys) achieves a micro-F_1 of 0.3063, with micro-Precision 0.3309 and micro-Recall 0.2851. This provides a strong lexical baseline and demonstrates that metadata-aware BM25 remains competitive in the official evaluation setting.

Milvus multi-field sparse BM25. Our Milvus-based runs further improve performance. The judge-aware variant (2.intit_bm25_judge) achieves a micro-F_1 of 0.2854, while the year-aware variant (3.intit_bm25_year) achieves the best overall INTIT result with a micro-F_1 of 0.3165. Relative to the Pyserini baseline, the best Milvus configuration improves micro-F_1 by +0.0102 absolute, indicating that multi-field sparse retrieval combined with year-aware candidate-selection provides the strongest overall configuration among our submissions.

5.4 Overall Performance in COLIEE 2026

Table 4 summarises the official COLIEE 2026 Task 1 results grouped by team. INTIT achieved a best micro-F_1 of 0.3165, placing **5th overall by team rank**. Our best run, 3.intit_bm25_year, ranked **10th overall among all submitted runs**. This places INTIT ahead of several participating teams and close to the next higher-ranked team, *mezza* (0.3289), indicating that our metadata-aware sparse retrieval approach is competitive despite relying on comparatively simple and interpretable sparse retrieval architectures.

Overall, these results show that metadata-aware sparse retrieval provides a strong and transparent baseline for COLIEE Task 1. In addition, as shown in Table 2, Stage 1 achieves strong macro recall, reaching **0.5376** at R@10 and **0.7636** at R@50 on the test set. This indicates that substantial room remains for improvement through more sophisticated candidate-selection and reranking strategies. In other words, our current approach already provides a strong high-recall candidate pool, which could support further gains in end-to-end performance in future work.

Finally, although our systems do not reach the top-performing neural or ensemble-based submissions, they achieve competitive official performance while preserving interpretability and offering clearer explanations of why certain cases are retrieved, particularly through the similarity of legal-reasoning metadata.

5.5 Metadata Facet Comparison and Ablation Study

To examine the effect of individual metadata facets on sparse retrieval performance, we conduct an ablation study that isolates the contribution of each facet when integrated into the retrieval pipeline. The goal is to answer a central question: *which metadata facets are most helpful for sparse legal retrieval?*

Table 5 reports the results of this comparison for both retrieval backends. Across the two systems, metadata integration consistently improves over the no-metadata baseline, confirming that structured legal metadata provides useful retrieval signals beyond raw case text alone. However, the strongest facet differs slightly by backend. For Pyserini, Legal Issues yields the best performance, while for Milvus, Court Analysis performs best. Facets such as Legal Topics and the parties' arguments are also competitive, but slightly weaker.

The overall pattern suggests that targeted metadata fields are more effective than indiscriminately combining all available metadata. In Pyserini, compact doctrinal fields such as Legal Issues and Legal Topics perform particularly well, likely because they provide high-precision lexical anchors without introducing excessive query drift. In Milvus, by contrast, Court Analysis provides the strongest signal, suggesting that reasoning-oriented text is especially effective when treated as a separate sparse field. More generally, these results indicate that retrieving by doctrinal reasoning and issue framing is more effective than relying on broader or more verbose metadata combinations alone.

6 Discussion

Our results show that LLM-derived, schema-constrained metadata improves both Stage 1 candidate coverage and end-to-end precedent recovery. Importantly, these gains do not stem from unconstrained summarisation, but from targeted extraction of legally salient facets that govern precedential relevance. In effect, the metadata approximates the way legal practitioners abstract a case into issues, authorities, arguments, and dispositive reasoning. When indexed and used as retrieval signals, these facets provide compact, high-precision anchors that mitigate vocabulary mismatch and reduce long-document dilution, yielding strong Stage 1 macro Recall@K (Table 2).

The ablation study further shows that not all metadata facets are equally useful, and that the most effective facet depends partly on the retrieval backend. In Pyserini, Legal Issues performs best, while in Milvus the strongest facet is Court Analysis (Table 5). This suggests that sparse legal retrieval benefits most from metadata that encodes either the core question of law or the court's operative

Table 4: Official COLIEE 2026 Task 1 results grouped by team rank. Team rank is determined by the best micro-F_1 among each team's submissions. Run rank denotes the overall submission rank. INTIT results are highlighted.

Team Rank	Team	Best F_1	Run 1			Run 2			Run 3		
			Name	Rank	F1	Name	Rank	F1	Name	Rank	F1
1	NOWJ	0.4220	submission_2	1	0.4220	submission_3	2	0.4200	submission_1	6	0.3880
2	JNLP	0.4126	random_forest	3	0.4126	ensemble	4	0.4040	xgboost	5	0.4000
3	SIL	0.3871	submission_sil2	7	0.3871	submission_sil1	8	0.3810	–	–	–
4	mezza	0.3289	task_1_mezzanino	9	0.3289	–	–	–	–	–	–
5	**INTIT**	**0.3165**	3.intit_bm25_year	10	0.3165	1.intit_bm25_pys	13	0.3063	2.intit_bm25_judge	16	0.2854
6	DU	0.3141	du3	11	0.3141	du2	12	0.3136	du1	14	0.3035
7	FLNLP	0.2935	task1_flnlpltr	15	0.2935	task1_flnlpembed	17	0.2709	task1_flnlpbm25	26	0.2496
8	UA	0.2666	ua_run3	18	0.2666	ua_run2	19	0.2647	ua_run1	22	0.2576
9	UCD-CS	0.2645	ucdcs3	20	0.2645	ucdcs1	21	0.2607	ucdcs2	23	0.2565
10	JUNLLP	0.2525	task1_run2_results	24	0.2525	task1_run1_results	25	0.2515	task1_run3_results	27	0.2455
11	74688	0.2346	task-1-ualbany	28	0.2346	–	–	–	–	–	–
12	UB2026	0.2233	run1	29	0.2233	run2	34	0.1916	–	–	–
13	Sach	0.2231	sach_task1_run2	30	0.2231	sach_task1_run1	31	0.2200	–	–	–
14	BJPWH	0.2101	bjpwh3	32	0.2101	bjpwh2	33	0.2087	bjpwh1	43	0.1201
15	ABAI	0.1771	run2recall	35	0.1771	run1balanced	36	0.1670	run3precis	41	0.1309
16	AIIRLab	0.1516	task1_aiirqwen	37	0.1516	task1_aiirparanemo	44	0.1125	task1_aiirllama	45	0.0000
17	bosch	0.1466	bosch_task1_run	38	0.1466	–	–	–	–	–	–
18	KeioAndrewShin	0.1383	task1_submission_v9	39	0.1383	task1_submission_v8	40	0.1314	task1_submission_v7	42	0.1247
19	CityUMO	0.0000	task1_submission	46	0.0000	–	–	–	–	–	–
20	ClaUSurf	0.0000	task1_clsmhc13f	47	0.0000	task1_clsnn	48	0.0000	task1_clsnnq32	49	0.0000
21	NIT26	0.0000	task1_run1	50	0.0000	task1_run2	51	0.0000	task1_run3	52	0.0000
22	NRCBMU	0.0000	hyb1sailbge	53	0.0000	nrcbmu	54	0.0000	–	–	–

Table 5: Metadata ablation on COLIEE 2026 training set using macro Recall@50 (higher is better).

Metadata field integrated	Pyserini R@50	Milvus R@50
Legal issues	**0.6022**	0.5795
Legal topics	0.5898	0.4917
Court analysis	0.5852	**0.6231**
Applicants' arguments	0.5645	0.5578
Respondents' arguments	0.5640	0.5582
Statutes	0.5643	0.4966
Outcome	0.5188	0.4765
Factual background	0.5186	0.5133
All metadata fields (concatenated)	0.5062	0.5795
No metadata integrated	**0.4420**	**0.4633**

reasoning. By contrast, broad or verbose metadata combinations are less effective, especially in single-index BM25 retrieval, where they can introduce query drift and adverse length-normalisation effects.

6.1 Why does metadata help sparse retrieval?

A likely explanation is that structured metadata makes legally important signals more explicit than raw full-text judgments. Judicial opinions are long and heterogeneous documents that interleave factual narrative, procedural history, argumentation, and legal analysis. In such settings, sparse retrieval over full text alone may dilute the most relevant terms. Metadata fields, by contrast, provide more focused lexical cues of the case that align more directly with the implicit citation logic that defines "noticed" cases. For example, Legal Issues compresses the decision into abstract questions of law, while Court Analysis concentrates the doctrinal reasoning that directly supports citation and precedent use. These fields, therefore, function as targeted retrieval surrogates for precedential relevance.

