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

COLIEE 2025 Organizers

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

June 18, 2025

Preface

We are pleased to present this volume which contains the papers accepted for presentation at COLIEE 2025, which is the milestone twelfth Competition on Legal Information Extraction and Entailment (COLIEE 2025). As in previous years when the International Conference on Artificial Intelligence (ICAIL) is held, we are again joining the ICAIL 2025 conference as a companion workshop, to be held at Northwestern University in Chicago, from June 16 to 20, 2025.

As in recent COLIEE summaries, what began as only a handful of competitors from Japan and Canada, COLIEE has spread world wide. This year the COLIEE 2025 competition attracted 24 teams from 13 different countries, demonstrating the growing global interest in legal AI research. For the first two tasks on case law, eight teams submitted a total of 21 runs for Task 1, and six teams submitted a total of 18 runs for Task 2. For the statute law tasks, eight teams submitted a total of 22 runs for Task 3, and ten teams submitted a total of 29 runs for Task 4. For the pilot task on Tort law four teams submitted a total of 10 runs.

In addition to this year's task participants, the COLIEE organizing team has continued to extend and maintain the history of data used in the COLIEE competition, and repond to a growing number of requests to share that data, all confirmed for use in research projects around the world. We now have a dedicated team member to moderate access to these data sets (Calum Kwan), which are curated at a facility at the National Institute of Informatics in Tokyo. As originally intended, we have not only managed to grow and sustain the COLIEE competition but now provide access to one of the world's most valued fully annotated legal data corpora.

The COLIEE organizers would like to acknowledge the continued support of people and organizations around the planet, including Colin Lachance from Compass Law/Vlex, Juliano Rabelo from Jurisage, both in Canada, and to Young Yik Rhim of Intellicon in Seoul, who has been our advocate since the beginning of COLIEE.

In addition, we acknowledge our combined Japanese team, founder Ken Satoh (NII), Yoshinobu Kano (Shizuoka University), Masaharu Yoshioka (Hokkaido University), Hiroaki Yamada (Institute of Science Tokyo, Tokyo), and our Canadian team of Mi-Young Kim, Calum Kwan, and Randy Goebel (University of Alberta). These people have tirelessly contributed in such a collegial manner, to grow and sustain the COLIEE competition for twelve years.

Randy Goebel, University of Alberta, Canada
Yoshinobu Kano, Shizuoka Univesity, Japan
Calum Kawn, University of Alberta, Canada
Mi-Young Kim, University of Alberta, Canada
Juliano Rabelo, Jurisage, Canada
Ken Satoh, Center for Juris-informatics, Japan
Hiroaki Yamada, Institue of Science Tokyo, Japan
Masaharu Yoshioka, Hokkaido University, Japan
COLIEE 2025 organizers

Program Committee

Randy Goebel	University of Alberta
Yoshinobu Kano	Shizuoka University
Mi-Young Kim	Department of Computing Science, U. of Alberta, Canada
Miyoung-Test Kim	u of a
Makoto Nakamura	Niigata Institute of Technology
María Navas-Loro	UPM
Le-Minh Nguyen	Graduate School of Information Science, Japan Advanced Institute of Science and Technology
Juliano Rabelo	Jurisage
Julien Rossi	Amsterdam Business School
Ken Satoh	Center for Juris-Informatics, ROIS, Japan
Akira Shimazu	JAIST
Satoshi Tojo	Asia University
Vu Tran	The Institute of Statistical Mathematics, Japan
Sabine Wehnert	Leibniz Institute for Educational Media — Georg Eckert Institute
Hiroaki Yamada	Institute of Science Tokyo
Masaharu Yoshioka	Hokkaido University

Additional Reviewers

Kadowaki, Kazuma
Nguyen, Minh-Phuong
Zin, May Myo

Table of Contents

From TF-IDF to Instruction-Tuned LLMs: Hybrid Legal Reasoning Systems for COLIEE 2025

Euijin Baek
Jiayi Dai
H M Quamran Hasan
Yeji Kim
Housam Khalifa Bashier Babiker
Department of Computing Science
University of Alberta
Edmonton, Alberta, Canada
{euijin1,dai1,hmquamra,yeji7,khalifab}@ualberta.ca

Mi-Young Kim
Department of Science
University of Alberta
Camrose, Alberta, Canada
miyoung2@ualberta.ca

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

ABSTRACT

In this paper, we present our techniques applied by the UA team in the 2025 Competition on Legal Information Extraction and Entailment (COLIEE 2025). We participated in both retrieval and entailment tasks for case law and statute law. Our information retrieval approach achieved an unofficial ranking of 7th in Task 1. For Task 2, our best approach-combing language models with natural language inference and BM25 was ranked 14th. In Task 3, our model was ranked 17th for the retrieval task, while in Task 4 our approach using a language model for binary classification achieved 11th place.

CCS CONCEPTS

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

KEYWORDS

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

ACM Reference Format:
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 COLIEE 2025 workshop, June 20, 2025, Chicago, USA.* ACM, New York, NY, USA, 8 pages.

1 INTRODUCTION

The adoption of artificial intelligence in the legal domain is rapidly increasing, leading to the development and deployment of a wide range of tools. This surge is driven by the vast quantity of legal information available from sources including law courts, legislators, legal firms, as well as government and corporate documentation. To establish rigorous research and evaluation standards for AI models in this field, the Competition on Legal Information Extraction and Entailment (COLIEE) [9] was created. COLIEE aims to build a research community dedicated to addressing complex legal challenges, such as retrieving relevant case law, determining case law entailment, identifying and comparing legal arguments, and processing both statute and case law retrieval and entailment relationships.

Held annually, the competition provides a benchmark for assessing the latest developments in AI research applied to legal problems. In this paper, we present our approaches to the four main COLIEE tasks. Our methods integrate a variety of algorithms designed to address both entailment and retrieval challenges. Specifically, we use an integration of Large Language Models (LLMs), embedding-based retrieval techniques, and traditional natural language processing (NLP) methods, all aimed at improving the accuracy and efficiency of legal information processing. We also incorporate traditional information retrieval techniques like TF-IDF and BM25 for some tasks (e.g., Tasks 1, 2, and 3), while implementing a hybrid approach that combines these methods with transformer-based models.

This paper is organized as follows: In Section 2, we describe each task. In Section 3, we briefly discuss related work on information retrieval and entailment problems. Section 4 presents our approaches. Section 5 shows our experimental results. Finally, Section 6 concludes the paper and outlines future work.

2 TASK DESCRIPTION

2.1 Task 1

In Task 1, known as Legal Case Retrieval, the goal is to develop and evaluate reliable legal document retrieval methods. For a given query case, the aim is to retrieve candidate cases from the candidate pool, which are referenced by the query case. Since the number of candidate cases is not a constant value, post-processing plays a major role in ranking and shortlisting the appropriate retrieved candidates.

COLIEE 2025, June 2025, Chicago, USA

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

Task 2 focuses on legal case entailment. The goal is to identify specific paragraphs from noticed cases that entail a legal decision contained within a query case. The input consists of a pair: a new case with a decision fragment, and paragraphs from noticed cases. The task is to identify which paragraphs in the noticed cases support the decision fragment from the query case. As noted above, this task ultimately requires the identification of legal arguments and their relationships; so the initial approach is to create approximations of such argument processing with efficient NLP methods.

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

2.3 Task 3

Task 3 aims to retrieve relevant statute law articles from a database of Japanese statutes, given queries from Japanese legal bar exam questions. The evaluation of the retrieval models in Task 3 has three aspects: Precision, Recall and F2 scores, which are separately defined as follows:

$$\text{Precision} = \frac{TP}{TP + FP}$$
$$\text{Recall} = \frac{TP}{TP + FN}$$
$$F_2\text{-measure} = \frac{5 \times Precision \times Recall}{4 \times Precision + Recall}$$

where TP, FP and FN are defined as in Task 1 evaluation. The design of the F_2 puts more weight on Recall.

2.4 Task 4

Task 4 requires building a fully automated system that answers yes/no questions about Japanese statute law, by performing textual entailment between retrieved Japanese Civil Code articles (from Task 3) and given yes/no queries from the Japanese bar exam. Performance is measured by how accurately the system answers "Yes" or "No" for each previously unseen query in the test set.

3 RELATED WORK

3.1 Task 1

With the widespread adoption of transformer-based deep learning (DL) and Large language Models, a plethora of techniques have emerged in the field of Information Retrieval. Generative Transformer models like LEGAL-BERT [3] 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, [10] 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 [11] achieves significant improvements in legal case relevance assessment by incorporating a structural understanding of legal documents. Its structure-aware design makes it particularly suitable for legal applications where case similarity is determined by nuanced legal reasoning. These developments lay the foundation of our approach towards the information retrieval tasks in COLIEE 2025.

3.2 Task 2

Many recent COLIEE competition teams have adopted hybrid approaches that combine traditional information retrieval (IR) methods with other methods, including transformer-based models [1, 25]. For example, classical ranking models such as BM25 [24] have been widely used due to their efficiency and simplicity, making them effective for downsizing large candidate sets. However, BM25 relies on simple keyword-based matching and may lead to the omission of semantically relevant paragraphs that unintentionally lack lexical overlap with the query fragment [24]. To address this limitation, many teams have integrated BM25 with re-ranker models or entailment classifiers based on transformer architectures, allowing the system to benefit from both efficient retrieval and deeper semantic understanding. In COLIEE 2024, the AMHR [19] team integrated monoT5 into a re-ranking pipeline and fine-tuned it, achieving top performance. Similarly, other teams adopted multilingual BERT and DeBERTa with fine-tuning for legal entailment classification tasks [22], thus demonstrating that such models can capture deeper semantic relationships than traditional IR techniques [2]. Moreover, the emergence of large-scale instruction-tuned language models has significantly expanded the scope of NLP applications, allowing them to perform a wide range of tasks [29]. Several studies have shown that supervised fine-tuning of LLMs on task-specific data can further enhance their performance [27].

3.3 Task 3

In Task 3, a variety of tools have been adopted to help understand the semantics of the legal textual information, including bag-of-words and large language model-based methods. The overall methodology of high performance COLIEE competition submissions follows the general workflow of combining an initial retrieval using simpler/smaller models, followed by further re-ranking with more sophisticated/larger models.

Bag-of-words methods, such as the syntactic methods of TF-IDF and BM25, have been used primarily to retrieve an initial set of top-k candidate articles [15, 16, 20]. Language models, e.g., various pre-trained BERT and T5 models, are often used to either retrieve or further re-rank the initial candidates. In COLIEE 2024, CAPTAIN [15] and JNLP.constr-join [16] used ensembles of multiple fine-tuned model checkpoints for possibly better capturing data diversity. JNLP.Mistral [16] used Sentence-BERT [23] for similarity score-based ranking and then prompted the candidates to Mistral 7B [8]

for re-ranking. Similarly, JNLP.constr-join [16] used a fine-tuned Tohoku BERT model to first retrieve a set of high-recall candidates, and then used Orca-2 13B (https://huggingface.co/microsoft/Orca-2-13b) and Qwen 14B (https://huggingface.co/Qwen/Qwen-14B-Chat) for further fine-grained ranking.