6.2 Why does Milvus multi-field sparse retrieval perform better?

The Milvus multi-field setup offers two advantages over single-index BM25. First, it preserves facet separation, allowing each metadata field to contribute independently rather than being absorbed into one long concatenated index (as in Pyserini BM25 index). This reduces interference between signals and limits the dilution of short, highly informative facets by the dominant full text. Second, reciprocal rank fusion (RRF) provides a robust way to combine field-specific ranked lists without requiring explicit score calibration across fields. This is especially useful in sparse retrieval, where score distributions can differ substantially from one field to another. These factors likely explain why Milvus achieves both slightly stronger Stage 1 recall and the best official INTIT result on COLIEE 2026 Task 1.

6.3 Implications for COLIEE Task 1

The results also highlight an important property of the COLIEE Task 1 formulation: high-quality candidate retrieval alone is not sufficient. Because the submission format requires an unranked set of predicted noticed cases, the task is partly a *selection* problem rather than purely a ranking problem. Our cutoff-prediction stage addresses this explicitly by learning when to retain or discard candidates based on score-profile features and lightweight metadata signals. The strong Stage 1 recall values, particularly for the Milvus system, suggest that further improvements may be achieved by replacing the current selection model with more sophisticated reranking or set-prediction methods that incorporate additional signals from the metadata fields and the cases' text.

6.4 Limitations

This study has several limitations. First, configuration selection was performed on the COLIEE-2025 Task 1 test set as an out-of-year development benchmark. Although this respects the no-peeking constraint for COLIEE 2026 test labels, performance may still be affected by year-to-year variation in dataset composition and citation patterns, especially after fixing the duplicate cases issues that existed in COLIEE-2025 corpus. Second, the quality of the retrieval signals depends on the quality of the metadata generation pipeline; systematic extraction errors may propagate into both Stage 1 retrieval and Stage 2 selection. Third, our systems remain purely sparse and do not explicitly model citation graphs, authority hierarchies, or deep semantic entailment. These factors have been explored in stronger neural and hybrid COLIEE systems and may offer promising directions for future work.

6.5 Future Work

A promising direction for future work is to retain the current high-recall sparse retrieval stage while adding a stronger neural or hybrid reranking module on top of it. The results in Table 2 show that our candidate generation stage already recovers a large proportion of gold precedents, especially for the Milvus multi-field system. This makes it a strong foundation for more advanced second-stage models, such as cross-encoders, graph-aware rerankers, or query-specific set-prediction methods. Another direction is to improve metadata generation itself, for example, by extracting case citations, adding consistency constraints across fields or by incorporating explicit modelling of authority and procedural hierarchy.

7 Conclusion

This paper presented two metadata-aware sparse retrieval architectures for COLIEE 2026 Task 1 on legal case retrieval: (i) a Pyserini/Lucene BM25+RM3 system with metadata augmentation, and (ii) a Milvus multi-field sparse BM25 system based on reciprocal rank fusion (RRF). Both systems treat COLIEE Task 1 as a *retrieve-then-select* problem: first generating a high-recall candidate list from masked query fragments, and then applying a learned cutoff predictor to output an unranked set of predicted noticed cases.

The results show that structured legal metadata are effective retrieval signals for precedent recovery. In Stage 1, both systems achieve strong macro Recall@K, with the Milvus multi-field configuration performing slightly better. In the ablation study, Legal Issues is the strongest facet for Pyserini, while Court Analysis is the strongest for Milvus, confirming that sparse retrieval benefits most from metadata that directly encodes legal reasoning. In the official COLIEE 2026 evaluation, our best run achieved a micro-F_1 of 0.3165, placing INTIT fifth overall by team rank.

Overall, these findings show that carefully designed sparse retrieval systems remain highly competitive for legal case retrieval when augmented with structured, reasoning-aware metadata. They also suggest that separating metadata facets into distinct searchable fields and combining them with robust rank fusion yields measurable gains over simpler single-index designs. More broadly, the paper supports the view that making legal reasoning explicit and searchable is a practical way to improve both effectiveness and interpretability in precedent retrieval.

Acknowledgments

We would like to thank NeSI (New Zealand eScience Infrastructure) for providing computational resources and technical support that enabled these experiments.

References

[1] Sophia Althammer, Arian Askari, Suzan Verberne, and Allan Hanbury. 2021. DoSSIER@COLIEE 2021: Leveraging dense retrieval and summarization-based re-ranking for case law retrieval. *arXiv preprint arXiv:2108.03937* (2021).

[2] Arian Askari, Georgios Peikos, Gabriella Pasi, and Suzan Verberne. 2022. LeiBi@COLIEE 2022: Aggregating Tuned Lexical Models with a Cluster-driven BERT-based Model for Case Law Retrieval. *arXiv preprint arXiv:2205.13351* (2022).

[3] Ilias Chalkidis, Manos Fergadiotis, Prodromos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos. 2020. LEGAL-BERT: The Muppets straight out of Law School. arXiv:2010.02559 [cs] doi:10.48550/arXiv.2010.02559

[4] Ying Chen, Yilu Zhou, Zhen Lu, Hao Sun, and Wenjun Yang. 2018. Legal information retrieval by association rules. In *Twelfth International Workshop on Juris-informatics (JURISIN)* (2018).

[5] COLIEE Organizers. 2026. *COLIEE 2026.* Retrieved March 24, 2026 from https://coliee.org/ Competition on Legal Information Extraction/Entailment.

[6] Damian Curran and Mike Conway. 2024. Similarity ranking of case law using propositions as features. In *JSAI International Symposium on Artificial Intelligence*. Springer, 156–166.

[7] Rohan Debbarma, Pratik Prawar, Abhijnan Chakraborty, and Srikanta Bedathur. 2023. Iitdli: Legal case retrieval based on lexical models. In *Workshop of the tenth competition on legal information extraction/entailment (COLIEE'2023) in the 19th international conference on artificial intelligence and law (ICAIL)* (2023).

[8] Wilco Draijer and Suzan Verberne. 2018. Case law retrieval with doc2vec and elastic search. In *Twelfth International Workshop on Juris-Informatics (JURISIN 2018)* (2018).

[9] Yi Feng, Chuanyi Li, and Vincent Ng. 2024. Legal Case Retrieval: A Survey of the State of the Art. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*. 6472–6485.

[10] Baban Gain, Dibyanayan Bandyopadhyay, Tanik Saikh, and Asif Ekbal. 2019. IITP@COLIEE 2019: Legal Information Retrieval using BM25 and BERT. (2019). arXiv:2104.08653 [cs] doi:10.13140/RG.2.2.24136.44804

[11] Randy Goebel, Yoshinobu Kano, Calum Kawn, Mi-Young Kim, Juliano Rabelo, Ken Satoh, Hiroaki Yamada, and Masaharu Yoshioka (Eds.). 2025. *Proceedings of the Workshop on the Twelfth International Competition on Legal Information Extraction and Entailment (COLIEE 2025)* (2025-06-18).

[12] Google DeepMind. 2025. Gemma-3-27B Instruct. https://huggingface.co/google/gemma-3-27b-it.

[13] Yoshinobu Kano, Randy Goebel, Mi-Young Kim, Calum Kwan, Ken Satoh, Hiroaki Yamada, and Masaharu Yoshioka. 2026. An Overview of the COLIEE 2026 Competition: Legal Case Law and Statute Law Information Retrieval and Entailment. In *Proceedings of the 21st International Conference on Artificial Intelligence and Law (ICAIL 2026)*. Singapore.

[14] T. Leburu-Dingalo, E. Thuma, N. Motlogelwa, and M. Mudongo. 2020. Ub_Botswana at COLIEE 2020 Case law retrieval. (2020).

[15] Moemedi Lefoane, Tshepho Koboyatshwene, and Lakshmi Narasimhan. 2018. KNN clustering approach to legal precedence retrieval. In *Twelfth International Workshop on Juris-Informatics (JURISIN 2018)* (2018).

[16] Haitao Li, You Chen, Zhekai Ge, Qingyao Ai, Yiqun Liu, Quan Zhou, and Shuai Huo. 2024. Towards an In-Depth Comprehension of Case Relevance for Better Legal Retrieval. In *JSAI International Symposium on Artificial Intelligence*. Springer, 212–227.

[17] Haitao Li, Weihang Su, Changyue Wang, Yueyue Wu, Qingyao Ai, and Yiqun Liu. 2023. THUIR@COLIEE 2023: Incorporating Structural Knowledge into Pre-trained Language Models for Legal Case Retrieval. *arXiv preprint arXiv:2305.06812* (2023).

[18] Jimmy Lin, Xueguang Ma, Sheng-Chieh Lin, Jheng-Hong Yang, Ronak Pradeep, and Rodrigo Nogueira. 2021. Pyserini: A Python Toolkit for Reproducible Information Retrieval Research with Sparse and Dense Representations. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval* (New York, NY, USA, 2021-07-11) *(SIGIR '21)*. 2356–2362. doi:10.1145/3404835.3463238

[19] Qian Liu, Hang Yu, Qiqi Wang, Qi Xu, Jinpeng Li, Zhuoqun Zou, Rui Mao, and Erik Cambria. 2026. Legal knowledge infusion for large language models: A survey. *Information Fusion* 125 (2026), 103426.

[20] D. Locke and G. Zuccon. 2022. Case law retrieval: accomplishments, problems, methods and evaluations in the past 30 years. *Comput. Surveys* 1, 1 (2022), 1–37.

[21] Yixiao Ma, Yunqiu Shao, Bulou Liu, Yiqun Liu, Min Zhang, and Shaoping Ma. 2021. Retrieving legal cases from a large-scale candidate corpus. *Proceedings of the Eighth International Competition on Legal Information Extraction/Entailment, COLIEE2021* (2021).

[22] A. Mandal, S. Ghosh, K. Ghosh, and S. Mandal. 2020. Significance of textual representation in legal case retrieval and entailment. *COLIEE (2020)* (2020).
[23] K. Tamsin Maxwell and Burkhard Schafer. 2008. Concept and context in legal information retrieval. In *Legal Knowledge and Information Systems*. IOS Press, 63–72.
[24] Hugo Mentzingen, Nuno António, Fernando Bacao, and Marcio Cunha. 2024. Textual similarity for legal precedents discovery: Assessing the performance of machine learning techniques in an administrative court. 4, 2 (2024), 100247. Publisher: Elsevier.
[25] Chau Nguyen, Thanh Tran, Khang Le, Hien Nguyen, Truong Do, Trang Pham, Son T Luu, Trung Vo, and Le-Minh Nguyen. 2024. Pushing the Boundaries of Legal Information Processing with Integration of Large Language Models. In *JSAI International Symposium on Artificial Intelligence*. Springer, 167–182.
[26] Ha-Thanh Nguyen, Phuong Minh Nguyen, Thi-Hai-Yen Vuong, Quan Minh Bui, Chau Minh Nguyen, Binh Tran Dang, Vu Tran, Minh Le Nguyen, and Ken Satoh. 2021. JNLP Team: Deep Learning Approaches for Legal Processing Tasks in COLIEE 2021. *arXiv preprint arXiv:2106.13405* (2021).
[27] Ha-Thanh Nguyen, Hai-Yen Thi Vuong, Phuong Minh Nguyen, Binh Tran Dang, Quan Minh Bui, Sinh Trong Vu, Chau Minh Nguyen, Vu Tran, Ken Satoh, and Minh Le Nguyen. 2020. JNLP Team: Deep Learning for Legal Processing in COLIEE 2020. *arXiv preprint arXiv:2011.08071* (2020).
[28] Luisa Pereira Novaes, Daniela Vianna, and Altigran da Silva. 2023. A topic-based approach for the legal case retrieval task. In *Workshop of the tenth competition on legal information extraction/entailment (COLIEE'2023) in the 19th international conference on artificial intelligence and law (ICAIL)* (2023).
[29] Juliano Rabelo, Randy Goebel, Mi-Young Kim, Yoshinobu Kano, Masaharu Yoshioka, and Ken Satoh. 2022. Overview and Discussion of the Competition on Legal Information Extraction/Entailment (COLIEE) 2021. *The Review of Socionetwork Strategies* 16, 1 (2022), 111–133. doi:10.1007/s12626-022-00105-z
[30] Juliano Rabelo, Mi-Young Kim, H. Babikar, Randy Goebel, and Nawshad Farruque. 2018. Information extraction and entailment for statute law and case law.
[31] Juliano Rabelo, Mi-Young Kim, and Randy Goebel. 2023. Semantic-Based Classification of Relevant Case Law. In *New Frontiers in Artificial Intelligence*, Yasufumi Takama, Katsutoshi Yada, Ken Satoh, and Sachiyo Arai (Eds.). Vol. 13859. Springer Nature Switzerland, 84–95. doi:10.1007/978-3-031-29168-5_6 Series Title: Lecture Notes in Computer Science.
[32] Guilherme Moraes Rosa, Ruan Chaves Rodrigues, Roberto Lotufo, and Rodrigo Nogueira. 2021. Yes, bm25 is a strong baseline for legal case retrieval. *arXiv preprint arXiv:2105.05686* (2021).
[33] Juline Rossi and Evangelos Kanoulas. 2019. Legal information retrieval with generalized language models. *Proceedings of the 6th Competition on Legal Information Extraction/Entailment. COLIEE* (2019).
[34] Yunqiu Shao, Bulou Liu, Jiaxin Mao, Yiqun Liu, Min Zhang, and Shaoping Ma. 2020. Thuir@ coliee-2020: Leveraging semantic understanding and exact matching for legal case retrieval and entailment. *arXiv preprint arXiv:2012.13102* (2020).
[35] Yanran Tang, Ruihong Qiu, Yilun Liu, Xue Li, and Zi Huang. 2024. CaseGNN++: Graph Contrastive Learning for Legal Case Retrieval with Graph Augmentation. *arXiv preprint arXiv:2405.11791* (2024).
[36] Yanran Tang, Ruihong Qiu, Yilun Liu, Xue Li, and Zi Huang. 2024. CaseGNN: Graph Neural Networks for Legal Case Retrieval with Text-Attributed Graphs. In *European conference on information retrieval*. Springer, 80–95.
[37] Yanran Tang, Ruihong Qiu, Hongzhi Yin, Xue Li, and Zi Huang. 2024. CaseLink: Inductive Graph Learning for Legal Case Retrieval. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Washington DC USA, 2024-07-10). ACM, 2199–2209. doi:10.1145/3626772.3657693
[38] Vu Tran, Minh Le Nguyen, and Ken Satoh. 2019. Building Legal Case Retrieval Systems with Lexical Matching and Summarization using A Pre-Trained Phrase Scoring Model. In *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Law* (Montreal QC Canada, 2019-06-17). 275–282. doi:10.1145/3322640.3326740
[39] Vu Tran, Son Truong Nguyen, and Minh Le Nguyen. 2018. JNLP group: legal information retrieval with summary and logical structure analysis. In *International Workshop on Juris-Informatics (JURISIN 2018)* (2018).
[40] Marc Van Opijnen and Cristiana Santos. 2017. On the concept of relevance in legal information retrieval. *Artificial Intelligence and Law* 25, 1 (2017), 65–87. doi:10.1007/s10506-017-9195-8
[41] Thi-Hai-Yen Vuong, Hai-Long Nguyen, Tan-Minh Nguyen, Hoang-Trung Nguyen, Thai-Binh Nguyen, and Ha-Thanh Nguyen. 2024. NOWJ at COLIEE 2023: Multi-task and Ensemble Approaches in Legal Information Processing. *The Review of Socionetwork Strategies* 18, 1 (2024), 145–165. doi:10.1007/s12626-024-00157-3
[42] J. Wen, Z. Zhong, Y. Bai, X. Zhao, and M. Yang. 2022. Siat@ coliee-2022: legal case retrieval with longformer-based contrastive learning. In *Sixteenth International Workshop on Juris-informatics (JURISIN)* (2022).
[43] Hannes Westermann, Jaromir Savelka, and Karim Benyekhlef. 2020. Paragraph similarity scoring and fine-tuned BERT for legal information retrieval and entailment. In *JSAI International Symposium on Artificial Intelligence*. Springer, 269–285.
[44] Qi Xu, Qian Liu, Hao Fei, Hang Yu, Shuhao Guan, and Xiao Wei. 2025. CLEAR: A Framework Enabling Large Language Models to Discern Confusing Legal Paragraphs. In *Findings of the Association for Computational Linguistics: EMNLP*. 8937–8953.
[45] Qi Xu, Xiao Wei, Hang Yu, Qian Liu, and Hao Fei. 2024. Divide and Conquer: Legal Concept-guided Criminal Court View Generation. In *Findings of the Association for Computational Linguistics: EMNLP*. 3395–3410.