3.4 Task 4

For Task 4, which focuses on yes/no legal entailment within the Japanese Civil Statute Code, multiple teams have explored prompt-based Large Language Model (LLM) strategies to handle the unique challenges of legal reasoning in a binary classification format.

In COLIEE 2024, CAPTAIN [17] builds on few-shot prompting, Auto-Chain-of-Thought (Auto-CoT), and data augmentation for a hybrid method to refine entailment judgment. Few-shot prompting leverages minimal labeled examples directly in the prompt, which helps the model generalize legal logic from demonstrations. Auto-CoT systematically generates and then incorporates intermediate reasoning chains into the final prompt, thus illustrating how the model arrived at its conclusion. Additionally, CAPTAIN employs data augmentation to mitigate limited training examples, by generating synthetic samples to enhance model robustness. NOWJ [18] focuses on prompt collection and answer extraction. It uses pre-trained models (e.g., Panda-7B-v0.1 and Flan-T5-XL) and a legally oriented IRAC (Issue, Rule, Application, Conclusion) structure in the prompts, which guides the model to express legal reasoning steps more clearly. Finally, the AMHR[19] team introduces so-called "Mixture of Expert" models (MoE). Their pipeline runs multiple "expert" prompts on the training set and then reevaluates the aggregated outputs via an additional prompt. This additional prompt interprets each expert's response and resolves ambiguities by assigning weighted voting scores. This approach is a new hybrid integration of methods that attempts to capture a broader spectrum of legal reasoning patterns.

4 OUR METHOD

4.1 Task 1

In this section, we present our solution for Task 1. We followed a strategy similar to last year's winning team, TQM [12], when it came to the pre-processing and post-processing steps. Initially, we removed everything before the "[1]" character in the document, as it contains procedural details. Then, we removed any references enclosed in square brackets, and any XML/HTML tags. Following that, we fed the document in chunks to Google Translate [6] to obtain the English translation in those cases for documents that were originally in French (note that Tasks 1 and 2 use Canadian federal case law, and Tasks 3 and 4 use Japanese civil statute law). Although several recent LLMs are capable of handling multilingual data, in our approach for Task 1, we leaned towards TF-IDF as the primary approach, and due to its lack of understanding of semantics and context, we chose to translate all text to English. Summaries were also generated for each of the pre-processed documents using a Qwen2-7B [28] model. The summaries were restricted to 200 words, with the model running on a system containing 2 NVIDIA A100 80GB GPUs.

For retrieval, we used the aforementioned TF-IDF measure, which provides a numerical statistic to evaluate the importance of a word to a document in a corpus. The TF-IDF score [5] for a term t in document d is calculated using the formula:

$$\text{TF-IDF}(t, d) = \text{TF}(t, d) \times \text{IDF}(t, D)$$

Term Frequency (TF) measures how frequently a term t appears in document d:

$$\text{TF}(t, d) = \frac{f_{t,d}}{\sum_{t',d} f_{t',d}}$$

where $f_{t,d}$ is the raw count of the term t in a document d. Inverse Document Frequency (IDF) evaluates how unique or rare a term is across all documents in the corpus D:

$$\text{IDF}(t, D) = \log\left(\frac{|D|}{1 + |\{d \in D : t \in d\}|}\right)$$

Initially, we converted the pre-processed texts to TF-IDF vectors (with 1-3 word n-grams), and then computed the cosine similarity between a query and all documents in the corpus, then returned the top 40 most similar documents for the query. The candidate set was further refined by calculating the cosine similarities of their summaries with the query summary, and then re-ranking and shortlisting the initial retrieval to 20 candidates using a cosine similarity threshold of 0.05. We experimented with several values in the range [0, 0.1] and selected 0.05, as it yielded the best results. If no candidates meet the similarity threshold, we fall back to the original list.

Subsequently, following [12], we filtered the candidate cases that have trial dates *after* the query case's trial date. This is a logical approach because a case cannot cite future cases. We assumed that the latest date in each document was the trial date for that particular case. This was confirmed using a Qwen2-7B [28] model, where we prompted the model to extract the latest date in yyyy-mm-dd format. In accordance with the constraint that a query case cannot serve as a candidate case for other queries [12], we removed any query case from the pool of retrieved candidates if it appears there. Since the average number of candidates retrieved per query case in the Task 1 training set is 4.10, we apply a dynamic threshold to our candidate list: we retain the top 50% when the number of candidates retrieved is greater than 10, to obtain an average of 5.33 candidates per query. We chose this approach because of the large variation in the similarity scores in the candidate set, where a similarity threshold would not have been effective.

4.2 Task 2

Task 2 is a multi-label training problem where a decision fragment may be entailed by multiple paragraphs; we first reformulated it into a binary classification task. By treating each (fragment, paragraph) pair as an independent instance, our approach minimizes the risk of omitting semantically relevant paragraphs that might be overlooked by fixed top-k retrieval methods [21]. However, this also introduces a severe class imbalance, as only a small fraction of candidate paragraphs are truly entailing. To address this imbalance, we tried two strategies: (1) generating synthetic positive examples using instruction-tuned LLMs, and (2) downsizing the training data by filtering out obviously irrelevant candidates using BM25 [24].

Before training or testing, we ensure input consistency by translating all French paragraphs into English using the Deep Translator

library's Google Translate API [4]. Language detection was performed using both the langdetect and langid Python libraries. While some LLMs such as Qwen support multilingual input, most of the models used in this work including LLaMA, DeBERTa, and BM25 are primarily English-based. To ensure consistent performance across components, we translated all non-English text into English prior to processing. Additionally, repeated bracketed numeric markers (e.g., "[2]", "[3]") beyond their first occurrence were removed to reduce potential confusion during tokenization and model fine-tuning.

Our overall system integrates three components: (1) large language models (LLMs) - Qwen2.5-14B Instruct [1] and LLaMA-3.1-8B Instruct [2]; (2) an NLI-specialized model, DeBERTa v3 base [3]; and (3) a traditional IR system using BM25 (BM25Okapi).

The motivation is as follows: first, LLMs such as Qwen and LLaMA are pretrained on vast corpora, which provides extensive contextual understanding that is essential for the nuanced language of legal documents [26, 28]. Second, DeBERTa v3 base excels in NLI tasks, making it well-suited for refining candidate selections [7]. Third, BM25 offers a computationally efficient method to downsize large candidate sets by filtering out clearly irrelevant paragraphs through keyword-based matching [14, 24]. In our system, BM25 is used not only for filtering out the irrelevant paragraphs, but also as a fallback mechanism during inference. When neither the LLMs nor DeBERTa returns a confident entailment prediction, BM25's top-ranked paragraph is used to ensure that potentially relevant candidates are not entirely missed.

Based on internal validation results, the ensemble's prediction priority was fixed as Qwen → LLaMA → DeBERTa → BM25, in descending order of observed performance. Each candidate paragraph was first processed by Qwen; if Qwen returned an entailment prediction, the result was accepted without evaluating the remaining models. If Qwen failed to produce a prediction, we used LLaMA, followed by DeBERTa, and finally BM25. This sequential model invocation was designed to prioritize the most accurate models while ensuring fallback coverage in uncertain cases. In all submissions, each candidate paragraph undergoes five independent inference passes by LLMs, and a majority vote across these runs determines the final prediction. Specifically, for Qwen and LLaMA, we generate five prediction outputs per paragraph (temperature=0.95, top_p=0.7) to account for sampling variation. The final label is then decided by majority voting across these five generations. This multi-inference voting strategy helps mitigate the inherent randomness of single-run outcomes, particularly for borderline cases.

For the first run (submission1.txt), we generated synthetic positive examples using LLaMA-3.1-8B Instruct. Specifically, we aimed to construct a synthetic training dataset with an approximate 10:1 ratio between negative and positive samples. After data synthesis, both Qwen and LLaMA were fine-tuned on this augmented dataset. DeBERTa was excluded from this submission to simplify the model pipeline. This design choice also enabled greater architectural diversity across our three submissions. For the second run (submission2.txt), we applied BM25 to select the top-10 most relevant paragraphs per fragment, effectively downsizing the training data. We explicitly included all true positives in the filtered set, in case they were not ranked within the BM25 top-10. The resulting dataset, which contains all true positives and a reduced set of negatives, was then used to fine-tune Qwen, LLaMA, and DeBERTa. To ensure efficient training under limited computational resources, we applied Low-Rank Adaptation (LoRA) for fine-tuning all transformer models. At inference time, DeBERTa reranks the LLM candidates and selects at most the top two paragraphs for submission, as the average number of gold label entailing paragraphs per case was 1.4 in the COLIEE 2024 test set. For the third run (submission3.txt), we used the same training pipeline as the second run, but modified the post-processing step. We used predefined score threshold on DeBERTa's prediction score to allow for a variable number of entailed paragraphs.

4.3 Task 3

We made three submissions for Task 3, which is the Japanese statute law retrieval task. In the first submission, we used an ensemble of BM25 and a sentence transformer model [23] (i.e., all-MiniLM-L6-v2) to perform hybrid retrieval. BM25 provided article relevance scores based on syntactic signals, such as word frequency and document length, and the sentence transformer was used to extract text embedding vectors for the articles and queries for computing cosine similarity scores. The results from both methods were re-ranked using a weighted combination of their scores (i.e., 0.1 for BM25 and 0.9 for the language model) to produce the final ranked list of articles retrieved for each query. In the second and third submissions, we relied solely on the text embedding vectors from two pre-trained language models to explore improved semantic representations of legal information. The two sentence transformer language models were, respectively, gte-large[4] [13] and all-mpnet-base-v2[5]. They were used, respectively, in the second and the third submissions, to obtain text embeddings for later cosine similarity computation, as in the first submission.

4.4 Task 4

For Task 4, the yes/no Japanese bar law exam questions, we use the **meta-llama/Meta-Llama-3-70B** [6] model (updated April 18, 2024), leveraging its strong multilingual capabilities and high parameter capacity to capture nuanced legal text. We adapt the model in two phases: (1) domain-focused pre-training and (2) task-specific fine-tuning, thus leveraging Low-Rank Adaptation (LoRA) to control computation overhead. We used the legal corpus from *civil_code_en-1to724-2.txt* provided by COLIEE, for pre-training and converted it following a "pre-training dataset" schema:

```
[ { "text": 'Article 1: ... <article content> ... " },
{ "text": "Article 2: ... <article content> ... "}, ... ]
```

. The objective here was to further adapt the base model's distribution to legal language, ensuring it captures domain-specific terminology, phrasing, and context beyond generic text corpora. After pre-training, we constructed an instruction-tuning dataset aligned with the Alpaca/Stanford format.[7] We constructed our instruction-tuning dataset solely from the COLIEE 2025 statute-law training

[1] https://huggingface.co/Qwen/Qwen2.5-14B-Instruct
[2] https://huggingface.co/meta-llama/Llama-3.1-8B-Instruct
[3] https://huggingface.co/microsoft/deberta-v3-base
[4] https://huggingface.co/thenlper/gte-large
[5] https://huggingface.co/sentence-transformers/all-mpnet-base-v2
[6] https://ai.meta.com/blog/meta-llama-3/
[7] https://github.com/tatsu-lab/stanford_alpaca

set. This dataset focuses on *yes/no* queries derived from bar exam-style questions or hypothetical legal statute scenarios. Each record consists of:

- `instruction`: A short directive, e.g., *Given the article(s) below, decide if the answer is "Yes" or "No."*
- `input`: The actual question or legal scenario (e.g., *"An unborn child may not be given a gift on the donor's death."*)
- `output`: The correct label, either *"Yes"* or *"No."*

By framing the classification task in an instruction-style format, we hope to encourage the LLM to treat each article-and-query pair as a self-contained prompt–response conversation. To keep GPU usage manageable, we employ LoRA to fine-tune a subset of model parameters instead of performing full-parameter updates. We used a rank of 8, a learning rate of 1e−4, 2 epochs, and a batch size of 4 across 4 NVIDIA A100 80GB GPUs.