Received 05 April 2026; accepted 02 May 2026; revised 29 May 2026

UCDCS at COLIEE 2026: A Multi-Stage Framework for Legal Case Retrieval via Structural Abstraction and Specialised LLMs

Yuchen Zhang
School of Computer Science
University College Dublin
Dublin, Ireland
yuchen.zhang3@ucdconnect.ie

David Lillis
School of Computer Science
University College Dublin
Dublin, Ireland
david.lillis@ucd.ie

Abstract

Automated legal case retrieval is a critical yet challenging task due to the extreme length of judicial documents and the complexity of judicial reasoning. In this paper, we present our multi-stage framework for the COLIEE 2026 Task 1 (Legal Case Retrieval). To address the challenges of long-form legal texts, we first employ a structural abstraction strategy that distills cases into key factual and logical components. Our retrieval pipeline utilises a hybrid strategy combining BM25 with BGE-M3 dense embeddings, establishing a high-recall foundation (≈0.85). For the subsequent re-ranking stage, we move beyond general-purpose semantic matching by leveraging fine-tuned MonoT5 models and SaulLM-7B, a specialised legal large language model. This transition allows the system to prioritise logic over surface-level topical similarity. Among the three runs we submitted, the best performance achieved by the proposed framework reached a final precision of 0.2480 and an F1-score of 0.2645 on the evaluation set, improving upon the retrieval-only baselines. These results indicate that combining hybrid retrieval with domain-adapted re-ranking is a promising approach.

CCS Concepts

• **Information systems** → **Retrieval models and ranking**; • **Applied computing** → **Law**.

Keywords

Legal Case Retrieval, Large Language Models, Structural Abstraction, Judicial Reasoning

ACM Reference Format:
Yuchen Zhang and David Lillis. 2026. UCDCS at COLIEE 2026: A Multi-Stage Framework for Legal Case Retrieval via Structural Abstraction and Specialised LLMs. In *COLIEE 2026, June 12, 2026, Singapore.* ACM, New York, NY, USA, 8 pages.

1 Introduction

As the body of case law continues to expand, it becomes increasingly challenging for legal professionals to identify relevant precedents efficiently. Automated legal Information Retrieval systems can help address this difficulty by retrieving prior cases that may support judicial decision-making. The Competition on Legal Information Extraction and Entailment (COLIEE) provides a standardised benchmark for evaluating such systems, with a particular focus on legal retrieval and entailment tasks [12].

In this paper, we focus on Task 1: Legal Case Retrieval, which aims to automatically retrieve "noticed cases" for a given query case. Following the official task definition[1], a noticed case refers to a prior case that is referenced by the query case and serves as supporting evidence for its legal decision. To ensure a realistic evaluation setting, all citation information is removed from the query cases, requiring systems to infer these relationships solely based on textual and semantic similarity. Given a corpus of Federal Court of Canada case law, the task requires fully automatic retrieval without any human intervention, and participants must predict the noticed cases without access to the test labels before submission.

Legal case retrieval poses several challenges. First, legal documents are often lengthy and structurally complex, making it difficult for traditional keyword-based methods to capture deep semantic relationships. Second, legal relevance is not determined by lexical overlap alone, and accurate retrieval often depends on semantic and reasoning cues beyond exact term matching [9]. Finally, in the COLIEE Task 1 setting, the number of relevant cases varies substantially across queries, which complicates threshold selection and result filtering.

To address these challenges, we propose a multi-stage retrieval and re-ranking framework. Our approach first leverages large language models (LLMs) to generate structured summaries that capture key legal elements of each case. We then employ a hybrid retrieval strategy that combines lexical matching using BM25 with semantic similarity based on BGE-M3 embeddings to generate candidate cases [3]. Finally, we apply domain-specific re-ranking to refine the retrieved candidates and improve final retrieval performance. This design is intended to better capture semantic relationships between legal cases and produce more accurate retrieval results.

Our main contributions are as follows:

- We developed an LLM-based legal abstraction step to generate structured case summaries, reducing noise while preserving essential legal information;
- We build a hybrid retrieval framework that combines lexical matching with dense semantic representations for legal case retrieval over long documents;
- We conduct a comparative study of general-purpose and domain-specific re-ranking models, demonstrating the effectiveness of legal-specialised models in capturing fine-grained case relationships.

[1] https://coliee.org/COLIEE2026/tasks/task1

COLIEE 2026, Singapore

2 Task Description

Task 1 of COLIEE focuses on legal case retrieval over a corpus of Federal Court of Canada decisions. The dataset consists of a collection of case law documents, where each case may serve as a query or a candidate case.

The training data used in this study is constructed from the union of the previous year's Task 1 training and test collections, yielding a candidate pool of 7,709 cases. The test collection consists of 1,848 candidate cases. Only the cases appearing as keys in the JSON file are treated as query cases; the system retrieves relevant noticed cases for each query from the corresponding candidate pool. In the training set, the annotations are provided in JSON format, where each query case is mapped to a set of ground-truth noticed case identifiers. For example:

```
{
 "000001.txt":
    ["000005.txt", "012101.txt"],
 "003423.txt":
    ["398421.txt", "012101.txt", "173651.txt"],
 "012831.txt":
    ["000001.txt"],
 ...
}
```

For the test set, only the query cases are provided, and the system is required to retrieve the corresponding noticed cases from the entire corpus. The retrieval process must be fully automatic and cannot rely on any manual intervention.

The task is formulated as a retrieval problem, where systems are expected to return a ranked list of candidate cases for each query, ordered by relevance.

System performance is evaluated using the standard Information Retrieval metrics of Precision, Recall, and F1-score, which together measure the accuracy and completeness of the retrieved results.

3 Related Work

Legal Information Retrieval, and Legal Case Retrieval in particular, has evolved alongside broader developments in Natural Language Processing, moving from keyword-based retrieval to semantic and neural methods. In the COLIEE competition, prior systems have mainly addressed three recurring challenges: vocabulary mismatch, the extreme length of legal documents, and the need to capture complex legal reasoning [11].

3.1 Lexical and Hybrid Retrieval

Traditional lexical methods such as BM25 [20] remain strong baselines because they perform well when relevant documents share explicit terms with the query. However, their reliance on surface-level term matching makes them vulnerable to vocabulary mismatch. To mitigate this limitation, many recent systems combine lexical retrieval with dense semantic retrieval [9, 17]. Transformer-based encoders such as BERT and its variants [8, 16], as well as Sentence-BERT [19], have made it possible to model semantic similarity beyond exact word overlap. One recent COLIEE submission reported that combining BM25 with dense retrievers can improve candidate recall [17].

3.2 Neural Re-ranking and Domain Adaptation

Neural re-ranking plays a crucial role in modern retrieval systems because it allows more fine-grained modelling of query-document interactions [27]. MonoT5 [18] has shown strong performance in re-ranking tasks by reformulating ranking as sequence generation. However, the domain gap between general-purpose training data and legal language remains a challenge [2, 9]. Domain adaptation techniques, such as fine-tuning on legal corpora or using domain-specific models like LEGAL-BERT [2] can improve performance in Legal NLP tasks. A prior COLIEE system has also explored related strategies, including LLM-based summarization and fine-tuned ranking models [24]. In this paper, we extend this line of work by comparing general-purpose and domain-adapted re-ranking models in our retrieval pipeline.

3.3 Legal-Specific Large Language Models

LLMs have also opened new possibilities for legal text modelling in the last two years, especially for tasks that require reasoning over facts, legal principles, and case relationships. Techniques such as Chain-of-Thought prompting [23] and Retrieval-Augmented Generation [13] have shown strong potential. In the legal domain, specialised models trained on legal corpora, such as SaulLM-7B [4], provide improved understanding of legal terminology and reasoning, making them suitable for identifying relationships between cases.

3.4 Summarisation and Information Distillation

The extreme length of legal documents presents significant challenges for neural models, often leading to information loss in long-context processing. This has resulted in a variety of approaches to address this issue, including representing long legal documents as alternative data structures such as graphs [26]. Other structure-aware approaches that leverage domain knowledge include LSDK-LegalSum [10], which emphasises partitioning judicial judgments into functional logical sections. In this work, we draw inspiration from the UMNLP team's strategy at COLIEE 2024 [6], which demonstrated the effectiveness of distilling cases into concise legal *propositions* to fit within model constraints while preserving essential facts. These studies support the broader idea that legal information distillation can improve retrieval effectiveness under long-context constraints.

4 Methodology

As shown in Figure 1, we propose a multi-stage retrieval and re-ranking framework designed to address the challenges of long-context legal documents, implicit semantic relationships, and the absence of explicit citations in COLIEE Task 1. Given a query case q and a corpus C, the system aims to retrieve a ranked list of noticed cases. The overall pipeline consists of four stages: (1) legal semantic abstraction, (2) hybrid first-stage retrieval, (3) neural re-ranking, and (4) result selection and filtering. These stages are described in the following sections.

4.1 Legal Semantic Abstraction

Legal case documents are often lengthy and complex, and not all parts of a full judgment are equally useful for notice-case retrieval.

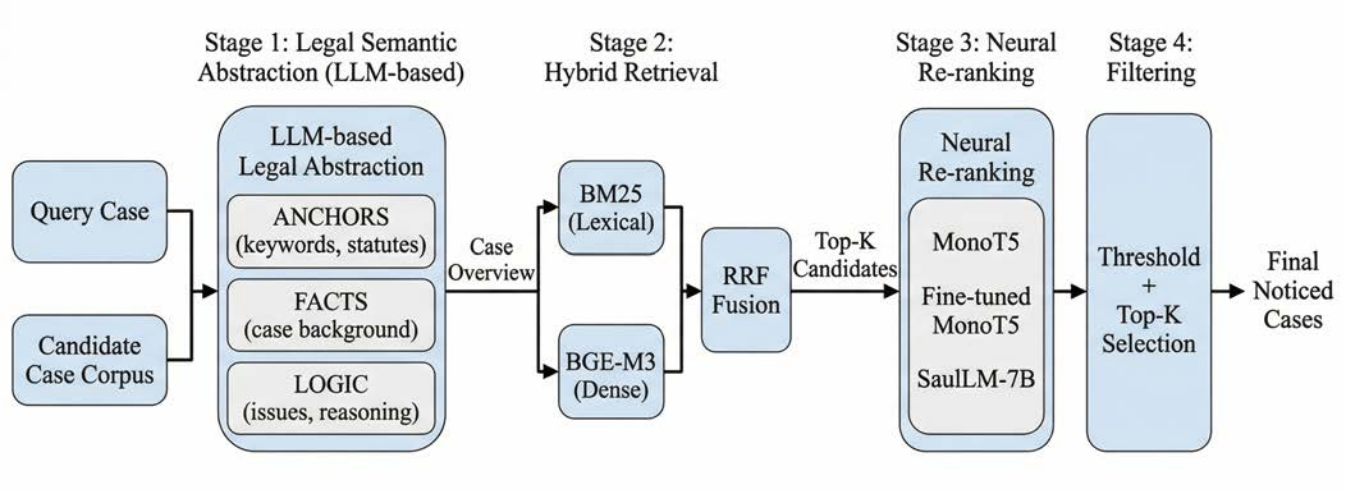


Figure 1: Overview of the proposed multi-stage legal case retrieval framework.

In particular, retrieval is more likely to benefit from information about the core legal issue, the reasoning of the decision, and the most salient facts than from procedural background or other material that is peripheral to the dispute [7, 21]. To reduce this noise, we generate a structured case overview for each document using the `deepseek-chat` LLM through an API-based prompting pipeline. The overview generation step is designed to preserve legally salient information while compressing the original judgment into a shorter and more retrieval-oriented representation.

Each case is converted into a structured summary with three predefined sections:

- **[ANCHORS]**: key legal terms and cited statutory provisions;
- **[LOGIC]**: the main legal issue and the core reasoning of the decision;
- **[FACTS]**: a concise description of the factual background.

The prompt explicitly instructs the model to extract legal keywords, identify exact statute names and section numbers, summarise the core legal issue and *ratio decidendi*, and provide a brief factual summary. It also includes a task-specific constraint requiring the tag *Foreign Law Interpretation* when the case turns on the interpretation of foreign law. To ensure consistent formatting, the model is required to output the exact section headers and terminate the response with a special `[END]` marker, after which the output is post-processed and cleaned. The full prompt is provided in Appendix A.

The intent of this structured abstraction is to reduce the length of the document while preserving information that is more likely to support notice-case retrieval. The resulting overviews are then used as inputs for both first-stage retrieval and downstream re-ranking.

4.2 Hybrid First-stage Retrieval

To obtain a high-recall candidate pool, we adopt a hybrid retrieval framework that combines lexical matching with dense semantic retrieval. All retrieval operations are performed on the generated case overviews.

Lexical Retrieval. We use BM25 [20] as a strong baseline for term-based matching. Documents are preprocessed using tokenisation and stemming. We tuned the BM25 hyperparameters empirically on COLIEE 2026 training set and found that $k_1 = 1.6$ and $b = 0.9$ gave the best performance for our legal retrieval task.

Dense Retrieval. We employ BGE-M3 [3], a dense embedding model capable of encoding long textual inputs. Each case overview is mapped to a dense vector, and similarity is computed using inner product.

Fusion. The ranked lists from BM25 and dense retrieval are combined using Reciprocal Rank Fusion (RRF) [5], with a smoothing parameter $K = 60$, following the original paper. RRF was chosen on the basis that it is a commonly-used fusion method that is straightforward to implement [14]. The fused score $RRF(d)$ for a document d is defined as:

$$\mathrm{RRF}(d) = \sum_{r \in R} \frac{1}{K + \mathrm{rank}_r(d)}$$

where R denotes the set of retrieval methods and $\mathrm{rank}_r(d)$ is the rank of document d in the set of results returned by method r. We retain the top 200 fused candidates for the next stage. We chose this cut-off to match the number of results returned by both BM25 and dense retrieval, so that fusion preserves the full candidate set contributed by each retriever while still providing a manageable pool with good recall for downstream re-ranking.

4.3 Neural Re-ranking

To further refine the candidate set, we apply neural re-ranking models that capture fine-grained relationships between query and candidate cases. Each query-candidate pair is formatted as a text pair and scored independently. We submitted three final runs to COLIEE, each corresponding to a different re-ranking pipeline: Pipeline-Base (submitted with the label `ucdcs1`), Pipeline-FineTune (`ucdcs2`), and Pipeline-SaulLM (`ucdcs3`). All three pipelines share the same document abstraction and first-stage retrieval framework,

and differ only in the re-ranking model used in the second stage. The following paragraphs describe the three submitted pipelines.

Pipeline-Base (ucdcs1). We use MonoT5 [18], a sequence-to-sequence model trained on MS MARCO [1], as a general-purpose re-ranker. The model takes a query-case pair as input and produces a relevance score in a zero-shot setting.

Pipeline-FineTune (ucdcs2). To reduce the domain gap between general-domain ranking data and legal case retrieval, we fine-tune MonoT5 on the training set, starting from `castorini/monot5-base-msmarco`. The task is formulated as binary sequence generation over query-document pairs. Both queries and candidate documents are represented by the generated case overviews rather than the full judgments.

Positive pairs are constructed from gold noticed-case labels. Negative pairs are limited to at most five per query. We define *hard negatives* as non-relevant candidate documents retrieved for the same query by the first-stage candidate pool. After excluding the query document itself, gold positives, and documents without available overviews, we take the top 30 such non-relevant candidates as a hard-negative pool and randomly sample up to five from this pool. Random negatives are used only as a fallback when fewer than five hard negatives are available, for example if the candidate pool is missing or if too few valid non-relevant candidates remain after filtering. To avoid query leakage, the data are split at the query level into training and development sets with an 80/20 split. We use a maximum input length of 1024 tokens to preserve richer legal context, and train the model for 2 epochs with a learning rate of 1×10^{-5}.

Pipeline-SaulLM (ucdcs3). Our third run is based on a domain-specific large language model, SaulLM-7B [4], which is pre-trained on large-scale legal corpora. We formulate the re-ranking task as an instruction-following problem, prompting the model to determine whether a candidate case provides decision-supporting evidence for a query case.

Instead of directly generating textual outputs, we convert the model's predictions into continuous relevance scores via a logit-based scoring mechanism. During inference, we extract the raw logits corresponding to the first generated token. Let L_{Yes} and L_{No} denote the logits for the tokens "Yes" and "No", respectively. We then define the relevance score S for a query–document pair as:

$$S = \frac{e^{L_{\text{Yes}}}}{e^{L_{\text{Yes}}} + e^{L_{\text{No}}}} \tag{1}$$

This formulation provides a relative relevance score for ranking, enabling fine-grained ranking and more stable threshold-based filtering in the final stage. In practice, we use a context window of up to 4,096 tokens to accommodate the structured case overviews.

4.4 Result Selection and Filtering

A key challenge in Task 1 is that the number of noticed cases varies substantially across queries and is unknown at inference time. We therefore require a decision rule that converts the continuous scores produced by the neural re-rankers into a discrete set of predictions. To address this issue, we apply an adaptive filtering procedure whose parameters are tuned on a hold-out development split of the training data and then fixed for inference on the unseen test set. The goal is to balance precision and recall while accounting for query-level variation in the number of relevant cases.