During inference, we generate 15 sampled responses (temperature=0.6, top_p = 0.9) for each query and use majority voting to obtain the final "Yes/No" label. This approach mitigates single-pass randomness and yields a more stable prediction. Although all three submissions (UA1, UA2, and UA3) employ the same overall pipeline—domain pre-training, instruction-based fine-tuning, and majority-vote inference, each LLM inference pass is based on sampling (temperature, top-p); thus different runs can yield slightly different predictions. Consequently, these "different" attempts can produce small but noticeable performance variations.

5 RESULTS

Here, we summarize the experimental results of all tasks.

5.1 Task 1

For Task 1, we made three official submissions and one unofficial submission. However, a bug in the code used for our official submissions resulted in invalid outputs. After identifying this issue, we conducted an additional unofficial submission to more accurately assess the true performance of our approach.

The results indicate better performance on the test set compared to our validation set, which consisted of 78 random queries selected from the 2025 training set. On the validation set, the approach yielded scores of 0.0986, 0.0793, and 0.1304 for F1, precision, and recall, respectively. Our unofficial submission indicates abundant room for improvement, which we speculate is a consequence of weak semantic understanding which is a key aspect for information retrieval.

5.2 Task 2

Table 2 presents the official results for Task 2 in COLIEE 2025, including our team's three runs (UA1, UA2, and UA3). On the internal validation set, UA2 exhibited the most stable performance. However, on the official test set, UA3 slightly outperformed UA2 in terms of F1 score. This suggests that the threshold-based selection mechanism of UA3 may provide greater flexibility.

Overall, the results indicate that there is still room for improvement. In particular, the performance gap between validation and test sets suggests that our current models may struggle with generalizing to account for varied paragraph structures.

Table 1: Task 1 Results. Results marked with * are unofficial submissions.

Team	F1	Precision	Recall
JNLP	0.3353	0.3042	0.3735
JNLP	0.3267	0.2945	0.3667
UQLegalAI	0.2962	0.2908	0.3019
UQLegalAI	0.2957	0.2903	0.3013
UQLegalAI	0.2940	0.2886	0.2996
AIIR Lab	0.2171	0.2040	0.2319
UA*	0.2073	0.1892	0.2291
NOWJ	0.1984	0.1670	0.2445
AIIR Lab	0.1879	0.2317	0.1580
AIIR Lab	0.1872	0.2308	0.1575
NOWJ	0.1708	0.1605	0.1825
NOWJ	0.1580	0.1485	0.1688
JNLP	0.1597	0.1307	0.2052
OVGU	0.1498	0.1743	0.1313
UB_2025	0.1363	0.1955	0.1046
UB_2025	0.1171	0.1818	0.0864
UB_2025	0.1051	0.0572	0.6379
SIL	0.0058	0.0054	0.0063
UA	0.0000	0.0000	0.0000
UA	0.0000	0.0000	0.0000
UA	0.0000	0.0000	0.0000
OVGU	0.0000	0.0000	0.0000

Table 2: Task 2 Results

Team	F1	Precision	Recall
NOWJ_Task2	0.3195	0.3788	0.2762
NOWJ_Task2	0.2865	0.2976	0.2762
NOWJ_Task2	0.2782	0.2650	0.2928
OVGU	0.2454	0.2759	0.2210
JNLP_task_2	0.2412	0.2000	0.3039
JNLP_task_2	0.2400	0.2708	0.2155
AIIR_Lab	0.2368	0.2927	0.1989
AIIR_Lab	0.2229	0.2632	0.1934
OVGU	0.1965	0.2692	0.1547
AIIR_Lab	0.1930	0.2050	0.1823
Task2_CAPTAIN	0.1882	0.2547	0.1492
Task2_CAPTAIN	0.1812	0.2453	0.1436
JNLP_task_2	0.1779	0.2500	0.1381
UA3	0.1778	0.2090	0.1547
Task2_CAPTAIN	0.1712	0.2252	0.1381
UA2	0.1712	0.2252	0.1381
OVGU	0.1708	0.2400	0.1326
UA1	0.1736	0.2077	0.1492

5.3 Task 3

Table 3 presents the official results for Task 3 in COLIEE 2025, including our team's three runs (UA-gte, UA-mpnet, and UA-bm25_allMini). The two methods that used only the language models, gte-large and all-mpnet-base-v2, achieved similar predictive performance with F2 scores of 0.2426 and 0.2377, respectively. The method that

used an ensemble of BM25 and all-MiniLM-L6-v2 gave a lower F2 score at 0.1978. In addition, we generally observed that using a smaller ensemble weight on the BM25 scores, the overall relevance estimation performance was higher. That is simply relying on the language model, i.e., making the weight on BM25 equal to 0, performs better than incorporating BM25 with the language model. Compared with other teams' submissions in Table 3, our methods achieved relatively low recall scores suggesting that pre-trained generic text embedding models may not be sufficient for detecting relevant law articles and that further improvements could be learning language models that are more domain-specific to the legal data. In addition, our methods returned low precision, relative to the recall, suggesting that more sophisticated re-ranking could be helpful in improving the relevance estimation of truly relevant articles.

Table 3: Task 3 Results

Team	F2	Precision	Recall
JNLP_RUN1	0.7829	0.7521	0.8184
CAPTAIN.H2	0.7769	0.7799	0.797
CAPTAIN.H3	0.7678	0.7489	0.8034
CAPTAIN.H1	0.7583	0.7671	0.7778
JNLP_RUN2	0.7359	0.6806	0.7863
JNLP_RUN3	0.7357	0.6944	0.7735
INFA	0.6474	0.7179	0.6389
mpnetAIIRLab	0.6246	0.3333	0.8291
mistralRerank	0.5672	0.3034	0.7521
OVGU3	0.5654	0.594	0.5748
OVGU2	0.5577	0.5705	0.5641
NVAIIRLab	0.5554	0.2863	0.7479
UIwa	0.5443	0.5481	0.5513
UImeta	0.5422	0.5417	0.5513
UIthr	0.5356	0.5641	0.5321
OVGU1	0.4372	0.4338	0.4487
UA-gte	0.2426	0.0949	0.4145
UA-mpnet	0.2377	0.0923	0.4081
UA-bm25_allMini	0.1978	0.0744	0.3462
NOWJ.H1	0.0128	0.0128	0.0128
NOWJ.H2	0.0128	0.0128	0.0128
NOWJ.H3	0.0128	0.0128	0.0128

5.4 Task 4

Table 4 summarizes the performance for Task 4 (Japanese statute-law entailment), where the baseline correctly answered 38 out of 74 questions (accuracy 0.5135). Our submissions are **UA1**, **UA2**, and **UA3**. Among these, **UA2** and **UA3** both correctly answered 58 out of 74 questions (accuracy 0.7838), tying for 11th–12th place overall, while **UA1** attained 56 correct (accuracy 0.7568). We see that UA2 and UA3 outperform UA1. Given that the fundamental architecture itself did not change drastically across the three submissions, these differences appear to be largely attributable to randomness in LLM sampling.

In our internal validation (using *riteval_R05_en.xml* as a local test set), we achieved around 84% accuracy using the instruction-tuned model without domain pre-training. Adding domain pre-training nudged it slightly higher (to about 86%), but results still varied from run to run. By applying majority voting over 15 sampled outputs per query, we maintained a relatively stable 84% in repeated experiments. Although our official test performance ranged from 75.68% to 78.38%, notably lower than the 84% we consistently observed during local trials. We attribute the discrepancy primarily to unobserved complexities in the official test questions.

Table 4: Task 4 Results.

Team	Correct	Accuracy
BaseLine	38	0.5135
KIS3	67	0.9054
KIS1	65	0.8784
LUONG01	64	0.8649
UIRunCot	63	0.8514
KIS2	63	0.8514
CAPTAIN2	60	0.8108
JNLP002	60	0.8108
JNLP003	59	0.7973
CAPTAIN1	58	0.7838
CAPTAIN3	58	0.7838
UA2	58	0.7838
UA3	58	0.7838
JNLP001	57	0.7703
KLAP.H2	57	0.7703
UA1	56	0.7568
NOWJ.run1	55	0.7432
NOWJ.run2	55	0.7432
NOWJ.run3	55	0.7432
OVGU1	55	0.7432
RUG_V1	49	0.6622
KLAP.H1	48	0.6486
OVGU3	47	0.6351
RUG_V3	46	0.6216
AIIRLlaMA	45	0.6081
OVGU2	45	0.6081
RUG_V2	45	0.6081
AIIRMistral	42	0.5676

5.5 Discussion

Based on our implementations using both LLMs and traditional methods such as TF-IDF and BM25, we can summarize our discussion as follows. For Task 1, using TF-IDF resulted in more reliable performance in the information retrieval task leveraging term frequency; however, this approach lacks semantic depth because it ignores contextual information, suggesting the need for a hybrid approach. For task 2, employing an LLM-based approach as a proxy for binary classification enabled the capture of deeper semantic relationships and reduced the risk of missing relevant paragraphs, although its generalization was limited due to variations in paragraph structure. In task 3, the TF-IDF resulted in lower precision, indicating the need for more domain-specific models and re-ranking techniques. Finally, for task 4, LLM-based models achieved robust performance in statute-law queries when majority voting was used

to address randomness in generation, but overall performance was somewhat lower, likely due to the increased complexity and poor generalization to unseen queries. Furthermore, the observed differences between our submissions (UA1, UA2, and UA3) suggest that minor variations in sampling randomness, although majority voting was employed, can lead to measurable performance shifts.

6 CONCLUSION

We have presented our techniques for the COLIEE 2025 competition using various approaches. Our findings highlight the trade-offs among methods and suggests the need for a hybrid approach that combines strategies such as frequency-based methods, semantic processing, and re-ranking. In addition, the suboptimal performance of generic LLMs in legal text representation demonstrates the need for more domain-specific tuning.

In future work, we will focus on developing more stable inference approaches and better fine-tuning techniques to address the generalization problem. We also plan to investigate hybrid strategies that integrate semantic models with traditional techniques, as well as to improve pre-processing and post-processing. For example, we aim to explore better approaches for dynamic thresholding and text segmentation. Although a quick comparison showed only a modest ~2 pp accuracy lift from continual pre-training (84 % → 86 %), the gain was not yet consistent across runs. We therefore intend to run targeted ablations on pre-training length and tuning method to pinpoint when continual domain pre-training really pays off. Concretely, we will checkpoint the model at multiple cut-offs (e.g., every 5 K and 10 K steps) to detect early saturation and compare full-parameter updates against parameter-efficient schemes such as LoRA or adapters to balance accuracy with compute cost.

ACKNOWLEDGEMENTS

This research was supported by the Alberta Machine Intelligence Institute (Amii), the University of Alberta, the Natural Sciences and Engineering Research Council of Canada (NSERC), [funding reference numbers DGECR-2022-00369 and RGPIN-2022-0346], and Alberta Innovates.

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. In *Proceedings of the COLIEE Workshop in ICAIL*.

[2] Ilias Chalkidis and ... 2020. LEGAL-BERT: The Muppets straight out of Law School. In *Findings of the Association for Computational Linguistics: EMNLP 2020*.

[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.CL] https://arxiv.org/abs/2010.02559

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

[5] Ethen. [n. d.]. TF-IDF: Term Frequency-Inverse Document Frequency. https://ethen8181.github.io/machine-learning/clustering_old/tf_idf/tf_idf.html Accessed: 2025-03-27.

[6] Google. n.d.. Google Translate. https://translate.google.com. Accessed: May 20, 2025.

[7] Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. DeBERTa: Decoding-Enhanced BERT with Disentangled Attention. In *Proceedings of the 9th International Conference on Learning Representations (ICLR 2021)*.

[8] Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7B. arXiv:2310.06825 [cs.CL] https://arxiv.org/abs/2310.06825

[9] 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) 2021. *Journal of Review of Socionetwork Strategies* 16(1) (2022), 111–133.

[10] Arfiani Nur Khusna and Indri Agustina. 2018. Implementation of Information Retrieval Using Tf-Idf Weighting Method On Detik.Com's Website. In *2018 12th International Conference on Telecommunication Systems, Services, and Applications (TSSA)*. 1–4. https://doi.org/10.1109/TSSA.2018.8708744

[11] 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:2304.11370 [cs.IR] https://arxiv.org/abs/2304.11370

[12] 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

[13] Zehan Li, Xin Zhang, Yanzhao Zhang, Dingkun Long, Pengjun Xie, and Meishan Zhang. 2023. Towards General Text Embeddings with Multi-stage Contrastive Learning. arXiv:2308.03281 [cs.CL] https://arxiv.org/abs/2308.03281

[14] Chia-Hsin Lu et al. 2021. Combining Text Retrieval and Deep Neural Networks for Open-Domain Question Answering. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics (ACL 2021)*.

[15] Chau Nguyen, Phuong Nguyen, Thanh Tran, Dat Nguyen, An Trieu, Tin Pham, Anh Dang, and Le-Minh Nguyen. 2024. CAPTAIN at COLIEE 2023: Efficient Methods for Legal Information Retrieval and Entailment Tasks. arXiv:2401.03551 [cs.CL] https://arxiv.org/abs/2401.03551

[16] 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 *New Frontiers in Artificial Intelligence*, Toyotaro Suzumura and Mayumi Bono (Eds.). Springer Nature Singapore, Singapore, 167–182.

[17] 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 *Lecture Notes in Computer Science*. Springer Nature Singapore, Singapore, 125–139.

[18] 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 *Lecture Notes in Computer Science*. Springer Nature Singapore, Singapore, 183–199.

[19] A. Nighojkar et al. 2024. AMHR COLIEE 2024 Entry: Legal Entailment and Retrieval. In *Proceedings of the Eighteenth International Workshop on Juris-Informatics (JURISIN 2024)*.

[20] 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*, Toyotaro Suzumura and Mayumi Bono (Eds.). Springer Nature Singapore, Singapore, 200–211.

[21] Juliano Rabelo et al. 2018. Legal Information Extraction and Entailment for Statute Law and Case Law. In *Proceedings of the 11th International Workshop on Juris-Informatics (JURISIN 2018)*.

[22] Rafael Rabelo, Randy Goebel, et al. 2024. Overview of Benchmark Datasets and Methods for the Legal Information Extraction/Entailment Competition (COLIEE) 2024. In *Proceedings of the Eighteenth International Workshop on Juris-Informatics (JURISIN 2024)*.

[23] Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. arXiv:1908.10084 [cs.CL] https://arxiv.org/abs/1908.10084

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

[25] Frank Schilder, Dhivya Chinnappa, Kanika Madan, Jinane Harmouche, Andrew Vold, Hiroko Bretz, and John Hudzina. 2021. A Pentapus Grapples with Legal Reasoning. In *Proceedings of the COLIEE Workshop in ICAIL*.

[26] Hugo Touvron et al. 2023. LLaMA: Open and Efficient Foundation Language Models. *arXiv preprint arXiv:2302.13971* (2023).

[27] Jason Wei, Maarten Bosma, Vincent Y Zhao, et al. 2021. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652* (2021). https://arxiv.org/abs/2109.01652

[28] An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jianxin Yang, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Xuejing Liu, Yang Fan, Yang Yao, Yichang Zhang,

Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, Zhifang Guo, and Zhihao Fan. 2024. Qwen2 Technical Report. arXiv:2407.10671 [cs.CL] https://arxiv.org/abs/2407.10671

[29] Wayne Xin Zhao, Kun Zhang, Jingyuan Liu, Zhicheng Wang, Yiqiao Hou, Junlin Xie, Zihan Tang, Zhaochun Du, and Ji-Rong Wen. 2023. A survey of large language models. *arXiv preprint arXiv:2303.18223* (2023). https://arxiv.org/abs/2303.18223

SIL@COLIEE 2025: A Cascading Framework for Finding Relevant Case Laws

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

Pooja Harde
National Institute of Technology
Kurukshetra
Haryana, India
pmharde29@gmail.com

Taha Sadikot
National Institute of Technology
Kurukshetra
Haryana, India
19bmiit087@gmail.com

Eric Namit Kujur
National Institute of Technology
Kurukshetra
Haryana, India
524110005@nitkkr.ac.in

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

Abstract

Legal case retrieval plays a crucial role in modern judicial systems, ensuring efficient access to relevant precedents. This paper presents a methodology for legal case retrieval, which integrates lexical and semantic retrieval techniques. Our approach begins by utilizing a retrieval strategy where MPNet vector similarity scores are used to select the top k candidate documents for each query, thereby reducing the search space. Subsequently, we extract nine distinct features from each query-document pair and leverage an LTR (Learning To Rank) model to predict their relevance scores. A predefined threshold is then applied to determine the final set of relevant documents.

CCS Concepts: • **Computing methodologies → Information extraction**.

Keywords: Legal Case Retrieval, MPNet, Learning to rank, FAISS Index

ACM Reference Format:
Bhavya Jain, Pooja Harde, Taha Sadikot, Eric Namit Kujur, and Sarika Jain. 2025. SIL@COLIEE 2025: A Cascading Framework for Finding Relevant Case Laws. In *Proceedings of COLIEE 2025 workshop, June 20, 2025, Chicago, USA*. ACM, New York, NY, USA, 5 pages.

1 Introduction

Legal case retrieval is a fundamental task in the judicial system, aiding legal professionals in identifying relevant precedents to support legal arguments and decisions. With the rapid expansion of legal databases, the need for efficient and accurate retrieval systems has become increasingly important. The Competition on Legal Information Extraction/Entailment (COLIEE) has emerged as a significant platform for advancing the state-of-the-art in legal information processing and retrieval. Traditional retrieval methods, such as lexical matching with BM25, often struggle to capture the complex semantic relationships inherent in legal texts. On the other hand, deep learning-based retrieval methods, while effective, require significant annotated data and computational resources. We are the SemIntLab group, and we provide here with the SIL methodology that we experimented with while participating in COLIEE 2025. We propose a hybrid legal retrieval approach that combines lexical and semantic retrieval techniques. Given the large number of candidate cases for every query case, the SIL team decided to employ a cascading framework to avoid high computational costs. We arrange a multistage pipeline with constructing the indexes in the first stage, then reducing the search space by retrieving top k candidates for every query case, and finally re-ranking the relevant document by LTR (Learning To Rank) model. Our methodology begins with an initial retrieval stage that ranks documents based on the similarity scores of MPNet embeddings, selecting the top 100 candidate documents for each query. Subsequently, we extract nine key features from each query-document pair and employ an LTR model to predict their relevance scores. These features include query length, document length, the number of references in the query, the number of references in the document, BM25 score, query likelihood (QLD) score, and Doc2Vec similarity. A predefined threshold is then applied to determine the final set of relevant documents. This multi-stage approach ensures that both surface-level term matching and deep semantic similarity are effectively captured, while the LTR model refines relevance judgments through supervised learning.

COLIEE 2025, June 20, 2025, Chicago, USA

By integrating retrieval strategy and leveraging feature-based learning-to-rank techniques, our approach seeks to enhance the accuracy and efficiency of legal case retrieval. This study contributes to the ongoing advancements in legal information retrieval by bridging traditional and modern retrieval paradigms for more reliable legal case retrieval systems.

The paper is structured as follows. Section 2 describes the task 1 of the COLIEE 2025 workshop in which we have participated. Section 3 consists of existing works for legal case retrieval and section 4 describes our SIL methodology and the results. Finally, the conclusion and suggestions for future work are provided in Section 5.

2 Problem Statement

Legal Case Retrieval (LCR) is primarily an Information Retrieval (IR) task but comes with its own set of challenges. The task aims to develop effective and reliable legal document retrieval systems. In this context, the cases referenced by a query case are called *noticed cases*, which serve as decision-supporting cases for that query. The objective is to input a query case and retrieve all the noticed cases from a given collection, focusing on measuring how accurately the system captures the relevant supporting cases. [9]

Unlike traditional keyword-based document search, LCR demands an understanding of how legal cases are connected and cited. The goal is to enhance the accuracy and relevance of search results by capturing semantic and referential relationships between cases.

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 a subset of relevant cases $S = \{r_1, r_2, \ldots, r_k \mid r_i \in C \wedge \text{support}(r_i, q)\}$, where $\text{support}(r_i, q)$ indicates that the case r_i supports the query case q in at least one aspect.

Data Corpus

Corpora commonly used in LCR systems span across multiple jurisdictions, including India, Canada, and China. For the COLIEE 2025 LCR task, a corpus comprising Federal Court of Canada case laws has been provided. All query and noticed cases are presented as a pool in JSON format.

An example of the training set (JSON file) is shown below:

```
{
  "000001.txt": ["000005.txt", "012101.txt"],
  "003423.txt": ["398421.txt", "012101.txt",
  "173651.txt"],
  "012831.txt": ["000001.txt"],
  ...
}
```

The dataset includes:

- **Training set:** 7350 documents containing query cases with their corresponding noticed cases.
- **Test set:** 2159 documents consisting of only the query cases.

3 Related Work

Legal Case Retrieval (LCR) has been an active area of research since its inception in 1994. Existing methods can be broadly categorized into three areas: traditional statistical methods, neural language models, and hybrid approaches.

3.1 Traditional Approaches

Early LCR systems represent cases using hand-crafted features such as n-grams or learned embeddings like *doc2vec*. Retrieval of 'noticed cases' is performed using non-learning-based methods (e.g., TF-IDF, BM25) or supervised learning-based methods such as classification and ranking [8][6]. These approaches are computationally efficient and interpretable, but they often fail to capture deeper semantic structures inherent in legal language.