Hybrid Cutoff Strategy. Our main method is a Hybrid Cutoff strategy that combines a rank-based limit with a score threshold. For each query, we first retain the top $K = 7$ candidates according to the re-ranking scores. We then apply a threshold $\tau = 0.75$ and remove candidates whose scores fall below this value. This two-stage procedure restricts the output to highly ranked candidates while filtering out low-scoring ones, which helps maintain precision. However, because the final output is selected from a much smaller subset of the top-200 candidate pool, relevant cases that are not assigned sufficiently high re-ranking scores may fall below the final cut-off and therefore be excluded from the submitted predictions.

Score Gap Analysis. For the fine-tuned models and SaulLM, we also examine a Score Gap strategy. Instead of using a fixed score threshold, this method scans the ranked list from top to bottom and computes the score difference between adjacent candidates. Let s_i denote the score of the candidate at rank i. We define the local score gap as $\Delta_i = s_{i-1} - s_i$. Starting from the top-ranked candidate, we retain candidates until the first position where $\Delta_i \geq \gamma$, where γ is a run-specific gap threshold, selected separately for each run on the development set by grid search; this point is treated as a boundary between the leading score cluster and lower-confidence candidates. This makes the cut-off less rigid than a fixed threshold alone.

Forced Selection Mechanism. To avoid the extreme case in which no candidate survives filtering, we further apply a Forced-One Fallback. If the filtered set S_{final} is empty, the highest-scoring candidate is returned. This guarantees that each query receives at least one predicted noticed case and avoids a zero-recall outcome for that query.

Overall, this result selection procedure provides a practical way to convert ranking scores into final predictions. It combines rank-based pruning, score-based filtering, and a simple fallback mechanism to better handle variation in the number of relevant cases across queries.

5 Pipeline Development

The development of our system was an iterative process informed by comparative analysis on the COLIEE training data. In this section, we summarise the main alternatives we explored and explain how these observations informed the design of our final submission pipelines.

5.1 Alternative Methodology Evaluation

We first evaluated retrieval-only approaches, including BM25, dense retrieval, and a hybrid of these. While these methods achieved relatively high recall, they consistently suffered from low precision, as many retrieved cases were only topically related rather than legally relevant. This suggests that retrieving a large candidate set is insufficient without accurately identifying decision-supporting relationships.

We also tested representative re-ranking models from prior work, such as `BGE-Reranker-v2-m3`. While such models are effective on general-domain retrieval benchmarks, they were less competitive

in our legal retrieval setting. In particular, they appeared to favour topical similarity over deeper legal relevance, and therefore often failed to capture the case relationships needed for noticed-case identification. For this reason, we did not include these models in our final submission pipelines.

5.2 Selection of Final Pipelines

Based on the above observations, we conclude that a two-stage framework combining high-recall retrieval with strong re-ranking is necessary for this task.

We therefore adopt MonoT5 (both base and fine-tuned variants) and SaulLM-7B as our primary re-ranking models. These models provide complementary strengths: MonoT5 offers stable performance as a cross-encoder, while SaulLM-7B demonstrates enhanced capability in modelling complex legal reasoning. Instead of relying on a single architecture, we design three independent pipelines to explore different trade-offs between generalisation and domain-specific reasoning.

6 Experiments and Results

In this section, we present the evaluation of our proposed framework on the COLIEE 2026 Task 1 dataset.

6.1 Evaluation Metrics and Setup

Following the official COLIEE evaluation protocol, we report Precision, Recall, and F1-score. For the official submissions, our models were trained on the released COLIEE training set and then used to predict noticed cases for the unlabelled test set. No test labels were available at submission time. All experiments were performed on an Apple M2 Max device with 32GB unified memory.

6.2 Results

Table 1 presents the official evaluation results of our three submitted runs. Among them, `ucdcs3`, which uses SaulLM as the re-ranking model, achieves the best F1-score. The differences among the three UCD-CS runs are relatively small, with all three producing comparable overall performance.

Overall, our system ranks in the middle tier of the shared task. These results indicate that the proposed framework provides a competitive baseline, while also leaving clear room for further improvement.

6.3 Comparison of Submitted Runs

Although the three submitted runs share the same first-stage retrieval framework, they differ in the re-ranking model and final selection strategy. As discussed in Section 3.2, `ucdcs1` uses zero-shot MonoT5-Base as the re-ranker, `ucdcs2` uses a domain-finetuned MonoT5-Base model, and `ucdcs3` uses SaulLM-7B with logit-based scoring. As shown in Table 1, these three configurations achieve similar official F1-scores despite using different re-ranking models.

To further compare the behaviour of the three runs, we measured the overlap between their final predicted noticed-case sets using pairwise Jaccard similarity. For each query, we computed the Jaccard similarity between the returned case sets of each pair of runs and then averaged the scores over all queries. To distinguish overall output overlap from overlap in relevant retrieved cases, we also computed a correctness-restricted Jaccard similarity using only correctly returned noticed cases, i.e., after intersecting each run's prediction set with the gold noticed-case set for that query. The results of these are shown in Table 2.

Team	Run	F1	P	R
NOWJ	submission_2	0.4220	0.4235	0.4206
JNLP	random_forest	0.4126	0.4341	0.3931
SIL	submission_sil2	0.3871	0.4186	0.3600
mezza	task_1_mezzanino	0.3289	0.3161	0.3429
INTIT	3.intit_bm25_year	0.3165	0.3249	0.3086
DU	du3	0.3141	0.2945	0.3366
FLNLP	task1_flnlpltr	0.2935	0.2619	0.3337
UA	ua_run3	0.2666	0.2038	0.3851
UCD-CS	**ucdcs3**	**0.2645**	**0.2480**	**0.2834**
UCD-CS	**ucdcs1**	**0.2607**	**0.2131**	**0.3354**
UCD-CS	**ucdcs2**	**0.2565**	**0.2405**	**0.2749**
JUNLLP	task1_run2_results	0.2525	0.2193	0.2977
74688	task-1-ualbany	0.2346	0.2130	0.2611
UB2026	run1	0.2233	0.2338	0.2137
Sach	sach_task1_run2	0.2231	0.1631	0.3531
BJPWH	bjpwh3	0.2101	0.1510	0.3451
ABAI	run2recall	0.1771	0.1596	0.1989
AIIRLab	task1_aiirqwen	0.1516	0.2642	0.1063
bosch	bosch_task1_run	0.1466	0.0892	0.4103
KeioAndrewShin	task1_submission_v9	0.1383	0.1527	0.1263

Table 1: COLIEE Task 1 results ranked by F1 score. For each team, only the best-performing run (by F1) is reported, except for UCD-CS where all runs are included (P=Precision, R=Recall).

Run Pair	Overall Jaccard	Correct-only Jaccard
ucdcs1 vs ucdcs2	0.4741	0.7860
ucdcs1 vs ucdcs3	0.4864	0.8010
ucdcs2 vs ucdcs3	0.7137	0.8659

Table 2: Average pairwise Jaccard similarity between the final returned noticed-case sets of our three submitted runs, and between their correctly returned noticed cases only, computed over all test queries.

The results show that `ucdcs2` and `ucdcs3` are substantially more similar to each other than either is to `ucdcs1` in their overall predictions. However, the correctness-restricted Jaccard scores are markedly higher for all three run pairs, ranging from 0.7860 to 0.8659. This indicates that, although the runs often return different final case sets overall, there is a much stronger overlap for relevant noticed cases they retrieve. In other words, much of the disagreement between the submitted runs arises from differences in non-relevant returned cases rather than from retrieving fundamentally different relevant cases.

As an additional analysis, we compared the three submitted runs using ranking-oriented metrics and also evaluated an exploratory fusion of the three runs using Reciprocal Rank Fusion (RRF).

While the official submission results in Table 1 are reported in terms the set-based metrics of precision, recall, and F1, we further computed Mean Average Precision (MAP) in a *post-hoc* analysis

Run	MAP	F1	Precision	Recall
ucdcs1	0.2928	0.2607	0.2131	0.3354
ucdcs2	0.2427	0.2565	0.2405	0.2749
ucdcs3	0.2660	0.2645	0.2480	0.2834
RRF-fused (Top-5)	0.2699	0.2587	0.2460	0.3511

Table 3: Comparison of the three submitted runs and an exploratory RRF fusion. The three submitted runs are evaluated over their full returned lists, while the RRF-fused run is evaluated at top 5.

using the official ground-truth labels released by the COLIEE organisers. As shown in Table 3, ucdcs1 achieved the highest MAP (0.2928) amongst our runs, followed by ucdcs3 and then ucdcs2. This suggests that, although ucdcs3 performed best under the official F1-based evaluation, ucdcs1 produced the strongest ranking quality under this additional ranking-based analysis.