3.2 Neural Models

Neural methods have significantly advanced LCR by modeling the semantic richness of legal texts. Architectures like CNNs[15], BiDAF[11], and SMASH-RNN[3] encode case semantics more effectively. Transformer-based models like BERT-PLI[12] process documents in segments to perform pairwise comparisons. Models such as SAILER employ input trimming but risk loss of contextual information. Pre-trained legal-specific models such as *LEGAL-BERT* and *Lawformer* further improve representation by training on legal corpora, although they often lack retrieval-specific supervision.

3.3 Hybrid Models

Hybrid methods combine both lexical and semantic features to enhance retrieval. Graph-based models like *CaseGNN*[14] represent documents as text-as-graph constructs (TACGs), modeling sentences as nodes and their interdependencies as edges. These models use graph attention mechanisms to learn structural relationships that go beyond surface text and often outperform transformer-based models on COLIEE benchmarks.

Recent advancements include the use of Large Language Models (LLMs), such as GPT-4, for summarization and legal reasoning. At COLIEE 2024, systems like *CAPTAIN* utilized LLMs for legal entailment, while others incorporated Chain-of-Thought prompting to enhance interpretability and accuracy. These developments suggest that LLMs can provide significant improvements in guiding retrieval via legal reasoning [1, 7].

4 SIL Methodology

Despite advances, most models either focus on semantics or lexical overlap—not both. Our system bridges this by combining dense retrieval, structural features, and supervised

ranking. This offers a more holistic and scalable solution for retrieving relevant legal cases.

Our proposed system for Legal Case Retrieval (LCR) employs a two-stage pipeline designed to optimize both retrieval efficiency and relevance accuracy. The methodology integrates state-of-the-art techniques in semantic representation, approximate nearest document search, and learning-to-rank. The pipeline consists of two main stages (see Figure 1)

(1) Document Index Construction and Candidate Retrieval
(2) Learning-based Re-ranking of the relevant documents

4.1 Document Index Construction and Candidate Retrieval

The initial stage focuses on creating a searchable index of the legal case document corpus and obtaining a fast initial search to retrieve a set of potentially relevant candidate documents for a given query.

Preprocessing: The input legal case documents are preprocessed to perform text cleaning and text normalization. To ensure language consistency, we first detect and remove French text using langdetect. If a document contains mostly French, it is translated into English. For query cases, we extract only sentences containing placeholders like FRAGMENT_SUPPRESSED, REFERENCE_SUPPRESSED, and CITATION_SUPPRESSED, as they typically indicate cited references . For candidate cases, we retain the full text. If a case includes a summary, it is extracted and prepended to the main content. Cases without summaries are left unchanged. [5]

After preprocessing, we rank documents based on the scores of MPNet from HuggingFace, selecting the top 100 candidate documents for each query. MPNet model is a variant of BERT optimized for semantic similarity tasks, to transform each pre-processed document into a high-dimensional dense vector embedding. This process captures the underlying semantic meaning of the legal text, mapping documents with similar legal concepts to nearby points in the vector space. Unlike BERT's masked language modeling (MLM), which independently predicts masked tokens without modeling inter-token dependencies, or XLNet's permuted language modeling (PLM), which disrupts positional information, MPNet unifies masked and permuted pretraining (PMLM) while preserving original token positions [13]. This architecture is particularly suited to legal documents, where long-range dependencies (e.g., between clauses in a contract) and precise word order (e.g., in statutory definitions like "knowingly and willfully") are semantically critical. Empirical studies demonstrate MPNet's superiority in semantic similarity tasks [10], with a 4.2% higher accuracy than RoBERTa on the LegalBench benchmark [2]. To enable an efficient large-scale similarity search, the generated document embeddings are indexed using the Facebook AI Similarity Search (FAISS) library [4]. This index stores the vectors directly without compression and utilizes Maximum Inner Product Search (MIPS) for similarity computation. Given that sentence-transformer models like MPNet often produce normalized embeddings, maximizing the inner product is equivalent to maximizing cosine similarity, effectively identifying vectors pointing in similar directions within the semantic space. The result is a persistent index containing all document vectors. Then, obtain the vector representation of the query document with the same MPNet model that was used in the document index construction stage. This stage ensures that both the query and the candidate document vectors are represented in the same semantic vector space. The query vector is then used to search the FAISS index. This search efficiently computes the inner product similarity between the query vector and all indexed document vectors, enabling fast retrieval of potentially relevant documents. The system retrieves the identifiers of the top-N (where N = 100) documents corresponding to the vectors that yield the highest inner product scores with the query. These top-scoring vectors form a candidate set, which prioritizes recall and serves as input to the subsequent reranking stage. The value of N is a tunable hyperparameter that controls the size of this candidate pool.

4.2 Learning-based Re-ranking of the Relevant Documents:

The final stage refines the candidate set using a more sophisticated machine learning model to improve the precision and ranking order of the final results. Simple semantic similarity, while effective for initial retrieval, may not capture all dimensions of legal relevance. Therefore, for each query-candidate pair where the candidate belongs to the retrieved set, a comprehensive feature vector is extracted to capture deeper semantic, lexical, structural, and contextual relationships. Below are the nine features considered for rich feature vector [5]:

1. query_length: Number of tokens in the query case.
2. doc_length: Number of tokens in the candidate case.
3. query_ref_num: Count of references/citations in the query (e.g., <...>).
4. doc_ref_num: Count of references/citations in the candidate document.
5. BM25 Score: BM25 ranking score based on bag-of-words relevance.
6. Bm25 rank: Rank of candidate document based on BM25 scores
7. QLD Score: Query Likelihood with Dirichlet Smoothing – a probabilistic IR model score.
8. Qld rank: Rank of candidate document based on QLD score
9. Doc2Vec Similarity: Cosine similarity between Doc2Vec embeddings of the query and candidate case (dense vector representation capturing semantic similarity).

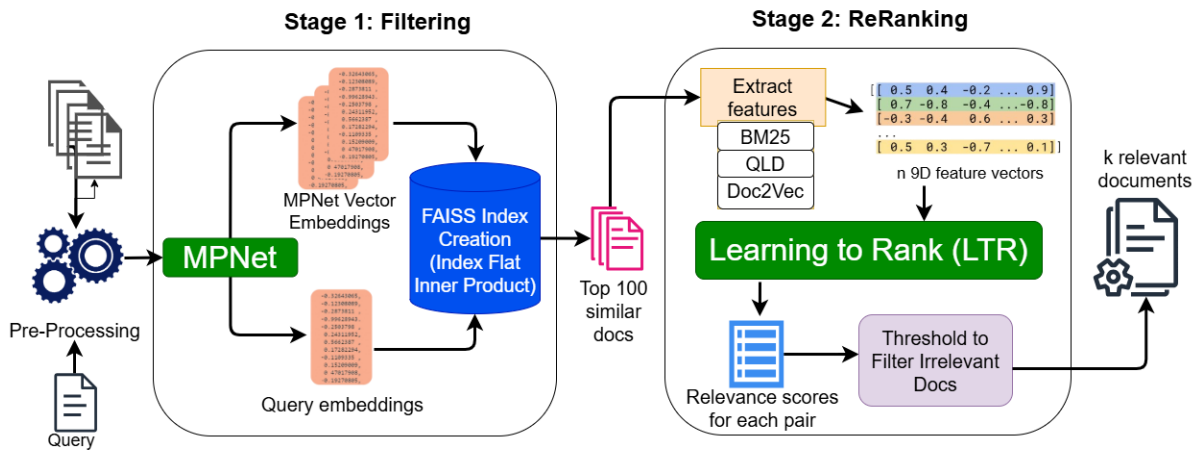


Figure 1. SIL Methodology for Legal Case Retrieval

The BM25 and QLD scores are calculated dynamically based on the top-k retrieved documents for each query. A Gradient Boosting Decision Tree (GBDT) model is employed for the re-ranking task, specifically LightGBM. This model is selected for its strong performance on tabular data, computational efficiency through techniques like gradient-based one-side sampling and exclusive feature bundling, and its native support for high-dimensional feature vectors. The pre-trained LightGBM model receives the constructed feature vector for each query-candidate pair and outputs a continuous relevance score, representing the predicted likelihood of the candidate document being relevant to the query. The LightGBM model is trained using a dataset of query-document pairs annotated with relevance labels (e.g., 'Relevant', 'Not Relevant'). Here, in our case, the relevant label means noticed cases. The model is trained specifically for a learning-to-rank task, using a listwise objective function. Normalized Discounted Cumulative Gain (NDCG) is used as the primary evaluation and optimization metric, as it accounts for both the relevance level of each document and its position in the ranked list. This ensures that the model learns to prioritize more relevant documents higher in the final output, thereby improving the quality of results presented to the query.

Thresholding: Once the relevance scores are generated for all candidate documents, a final filtering step is applied using a predefined relevance threshold. Documents with scores below this threshold are excluded as they are likely to be irrelevant. The threshold used is the similarity scores of the query-candidate greater than 0. It was selected as it was giving suitable number of candidate documents. Some other threshold can also be used. The relevant documents are sorted in descending order by relevance scores to form the final ranked list. This list represents the system's best estimate of the most relevant legal cases for the given query.

5 Results

Evaluation Metrics

The system's performance is evaluated using precision, recall, and F-measure. These are computed using micro-averaging across all queries.

- **Precision:** $\frac{\text{Number of correctly retrieved cases}}{\text{Total number of retrieved cases}}$
- **Recall:** $\frac{\text{Number of correctly retrieved cases}}{\text{Total number of relevant cases}}$
- **F-measure:** $2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$

Micro-averaging is used, meaning that the evaluation metrics are computed globally across all queries, rather than individually (macro-average).

Table 1. Evaluation Metrics for SIL System Methodology

System	Precision	Recall	F1 Score
SIL	0.0054	0.0063	0.0058

The task is very challenging, and while the SIL team's performance is not strong (about 8th in 12 submissions), it still shows promise for two reasons

- It is argued that managing computational resources for such a challenging problem is important as improvements based only on more computation don't provide insight into semantic structure of the case retrieval challenge.
- The idea of a "cascaded" structure of incremental heuristic methods to successively filter case candidates offers the opportunity to identify what kinds of lexical and semantic heuristics offer potential high advantage in this hybrid approach.

6 Conclusion

The COLIEE 2025 competition has provided a wonderful opportunity for our SIL team to experiment with different techniques to address legal case retrieval. We plan to use more features in stage 2 in our future work. We also look forward to developing a dataset for the Indian jurisdiction and bringing it to the COLIEE challenge. We are committed to employing the lessons learned throughout this competition.

Acknowledgments

We acknowledge the support of the IHUB-ANUBHUTI-IIITD FOUNDATION set up under the NM-ICPS scheme of the Department of Science and Technology, India

References

[1] Masaki Fujita, Takaaki Onaga, and Yoshinobu Kano. 2024. LLM Tuning and Interpretable CoT: KIS Team in COLIEE 2024. In *New Frontiers in Artificial Intelligence*, Toyotaro Suzumura and Mayumi Bono (Eds.). Springer Nature Singapore, Singapore, 140–155.

[2] Neel Guha, Julian Nyarko, Daniel E. Ho, Christopher Ré, Adam Chilton, Aditya Narayana, Alex Chohlas-Wood, Austin Peters, Brandon Waldon, Daniel N. Rockmore, Diego Zambrano, Dmitry Talisman, Enam Hoque, Faiz Surani, Frank Fagan, Galit Sarfaty, Gregory M. Dickinson, Haggai Porat, Jason Hegland, Jessica Wu, Joe Nudell, Joel Niklaus, John Nay, Jonathan H. Choi, Kevin Tobia, Margaret Hagan, Megan Ma, Michael Livermore, Nikon Rasumov-Rahe, Nils Holzenberger, Noam Kolt, Peter Henderson, Sean Rehaag, Sharad Goel, Shang Gao, Spencer Williams, Sunny Gandhi, Tom Zur, Varun Iyer, and Zehua Li. 2023. LegalBench: A Collaboratively Built Benchmark for Measuring Legal Reasoning in Large Language Models. arXiv:2308.11462 [cs.CL] https://arxiv.org/abs/2308.11462

[3] Jyun-Yu Jiang, Mingyang Zhang, Cheng Li, Michael Bendersky, Nadav Golbandi, and Marc Najork. 2019. Semantic Text Matching for Long-Form Documents. In *The World Wide Web Conference* (San Francisco, CA, USA) *(WWW '19)*. Association for Computing Machinery, New York, NY, USA, 795–806. https://doi.org/10.1145/3308558.3313707

[4] Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2017. Billion-scale similarity search with GPUs. arXiv:1702.08734 [cs.CV] https://arxiv.org/abs/1702.08734

[5] 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

[6] Yixiao Ma, Yunqiu Shao, Yueyue Wu, Yiqun Liu, Ruizhe Zhang, Min Zhang, and Shaoping Ma. 2021. LeCaRD: A Legal Case Retrieval Dataset for Chinese Law System. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Virtual Event, Canada) *(SIGIR '21)*. Association for Computing Machinery, New York, NY, USA, 2342–2348. https://doi.org/10.1145/3404835.3463250

[7] 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*, Toyotaro Suzumura and Mayumi Bono (Eds.). Springer Nature Singapore, Singapore, 125–139.

[8] Jay M. Ponte and W. Bruce Croft. 1998. A language modeling approach to information retrieval. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (Melbourne, Australia) *(SIGIR '98)*. Association for Computing Machinery, New York, NY, USA, 275–281. https://doi.org/10.1145/290941.291008

[9] 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 (April 2022), 111–133. https://doi.org/10.1007/s12626-022-00105-z

[10] Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. arXiv:1908.10084 [cs.CL] https://arxiv.org/abs/1908.10084

[11] Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. 2018. Bidirectional Attention Flow for Machine Comprehension. arXiv:1611.01603 [cs.CL] https://arxiv.org/abs/1611.01603

[12] Yunqiu Shao, Jiaxin Mao, Yiqun Liu, Weizhi Ma, Ken Satoh, Min Zhang, and Shaoping Ma. 2021. BERT-PLI: modeling paragraph-level interactions for legal case retrieval. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence* (Yokohama, Yokohama, Japan) *(IJCAI'20)*. Article 484, 7 pages.

[13] Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2020. MPNet: Masked and Permuted Pre-training for Language Understanding. arXiv:2004.09297 [cs.CL] https://arxiv.org/abs/2004.09297

[14] Yanran Tang, Ruihong Qiu, Yilun Liu, Xue Li, and Zi Huang. 2023. CaseGNN: Graph Neural Networks for Legal Case Retrieval with Text-Attributed Graphs. arXiv:2312.11229 [cs.IR] https://arxiv.org/abs/2312.11229

[15] 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 (ICAIL '19)*. ACM. https://doi.org/10.1145/3322640.3326740

A ModernBERT-Based System by Team KIS for the COLIEE 2025 Pilot Task: Toward Robust Evaluation in Legal Judgment Prediction

Kazuma Kadowaki
Shizuoka University
Hamamatsu, Shizuoka, Japan
kkadowaki@kanolab.net

Yoshinobu Kano
Shizuoka University
Hamamatsu, Shizuoka, Japan
kano@inf.shizuoka.ac.jp

ABSTRACT

Legal Judgment Prediction (LJP) has emerged as a promising task in the legal domain, aiming to support decision-making processes by predicting court outcomes. The COLIEE 2025 shared task introduced a new pilot subtask, LJPJT 2025, focusing on Japanese civil tort cases and comprising two subtasks: tort prediction (TP) and rationale extraction (RE). In this paper, we present the system developed by Team KIS for LJPJT 2025. Our system employs a simple yet effective architecture based on ModernBERT, and achieves competitive results, including the top F1 score on the RE task among all participants.

Beyond system implementation, we conduct an in-depth analysis of evaluation metrics using over 200 models trained with diverse hyperparameter configurations and data splits. Our findings reveal substantial variation in model performance across data splits and metrics, highlighting the difficulty of evaluating model performance with respect to generalization under such variability. We also demonstrate that binary F1 scores, officially used in RE evaluation, are highly sensitive to subjective design choices, such as label definitions, making them potentially unsuitable for consistent model evaluation.

Our study underscores the importance of metric design in legal NLP tasks and offers insights for future research on robust evaluation methods.

CCS CONCEPTS

• **Computing methodologies** → **Natural language processing**; Cross-validation; • **Applied computing** → *Law*.

KEYWORDS

COLIEE, Legal Judgment Prediction, ModernBERT, Model Ensemble, Evaluation Metrics

ACM Reference Format:
Kazuma Kadowaki and Yoshinobu Kano. 2025. A ModernBERT-Based System by Team KIS for the COLIEE 2025 Pilot Task: Toward Robust Evaluation in Legal Judgment Prediction. In *Proceedings of COLIEE 2025 workshop, June 20, 2025, Chicago, USA.* ACM, New York, NY, USA, 9 pages.

COLIEE 2025, June 20, 2025, Chicago, USA

1 INTRODUCTION

AI-assisted workflows have enabled the processing of larger volumes of documents than ever before. As natural language processing (NLP) technology continues to advance, its applications have expanded across a wide range of domains. The legal field is no exception. For over a decade, the Competition on Legal Information Extraction and Entailment (COLIEE) [21] has been held to promote the automatic processing of legal documents, attracting many researchers worldwide.

Previous editions of COLIEE have focused primarily on legal information retrieval and entailment tasks. However, COLIEE 2025 [10] introduced a new pilot task: the Legal Judgment Prediction for Japanese Tort cases (LJPJT) 2025 task [32].

The Legal Judgment Prediction task targets civil litigation cases involving torts. It comprises two subtasks: rationale extraction (RE), which predicts whether each claim by the plaintiff and defendant is accepted by the court; and tort prediction (TP), which predicts the court's final decision. While this task is technically related to judicial automation, it also supports real-world use cases such as enabling litigants to select favorable claims, thereby facilitating faster settlements and lowering the cost of legal services.

The dataset constructed by Yamada et al. [32] represents a significant addition to the field of Legal Judgment Prediction, following previous efforts in China [31] and Europe [4, 5, 16]. It is the first dataset of its kind designed specifically for the Japanese legal system, where legal conventions and civil procedures differ substantially from those in other jurisdictions.

We approached the task by developing a simple system based on ModernBERT [29]. Our system uses the 130M parameter variant of ModernBERT, allowing for efficient inference and training within limited computational budgets. Although the architecture itself is straightforward, we achieved slightly improved performance through model ensembling.

This paper provides an overview of our system and compares multiple models built for the task. Based on these comparisons, we discuss the inherent difficulty of the task and examine which evaluation metrics may better capture model performance in a robust manner.

Our contributions are summarized as follows:

- We developed a ModernBERT-based system for the LJPJT 2025 task. Our system achieved the best performance among participants on the rationale extraction (RE) task.
- Using a large collection of trained models, we investigated the effects of different evaluation metrics and data splits, and

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.	✓
PC_2	There are only two persons with the surname “X1” who work at factory B: the plaintiff and his cousin.	✓
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.	✗
PC_4	It is possible to identify the subject of this posting as the plaintiff.	✗
Defendant's Claims (DCs)		
DC_1	We do not admit all the allegations from plaintiff.	✓
Tort Prediction		
Court Decision		✗

Figure 1: An example tort instance, with texts cited from [32].

discussed the challenges of the task and the design of better evaluation criteria.

- We demonstrate that binary F1 scores, officially used in RE evaluation, are highly sensitive to subjective label choices and split conditions, making them less suitable as robust evaluation metrics.

2 RELATED WORK

2.1 The LJPJT 2025 Task

The dataset for the LJPJT 2025 task is constructed from Japanese court judgments. Each judgment may involve one or more instances of torts, specifically those defined under Article 709 of the Japanese Civil Code (unlawful acts). For each tort, the dataset includes three categories of text segments extracted from the court judgment documents: undisputed facts, claims from the plaintiff, and claims from the defendant.

This task consists of two subtasks: tort prediction (TP) and rationale extraction (RE). The RE subtask involves predicting whether each individual claim, either from the plaintiff or the defendant, was accepted by the court. The TP subtask requires predicting the court’s final decision regarding each tort case.

Figure 1 shows an example tort instance used in the task, and Figure 2 shows its corresponding JSON-style input. Our system aims to predict the `court_decision` for each tort, along with the `is_accepted` status for each claim. Further details on the textual structure of the input can be found in [32]. Unlike other COLIEE datasets, this dataset is provided solely in Japanese (monolingual).

```
{
    "version": "train001.jsonl",
    "tort_id": "0",
    "undisputed_facts": [
        {"id": "0", "description": "A posting ..."}
    ],
    "plaintiff_claims": [
        {"id": "0", "description": "This posting ...",
         "is_accepted": true},
        {"id": "1", "description": "There are ...",
         "is_accepted": true},
        {"id": "2", "description": "The viewers ...",
         "is_accepted": false},
        {"id": "3", "description": "It is ...",
         "is_accepted": false}
    ],
    "defendant_claims": [
        {"id": "0", "description": "We do not ...",
         "is_accepted": true}
    ],
    "court_decision": false
}
```

Figure 2: An example JSON-style input for the LJPJT 2025 task.

The official baseline system for the LJPJT 2025 task is the Inter-Span Transformer (IST), an improved variant of the model proposed by Chalkidis et al. [5]. It has been evaluated by the task organizers as a strong benchmark [32]. The IST model encodes each claim using BERT [7] and applies a shared Transformer [26] model to jointly predict both rationale extraction (RE) and tort prediction (TP). During training, hyperparameter optimization was performed not only for standard parameters such as learning rate and model size, but also for the loss weighting ratio between the two subtasks. Specifically, the overall loss is defined as $\alpha \cdot \text{loss}_{\text{TP}} + (1-\alpha) \cdot \text{loss}_{\text{RE}}$, where α is a tunable hyperparameter.

A notable characteristic of this task is its difficulty, as even human annotators often fail to make correct predictions. In our preliminary experiments, modifications to the model architecture yielded only marginal improvements. Consequently, detailed architecture exploration is considered out of scope for this paper.

For reproducibility, the task organizers allow the use of publicly available LLMs, but prohibit the use of closed models such as GPT-4o [19] or Gemini [24].

2.2 Ensemble Approaches in Other Tasks

Model ensembling has become a common and effective approach in legal-domain tasks, particularly in recent COLIEE competitions. Many participating teams have employed ensemble methods to boost performance [1, 9, 13, 17, 28]. In line with this trend, our system also incorporates ensembling to enhance prediction accuracy.