The Jaccard similarities shown above in Table 2 also motivated us to conduct an additional *post-hoc* analysis, on the basis that ranked lists with a high level of agreements on relevant documents can often be effectively aggregated under the "chorus effect" of data fusion [22]. We therefore fused the three submitted runs, again using RRF with the standard constant $K = 60$, and evaluated the fused ranking at top 5. This fixed cutoff was chosen as it is reflective of the typical cutoff values selected for the individual runs during training under the process outlined in Section 4.4 (which in practice ranged between 5 and 7 when optimising for F1). The fused run achieved MAP@5 of 0.2699, Precision@5 of 0.2460, Recall@5 of 0.3511, and F1@5 of 0.2587, as also shown in Table 3. While this setting does not exactly replicate the selection mechanisms used in the individual runs, it still allows for a broadly comparable analysis of ranking behaviour. Taken together with the Jaccard analysis, these results suggest that the three runs differ noticeably in their overall returned sets, but already overlap strongly in the correctly retrieved noticed cases, which limits the gains obtainable from simple post-hoc fusion.

6.4 Impact of Retrieval and Re-ranking

To analyse the role of each stage, we examine the performance of retrieval-only methods versus our best-performing full pipeline, which uses SaulLM for re-ranking. This comparison is based on our own post-hoc evaluation using the ground-truth labels released by COLIEE, and is therefore intended as an internal analysis of system behaviour rather than as part of the official competition results.

Method	MAP	F1	Precision	Recall
BM25 (Top-200)	0.3530	0.0634	0.0339	0.7866
Dense (Top-200)	0.3055	0.0348	0.0180	0.8210
Hybrid (Top-200)	0.3455	0.0359	0.0186	**0.8493**
Hybrid (Top-20)	0.3243	0.1784	0.1156	0.5760
Full Pipeline (SaulLM)	0.2660	**0.2645**	0.2480	0.2834

Table 4: Comparison between retrieval-only methods and the full pipeline.

As shown in Table 4, retrieval-only methods serve as an initial coarse filtering stage: they achieve high recall while substantially reducing the number of candidate cases that need to be examined. Their MAP values also show that they can rank some relevant cases near the top of the candidate list. However, their precision is extremely low, resulting in poor F1-scores. Applying neural re-ranking on these filtered candidates substantially improves precision and overall F1 performance, at the cost of recall and consequently MAP.

This result demonstrates that in legal case retrieval, retrieval acts primarily to narrow down the search space and ensure relevant cases are included, while re-ranking is essential for accurately identifying the most relevant cases.

7 Discussion

The Precision-Recall Trade-off in Legal Retrieval. The results in Table 4 show clear performance shifts across different stages of the pipeline. The hybrid retrieval stage (BM25 + BGE-M3) achieves high recall (0.8493), but recall decreases to 0.2834 in the full pipeline, together with a substantial increase in precision. This trade-off reflects the design and current limitation of our system. The first-stage retriever is intended to preserve a broad set of potentially relevant cases, and the top-200 candidate pool already contains many true noticed cases. However, the re-ranking stage does not consistently assign sufficiently high scores to all of these true noticed cases. As a result, some relevant cases remain in the candidate pool but are ranked below the final output cut-off and are not included in the final predictions.

In COLIEE Task 1, where the number of noticed cases varies across queries and is unknown at inference time, this ranking difficulty becomes especially important. Legal relevance is not determined by topical similarity alone, and some noticed cases may support the query case through more subtle reasoning links. This explains why the system achieves a large precision gain over retrieval-only baselines but does not obtain a proportional improvement in recall or overall F_1.

Beyond Semantic Similarity: Judicial Logic vs. Topicality. One important observation from our development experiments is the difference between general semantic similarity and legal relevance. General-purpose rerankers often assign high scores to case pairs that are topically similar but do not have a genuine relationship in terms of supporting legal reasoning.

Interestingly, the zero-shot MonoT5-Base model (ucdcs1) provided a strong baseline and outperformed our fine-tuned variant (ucdcs2). One possible explanation is that the available legal training data are limited, which may make the fine-tuned model more vulnerable to overfitting. At the same time, the better performance of SaulLM-7B suggests that legal-domain pre-training remains useful for capturing case relationships that depend on judicial reasoning, which is consistent with our previous findings on domain pre-training in the context of legal argumentation mining [25]. Our logit-based scoring method also allows SaulLM outputs to be used as continuous ranking scores rather than simple binary decisions.

Efficacy of Structural Legal Abstraction. The transition from full-text processing to structured *Legal Overviews* appears to play an important role. By distilling legal documents into structured components—[ANCHORS], [FACTS], and [LOGIC], we aim to reduce procedural noise and preserve the information most relevant to

noticed-case retrieval. This design is also motivated by the 'lost in the middle' phenomenon common in long-context Transformer architectures, where models tend to make less effective use of information located in the middle of long input sequences [15]. Similar structured prompting methodologies have also shown promise in handling extensive legal datasets where procedural noise often masks critical signals [10].

Limitations and Future Work. Our experimental results demonstrate that the hybrid retrieval stage achieves high recall (i.e., 0.8493 at $K = 200$), indicating that our framework provides a reliable candidate pool for downstream re-ranking. However, although the subsequent re-ranking stage improves precision, it does not produce a comparable gain in overall F_1, partly because the increase in precision is accompanied by a reduction in recall. This suggests that the current neural re-rankers may still struggle to consistently identify all true noticed cases from the top-200 candidates, especially when the relevant cases depend on complex logical signals rather than strong topical similarity.

Furthermore, the system remains sensitive to the quality of the LLM-generated overviews; missing critical legal anchors during the abstraction phase can lead to cascading errors in the downstream process. Despite the long-context capabilities of BGE-M3, our system also faces challenges in handling subtle, cross-case analogies that necessitate multi-hop reasoning.

Consequently, our future research will prioritise the exploration of more effective re-ranking architectures and specialised fine-tuning strategies specifically tailored for judicial logic. By investigating alternative model variants and more intensive domain-specific training, we expect to further bridge the gap between candidate retrieval and precise case identification.

8 Conclusion

In this work, we presented a multi-stage retrieval and re-ranking framework for legal case retrieval in COLIEE 2026 Task 1. The framework combines structured legal abstraction, hybrid first-stage retrieval, and neural re-ranking to address several core challenges of the task, including long judicial documents, vocabulary mismatch, and the difficulty of identifying decision-supporting case relationships.

Our experiments show that hybrid retrieval can provide a high-recall candidate pool, while downstream re-ranking improves the precision of the final predictions. The results also suggest that legal-domain adaptation remains important for this task, although accurately distinguishing subtle decision-supporting relationships between topically similar cases is still difficult. In particular, the gap between candidate recall and final ranking performance indicates that re-ranking remains the main challenge in the current system.

Future work will focus on stronger legal re-ranking strategies, more effective domain-specific training, and improved use of structured legal representations. These directions may help narrow the gap between broad candidate retrieval and precise noticed-case identification.

References

[1] Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, Mir Rosenberg, Xia Song, Alina Stoica, Saurabh Tiwary, and Tong Wang. 2016. MS MARCO: A Human Generated MAchine Reading COmprehension Dataset. https://arxiv.org/abs/1611.09268v3

[2] Ilias Chalkidis, Manos Fergadiotis, Prodromos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos. 2020. LEGAL-BERT: The Muppets straight out of Law School. In *Findings of the Association for Computational Linguistics: EMNLP 2020*. Association for Computational Linguistics, Online, 2898–2904. doi:10.18653/v1/2020.findings-emnlp.261

[3] Jianlyu Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2024. M3-Embedding: Multi-Linguality, Multi-Functionality, Multi-Granularity Text Embeddings Through Self-Knowledge Distillation. In *Findings of the Association for Computational Linguistics: ACL 2024*, Lun-Wei Ku, Andre Martins, and Vivek Srikumar (Eds.). Association for Computational Linguistics, Bangkok, Thailand, 2318–2335. doi:10.18653/v1/2024.findings-acl.137

[4] Pierre Colombo, Telmo Pessoa Pires, Malik Boudiaf, Dominic Culver, Rui Melo, Caio Corro, Andre F. T. Martins, Fabrizio Esposito, Vera Lúcia Raposo, Sofia Morgado, and Michael Desa. 2024. SaulLM-7B: A pioneering Large Language Model for Law. doi:10.48550/arXiv.2403.03883 arXiv:2403.03883 [cs].