2.3 ModernBERT

BERT [7] has been widely used across various natural language processing (NLP) tasks, including classification, information retrieval, ranking, and named entity recognition. A common paradigm is to pre-train the model on a large general-purpose corpus, such as Wikipedia or web pages, and then fine-tune it on a smaller dataset

specific to the downstream task. Compared to large language models (LLMs), BERT-based models are lightweight and easier to train, making them still attractive for domain-specific tasks where general LLMs may lack sufficient coverage. However, BERT has several limitations, including a relatively short maximum sequence length of 512 tokens and architectural inefficiencies that arise from hardware-agnostic design choices.

To address these issues, Warner et al. [29] recently proposed ModernBERT, which introduces several architectural and training improvements. Most notably, it supports sequence lengths of up to 8,192 tokens and demonstrates improved performance across various tasks, partly due to pretraining on a large-scale corpus that includes a diverse range of sources, such as source code. To handle long sequences efficiently, ModernBERT adopts a hybrid attention mechanism: while some Transformer layers [26] retain full attention across all tokens, others (specifically two-thirds) use local attention restricted to neighboring tokens. Additional improvements include the removal of next-sentence prediction (NSP), a pretraining task originally used in BERT to model sentence-level coherence, based on findings that it contributes little to downstream performance; the replacement of absolute positional embeddings with rotary embeddings; and various architectural refinements such as modifications to bias terms and LayerNorm. Training efficiency is further enhanced through various improvements, including changes to the optimizer, learning rate scheduling, and unpadding strategies.

While Warner et al.'s original ModernBERT targets English, a Japanese counterpart `sbintuitions/modernbert-ja-130m`[1] has been released as a publicly available pretrained model. It inherits many of the same design principles, including support for long sequences and architectural efficiency. One key difference, however, lies in its tokenizer: whereas the original ModernBERT uses the `[CLS]` … `[SEP]` format similar to BERT, the Japanese version adopts a RoBERTa-style format using `<s>` … `</s>` [14]. Although the vocabulary includes special tokens such as `<cls>` and `<sep>`, these tokens were not used in the pretraining corpus and thus lack specific learned behavior. The `modernbert-ja` models are also available in multiple size variants: tiny (30M), small (70M), base (130M), and large (315M). Among these, we use the base-size (130M) model, which consists of 19 layers, with a hidden size of 512 and an intermediate dimension of 2,048.

2.4 Evaluation Metrics and their Robustness

To develop more robust NLP models, many prior studies have explored strategies that improve the reliability of model evaluation under varying conditions. These include addressing biases in label definitions [8, 11, 23] and enhancing interpretability and explainability [3, 12].

From the perspective of evaluation methodology, Moss et al. [15] proposed J-K-fold cross-validation as a more robust alternative to standard train-test splits. Moreover, while metrics such as accuracy and F1-score are commonly used, they often fail to capture issues such as model bias or generalizability. As a result, there has been growing interest in alternative metrics [2, 20, 22, 27].

[1] https://huggingface.co/sbintuitions/modernbert-ja-130m

In particular, Vickers et al. [27] conducted a large-scale empirical comparison of several evaluation metrics, including Accuracy, Balanced Accuracy, F1 (especially Macro-F1), Informedness, Matthews Correlation Coefficient (MCC), and Normalized Information Transfer (NIT), across tasks such as natural language understanding (NLU), visual question answering (VQA), and machine translation (MT). They concluded that Informedness is not only intuitive, as it interprets evaluation as an "odds game" in which chance-level performance receives no credit, but also more effective at capturing model generalizability. Based on these findings, they recommend reporting Informedness alongside standard metrics in future research. We briefly summarize the definitions of Informedness, MCC, and NIT below.

Informedness. Informedness measures the probability that a prediction is informed, rather than due to class bias or random guessing. It is defined as:

$$I = \sum_{t=1}^{N} \frac{\Pr(\hat{Y} = y_t)}{N} \cdot \begin{cases} \frac{1}{\Pr(Y=y_t)} & \text{if } \hat{y}_t = y_t \\ -\frac{1}{1-\Pr(Y=y_t)} & \text{if } \hat{y}_t \neq y_t \end{cases}$$

where N is the number of samples, and $\hat{y}_t$ and y_t denote the predicted and true labels for the t-th sample, respectively. Here, $\Pr(Y = y)$ represents the empirical distribution of true labels, and $\Pr(\hat{Y} = y)$ is the empirical distribution of predicted labels. This formulation rewards correct predictions more when the true class is rare (i.e., when $\Pr(Y = y)$ is small), and penalizes incorrect predictions more when the true class is common (i.e., when $\Pr(Y = y)$ is large).

Matthews Correlation Coefficient (MCC). MCC [6] is defined as the correlation between predicted and true labels:

$$\mathrm{MCC} = \frac{\mathrm{Cov}(\hat{y}, y)}{\sigma_{\hat{y}} \cdot \sigma_y},$$

where $\mathrm{Cov}(\hat{y}, y)$ is the covariance between the predicted labels $\hat{y}$ and the true labels y, and $\sigma_{\hat{y}}, \sigma_y$ are the corresponding standard deviations. It takes the value 1 for perfect predictions and 0 when predictions are uncorrelated with the true labels.

Normalized Information Transfer (NIT). NIT [25] is an information-theoretic measure that quantifies how much uncertainty is reduced by a classifier compared to a uniform random guess. It is defined as:

$$\mathrm{NIT} = \frac{2^{I(\hat{Y};Y)}}{K},$$

where $I(\hat{Y}; Y)$ is the mutual information between the predicted labels $\hat{Y}$ and the true labels Y, and K is the number of classes. A value close to $\frac{1}{K}$ indicates random-level performance, while higher values reflect more informative predictions.

3 IMPLEMENTATION AND EXPERIMENT

Our system is based on ModernBERT [29]. In this section, we describe the implementation details.

ModernBERT offers two key advantages: strong baseline performance and support for long input sequences. Traditional BERT models are limited in input length, requiring segmentation of texts into small units, such as individual facts or claims. Ideally, however, joint

modeling of all information is desirable, as demonstrated by the effectiveness of multitask learning in the IST baseline for TP and RE subtasks. Accordingly, we leverage ModernBERT's long-context capability and design our system to process input sequences as unified documents. We fine-tuned the publicly available `sbintuitions/modernbert-ja-130m` model for our task.

3.1 Model Input and Output for Fine-Tuning

The input to our model is a concatenated string comprising undisputed facts and claims, formatted as follows:

```
<cls> <s> <sep> UF_1 <sep> UF_2 ... </s>
      <s> <sep> PC_1 <sep> PC_2 ... </s>
      <s> <sep> DC_1 <sep> DC_2 ... </s>
```

Here, UF_i, PC_i, and DC_i denote individual undisputed facts, plaintiff claims, and defendant claims, respectively.

The tokens `<cls>` and `<sep>` are special tokens defined in the ModernBERT vocabulary, although they are not used in the pretraining of `sbintuitions/modernbert-ja-130m` but included in our fine-tuning; `<s>` and `</s>` are special tokens used to separate sentences, included in the pretraining. We deliberately employ `<s>` and `</s>` to separate the three input segments, and use `<sep>` to mark individual claims, since `<s>` and `</s>` alone do not indicate claim boundaries.

During our fine-tuning, each `<sep>` token is associated with a binary label for rationale extraction (RE). In particular, all `<sep>` tokens within the UF segment are labeled as true (accepted), and the model is trained accordingly. This labeling reflects the fact that undisputed facts are always accepted by both parties and are generally considered valid reasoning components in legal documents[2]. However, during inference, outputs from the UF segment are discarded.

ModernBERT supports sequences of up to 8,192 tokens. For practical reasons, including memory constraints, we set an upper limit of 6,144 tokens. If the concatenated input exceeds this threshold, we first remove the undisputed facts. If the input remains too long, we truncate tokens from the end of the remaining sequence (i.e., the DC segment first, followed by the PC segment if needed)[3]. During truncation, we preserve all `<sep>` tokens, even if the corresponding text has been removed, because each `<sep>` corresponds to a prediction target; Removing them would cause inconsistencies between the input and the model's output structure.

For our model output, the `<cls>` token is used for tort prediction (TP), while each `<sep>` token performs binary classification for rationale extraction (RE). As we fine-tuned the model with a standard cross-entropy classification loss, it naturally produces probability scores, which we retain during inference to support ensembling.

During fine-tuning, we compute a combined loss function from both TP and RE outputs, following the implementation of the IST baseline. Specifically, the overall loss is given by

$$\alpha \cdot \mathrm{loss}_{\mathrm{TP}} + (1 - \alpha) \cdot \mathrm{loss}_{\mathrm{RE}},$$

where α is a tunable hyperparameter.

[2]In our preliminary experiments, we did not observe any significant difference in performance when excluding the loss contributions from the UF segments.

[3]Among the 6,508 instances in the dataset, the undisputed facts (UFs) were removed in only 14 instances. Of these, 4 and 9 instances also involved truncation of at least one plaintiff claim (PC) and defendant claim (DC), respectively.

3.2 Hyperparameter Selection

We conducted grid search over several hyperparameters to train multiple ModernBERT models. The parameters explored were:

- **Epochs:** 3, 5, 10, 20
- **Learning Rate:** 5e-6, 1e-5, 2e-5, 3e-5, 5e-5, 1e-4
- $\boldsymbol{\alpha}$ (loss weighting for TP task): 0.1, 0.2, …, 0.9

The selection of epochs and learning rates was informed by prior experiments on the base `sbintuitions/modernbert-ja-130m` model. However, we added 20 epochs based on preliminary findings suggesting better performance with longer training.

The following hyperparameters were fixed throughout all experiments:

- Batch size: 32
- Warmup ratio: 0.01
- Early stopping: Disabled

All other parameters were left at their default values in the Huggingface Transformers framework [30].

3.3 Ensembling Based on Hyperparameters

As a result of hyperparameter tuning, multiple trained models were obtained. Among them, we selected a single model that performs the best in the validation set to serve as one of our final systems.

In addition, we constructed an ensemble system using the top five models according to validation performance. For each label (TP and RE), we aggregated the predicted probabilities from these models to obtain a final probability. If the aggregated probability exceeded 0.5, the label was predicted as true. This ensembling strategy effectively increases the parameter capacity of the system and is expected to improve performance.

3.4 Data Splits for Development

The official LJPJT 2025 task provides only training and test datasets. To facilitate development, we split the training data (`train001.jsonl`) into three subsets with an 8:1:1 ratio, resulting in 5,206 samples for training, 651 for validation, and 651 for development.

While such splits are typically randomized, we performed the split sequentially, without shuffling the dataset. This decision was motivated by the observation that a single court judgment may contain multiple tort cases that share the same undisputed facts but differ in claims. Shuffling the data may cause related examples to appear in both training and validation/development sets, potentially leading to data leakage[4].

3.5 Ensembling Based on Data Splits

Using a fixed split of the training data may result in suboptimal use of available data, especially since the development and validation subsets are excluded from final training. To address this, we also explored an alternative ensembling strategy based on multiple data splits.