[5] Gordon V. Cormack, Charles L A Clarke, and Stefan Buettcher. 2009. Reciprocal rank fusion outperforms condorcet and individual rank learning methods. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*. ACM, Boston MA USA, 758–759. doi:10.1145/1571941.1572114

[6] Damian Curran and Mike Conway. 2024. Similarity Ranking of Case Law Using Propositions as Features. In *New Frontiers in Artificial Intelligence*, Toyotaro Suzumura and Mayumi Bono (Eds.). Springer Nature, Singapore, 156–166. doi:10.1007/978-981-97-3076-6_11

[7] Aniket Deroy, Kripabandhu Ghosh, and Saptarshi Ghosh. 2025. Applicability of large language models and generative models for legal case judgement summarization. *Artificial Intelligence and Law* 33, 4 (Dec. 2025), 1007–1050. doi:10.1007/s10506-024-09411-z

[8] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, Jill Burstein, Christy Doran, and Thamar Solorio (Eds.). Association for Computational Linguistics, Minneapolis, Minnesota, 4171–4186. doi:10.18653/v1/N19-1423

[9] Yi Feng, Chuanyi Li, and Vincent Ng. 2024. Legal Case Retrieval: A Survey of the State of the Art. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Lun-Wei Ku, Andre Martins, and Vivek Srikumar (Eds.). Association for Computational Linguistics, Bangkok, Thailand, 6472–6485. doi:10.18653/v1/2024.acl-long.350

[10] Wei Gao, Shuai Yu, Yongbin Qin, Caiwei Yang, Ruizhang Huang, Yanping Chen, and Chuan Lin. 2025. LSDK-LegalSum: improving legal judgment summarization using logical structure and domain knowledge. *Journal of King Saud University Computer and Information Sciences* 37, 1 (March 2025), 3. doi:10.1007/s44443-025-00022-5

[11] Randy Goebel, Yoshinobu Kano, Mi-Young Kim, Juliano Rabelo, Ken Satoh, and Masaharu Yoshioka. 2024. Overview and Discussion of the Competition on Legal Information, Extraction/Entailment (COLIEE) 2023. *The Review of Socionetwork Strategies* 18, 1 (April 2024), 27–47. doi:10.1007/s12626-023-00152-0

[12] Yoshinobu Kano, Randy Goebel, Mi-Young Kim, Calum Kwan, Ken Satoh, Hiroaki Yamada, and Masaharu Yoshioka. 2026. An Overview of the COLIEE 2026 Competition: Legal Case Law and Statute Law Information Retrieval and Entailment. In *Proceedings of the 21st International Conference on Artificial Intelligence and Law (ICAIL 2026)*.

[13] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Proceedings of the 34th International Conference on Neural Information Processing Systems (NIPS '20)*. Curran Associates Inc., Red Hook, NY, USA, 9459–9474. https://dl.acm.org/doi/10.5555/3495724.3496517

[14] David Lillis. 2020. On the Evaluation of Data Fusion for Information Retrieval. In *Forum for Information Retrieval Evaluation (FIRE '20)*. Hyderabad, India. doi:10.1145/3441501.3441506

[15] Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024. Lost in the Middle: How Language Models Use Long Contexts. *Transactions of the Association for Computational Linguistics* 12 (2024), 157–173. doi:10.1162/tacl_a_00638

[16] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. doi:10.48550/arXiv.1907.11692 arXiv:1907.11692 [cs].

[17] Hoang-Trung Nguyen, Tan-Minh Nguyen, Xuan-Bach Le, Tuan-Kiet Le, Khanh-Huyen Nguyen, Ha-Thanh Nguyen, Thi-Hai-Yen Vuong, and Le-Minh Nguyen. 2025. NOWJ@COLIEE 2025: A Multi-stage Framework Integrating Embedding Models and Large Language Models for Legal Retrieval and Entailment. In *Proceedings of the Workshop on the Twelfth International Competition on Legal*

Information Extraction and Entailment (COLIEE 2025). Chicago, USA, 47–56.
[18] Rodrigo Nogueira, Zhiying Jiang, Ronak Pradeep, and Jimmy Lin. 2020. Document Ranking with a Pretrained Sequence-to-Sequence Model. In *Findings of the Association for Computational Linguistics: EMNLP 2020*. Association for Computational Linguistics, Online, 708–718. doi:10.18653/v1/2020.findings-emnlp.63
[19] Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Association for Computational Linguistics, Hong Kong, China, 3980–3990. doi:10.18653/v1/D19-1410
[20] Stephen Robertson and Hugo Zaragoza. 2009. The Probabilistic Relevance Framework: BM25 and Beyond. *Found. Trends Inf. Retr.* 3, 4 (April 2009), 333–389. doi:10.1561/1500000019
[21] Olga Shulayeva, Advaith Siddharthan, and Adam Wyner. 2017. Recognizing cited facts and principles in legal judgements. *Artificial Intelligence and Law* 25, 1 (March 2017), 107–126. doi:10.1007/s10506-017-9197-6
[22] Christopher C Vogt and Garrison W Cottrell. 1999. Fusion via a linear combination of scores. *Information Retrieval* 1, 3 (1999), 151–173. doi:10.1023/A:1009980820262
[23] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Proceedings of the 36th International Conference on Neural Information Processing Systems (NIPS '22)*. Curran Associates Inc., Red Hook, NY, USA, 24824–24837.
[24] Deiby Wu, Sarah Lawrence, and Behrooz Mansouri. 2025. AIIR Lab at COLIEE 2025: Exploring Applications of Large Language Models for Legal Text Retrieval and Entailment. In *Proceedings of the Workshop on the Twelfth International Competition on Legal Information Extraction and Entailment (COLIEE 2025)*. Chicago, USA, 112–118.
[25] Gechuan Zhang, David Lillis, and Paul Nulty. 2021. Can Domain Pre-training Help Interdisciplinary Researchers from Data Annotation Poverty? A Case Study of Legal Argument Mining with BERT-based Transformers. In *Proceedings of the Workshop on Natural Language Processing for Digital Humanities (NLP4DH)*. Association for Computational Linguistics, 121–130. https://aclanthology.org/2021.nlp4dh-1.14
[26] Gechuan Zhang, Paul Nulty, and David Lillis. 2023. Argument Mining with Graph Representation Learning. In *Proceedings of the Nineteenth International Conference on Artificial Intelligence and Law (ICAIL '23)*. Association for Computing Machinery, New York, NY, USA, 371–380. doi:10.1145/3594536.3595152
[27] Yucheng Zhou, Tao Shen, Xiubo Geng, Chongyang Tao, Can Xu, Guodong Long, Binxing Jiao, and Daxin Jiang. 2023. Towards Robust Ranker for Text Retrieval. In *Findings of the Association for Computational Linguistics: ACL 2023*, Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (Eds.). Association for Computational Linguistics, Toronto, Canada, 5387–5401. doi:10.18653/v1/2023.findings-acl.332

A Prompt for Legal Semantic Abstraction

For transparency and reproducibility, we provide below the exact prompt used to generate the structured case overviews in our implementation.

```
You are an expert Legal Case Analyst. Your goal is
     to extract key
information from the provided Canadian Federal
    Court judgment to help
find similar cases in a database.

**INSTRUCTIONS:**
Analyze the text and extract information into the
    following THREE sections.

1. **[ANCHORS]** (High Priority):
  - **KEYWORDS**: Extract or generate 5-7 distinct
      legal terms
    (e.g., "Family Class", "Adoption", "
        Procedural Fairness").
    *Crucial: If the case turns on the
        interpretation of a foreign
    country's law, MUST include the tag "Foreign
        Law Interpretation".*
  - **STATUTES**: List EXACT statute names and
      section numbers cited.

2. **[LOGIC]** (High Priority):
  - **Issue**: What is the primary legal question?
  - **Ratio**: The core reasoning for the decision
      .
    *Focus on the specific legal error or
        principle.
    Do NOT output generic "Standard of Review"
        boilerplate.*

3. **[FACTS]** (Medium Priority):
  - Provide a concise summary (max 50 words) of
      the factual background.
    Who are the parties? What triggered the
        dispute?

**CONSTRAINTS:**
- Output format must use the exact headers: [
    ANCHORS], [LOGIC], [FACTS].
- **CRITICAL**: You MUST write "[END]" immediately
     after the Facts
  section to signal completion.

**TEXT TO PROCESS:**
'''
{text}
'''

**FINAL CHECKLIST FOR YOU:**
1. Did you include [ANCHORS]?
2. Did you include [LOGIC]?
3. Did you include [FACTS]? (DO NOT FORGET THIS
    SECTION)
4. Did you end with [END]?

**OUTPUT:**
```

Author Index

Keyword Index