Specifically, we partitioned the training dataset into five folds and trained five models, each using four folds for training and the

[4]In our preliminary experiments, random shuffling indeed resulted in higher performance compared to sequential splitting, supporting our concern about possible leakage.

remaining fold for validation and development. To keep the computational workload manageable, this experiment was conducted with α fixed at 0.5.

Using a similar ensembling approach, we averaged the predicted probabilities across the five models and output true when the average exceeded 0.5. This ensemble was used to examine whether utilizing the entire training data, including portions that were used for validation and development, could lead to improved performance.

4 RESULTS

In this section, we describe the systems and results submitted by our team, KIS, to the LJPJT 2025 pilot task of COLIEE 2025. We present evaluation results on our development set and report our official formal run scores.

4.1 Our Formal Run Submissions

We submitted three systems to the LJPJT 2025 task[5]:

- **KIS4**: the best-performing individual model.
- **KIS5**: an ensemble of the top five models trained with different hyperparameters, as described in Section 3.3.
- **KIS6**: an ensemble of five models trained on different data splits using the same hyperparameters, as described in Section 3.5.

4.2 Results on Development Set

Table 1 shows evaluation results of our models on our development set, which was created by splitting the provided training data. Here, **F1 (True)** refers to the standard binary F1 score where the positive class is "true" (i.e., `court_decision` for TP and `is_accepted` for RE), and **F1 (Macro)** is the unweighted mean of F1 scores computed separately for the "true" and "false" classes.

Note that KIS6 itself is not evaluated on the development set, since four of its five component models were trained using this set. Comparisons across different data splits may not be directly meaningful, as the models were evaluated on different test data.

Our results show that ensemble learning, as in previous COLIEE tasks, offers modest improvements for the target metrics: Accuracy for TP and F1 (True) for the All category in RE. However, the best-performing model may vary depending on the evaluation metric, and no single model consistently outperforms the others across all criteria.

4.3 Results on Formal Run Dataset

Table 2 shows the formal run results for the TP and RE tasks in LJPJT 2025[6]. Among our systems, KIS5 and KIS6 performed better in the TP task, and KIS5 also achieved the highest F1 (True) score for the All category in the RE task.

Interestingly, the trends observed in the formal runs do not always match those seen on the development set. For example, in the RE task, the F1 (True) scores were relatively similar across all categories (All, Plaintiff, Defendant) in the development results, but the formal run showed a notably lower score for Plaintiff. Similarly, the KIS4 model exhibited significantly lower F1 (Macro) on the development set[7], but this was not the case in the formal evaluation. In fact, KIS4 achieved the highest Plaintiff F1 (Macro) among all models, despite having the lowest Plaintiff Accuracy. These results suggest that the current evaluation metrics do not always clearly capture the strengths and weaknesses of each system.

Moreover, the limited improvement from using all available training data in the KIS6 ensemble suggests that simply increasing training size does not necessarily lead to better performance in this task. While the overall system performance remains insufficient for real-world deployment, our findings indicate that current performance limitations may stem more from task complexity than from data volume. Future improvements may benefit from revisiting aspects of data design or incorporating additional knowledge sources, while continuing to build on the strengths of the existing dataset.

5 RECONSIDERING EVALUATION METRICS

In this section, we reconsider the evaluation metrics used in the LJPJT 2025 task based on the models we developed. The official metrics were accuracy for the TP task and F1 score (with "true" as the positive class) for the RE task. Additionally, Yamada et al. [32] used accuracy for both tasks. Our research question is whether these metrics are indeed the most appropriate ones, and if not, what alternatives may serve as more robust evaluation measures[8].

The following discussion is based on the results of the models we trained during hyperparameter search. All of these models share the same architecture and differ only in hyperparameters (including four epoch settings, six learning rates, and nine values of α) or in the data used for training and evaluation (five data splits). Other potential sources of model variability, such as random seeds for initialization, were fixed in our experiments. Furthermore, a more comprehensive analysis would include comparisons across different model architectures. This work should therefore be understood as a first step toward better metric design, and we leave the investigation of these additional factors for future work.

5.1 Requirements for Robust Evaluation Metrics

Robust evaluation metrics should exhibit the following properties:

- **Interpretability and task alignment**: The score should reflect actual use cases or be intuitively interpretable. In LJPJT, for example, the RE and TP tasks are logically connected, as final decisions are based on whether each claim is accepted. Ideally, such inter-task dependencies should be captured in the evaluation as well.
- **Robustness to data splits**: A model that scores well on one test set should also generalize to other unseen data. Evaluation metrics should be stable regardless of how the data is split.

[5]The names KIS1, KIS2, and KIS3 were reserved for systems submitted to COLIEE 2025 Task 4. Details of those systems are available in a separate paper by the KIS team [18]
[6]Only TP Accuracy and RE F1 (True) for the All category are officially considered as ranking metrics [10]. While the table includes both TP and RE results, note that some participating systems may use separate models for each task internally.

[7]We did not monitor this metric during model selection.
[8]Note that robustness is a necessary condition for adoption in shared tasks, but not a sufficient one.

Table 1: Evaluation results of our models on the development set. KIS5 is an ensemble of KIS4 and Models 2–5 for KIS5. KIS6 is omitted because its training data overlaps with the development set.

		TP	RE								
		Accuracy	Accuracy			F1 (True)			F1 (Macro)		
Model	Split		All	Pltf.	Deft.	All	Pltf.	Deft.	All	Pltf.	Deft.
KIS5 (Ensembled)	devel	0.6482	0.6217	**0.6515**	0.5916	**0.7100**	**0.7346**	**0.6849**	0.5830	0.6136	0.5523
KIS4	devel	0.6528	0.5311	0.5566	0.5053	0.6937	0.7151	0.6714	0.3469	0.3576	0.3357
Model 2 for KIS5	devel	0.6298	0.6205	0.6448	0.5959	0.6509	0.6810	0.6188	0.6176	0.6401	0.5945
Model 3 for KIS5	devel	0.6436	0.6181	0.6256	0.6105	0.6545	0.6621	0.6467	0.6138	0.6212	0.6063
Model 4 for KIS5	devel	0.6160	0.6089	0.6261	0.5916	0.6372	0.6579	0.6156	0.6065	0.6228	0.5900
Model 5 for KIS5	devel	0.6513	0.6342	0.6433	0.6250	0.6491	0.6690	0.6275	0.6336	**0.6412**	0.6250
KIS6 (Ensembled)	-										
Model 1 for KIS6	devel	0.6329	**0.6371**	0.6333	**0.6410**	0.6525	0.6556	0.6493	**0.6364**	0.6317	**0.6408**
Model 2 for KIS6	devel-2	0.6743	0.5983	0.6188	0.5752	0.6149	0.6212	0.6084	0.5976	0.6188	0.5721
Model 3 for KIS6	devel-3	0.6662	0.6053	0.6042	0.6064	0.6189	0.6092	0.6287	0.6048	0.6042	0.6050
Model 4 for KIS6	devel-4	**0.6759**	0.6187	0.6264	0.6104	0.6398	0.6321	0.6475	0.6174	0.6263	0.6060
Model 5 for KIS6	devel-5	0.6528	0.6124	0.6083	0.6170	0.6282	0.6417	0.6113	0.6117	0.6048	0.6169

Table 2: Formal run results for LJPJT 2025, a pilot task of COLIEE 2025.

			TP	RE								
Rank			Accuracy	Accuracy			F1 (True)			F1 (Macro)		
(RE)	Team	Model		All	Pltf.	Deft.	All	Pltf.	Deft.	All	Pltf.	Deft.
1	**KIS**	**KIS5**	0.7131	0.6414	0.6452	0.6379	**0.7124**	**0.6734**	0.7402	**0.6560**	0.3862	0.4817
2	CAPTAIN	JAIST-LJPJT25	**0.7648**	**0.6865**	0.6455	**0.7238**	0.7055	0.6631	**0.7434**	0.6187	0.3616	0.3640
3	NOWJ	system2	0.6712	0.6691	0.6431	0.6930	0.6921	0.6401	0.7331	0.6073	0.3060	0.4207
4	omega	modernbert	0.6663	0.6780	**0.6998**	0.6582	0.6915	0.6708	0.7063	0.5937	0.3099	0.4147
5	**KIS**	**KIS4**	0.6970	0.5171	0.4609	0.5684	0.6816	0.6310	0.7247	0.6392	**0.4635**	**0.5236**
6	NOWJ	system1	0.6379	0.6555	0.6279	0.6807	0.6812	0.6263	0.7243	0.6015	0.3131	0.4276
7	**KIS**	**KIS6**	0.7131	0.6696	0.6642	0.6746	0.6730	0.6054	0.7185	0.5808	0.2912	0.3947
8	OVGU	OVGU1	0.5148	0.5225	0.4806	0.5607	0.6568	0.6101	0.6962	0.6045	0.4180	0.4892
9	NOWJ	system3	0.5973	0.5408	0.5317	0.5491	0.5587	0.6368	0.4456	0.5257	0.4141	0.1712
10	OVGU	OVGU2	0.5530	0.5273	0.5240	0.5304	0.4863	0.4497	0.5161	0.3879	0.2264	0.2811
11	OVGU	OVGU3	0.5320	0.5146	0.5264	0.5037	0.3164	0.2904	0.3376	0.2044	0.1087	0.1537

- **Handling Ambiguous or Noisy Samples**: Metrics should account for samples that are ambiguous or noisy and thus difficult to predict, even for humans. This can be done, for example, by assigning different weights based on inter-annotator agreement, or by reporting performance separately for such samples.
- **Low sensitivity to label definitions**: Metrics such as recall or precision can change drastically depending on which class is considered "true". Robust metrics should either avoid such subjective choices or remain stable under alternative formulations.
- **Scalability and feasibility**: The computational or manual cost of calculating the metric should remain acceptable as the dataset grows.

In the following subsections, we empirically investigate (1) robustness to data splits, (2) interdependence between the TP and RE tasks, and (3) correlations among metrics, based on our collection of trained models. Aspects such as ambiguity or noise, which cannot be inferred from the current dataset, are outside the scope of this paper.

5.2 Generalization: Metric Stability Across Data Splits

We assessed how consistent each evaluation metric is when the model is trained and tested on different data splits using the same architecture and hyperparameter settings. Specifically, we applied 24 hyperparameter configurations (4 epochs × 6 learning rates), training and evaluating a separate model on each of the five data splits. Table 3 reports the pairwise correlation coefficients of scores obtained across these splits for TP accuracy and RE F1 (True).

For the TP task, accuracy shows high correlation among splits 2, 3, and 4, but split 0 displays much lower correlation with the others. This indicates that a model performing well on one split does not necessarily generalize well to others.

Table 3: Correlation coefficients of scores across data splits.

a. TP task: Accuracy

split	0	1	2	3	4
0	1.0000	0.3871	0.1318	0.2680	0.3907
1	0.3871	1.0000	0.7765	0.8229	0.7228
2	0.1318	0.7765	1.0000	0.9346	0.8864
3	0.2680	0.8229	0.9346	1.0000	0.9727
4	0.3907	0.7228	0.8864	0.9727	1.0000

b. RE task: F1 (True)

split	0	1	2	3	4
0	1.0000	0.7960	0.5545	0.7887	0.8445
1	0.7960	1.0000	0.9013	0.8151	0.8580
2	0.5545	0.9013	1.0000	0.7675	0.6911
3	0.7887	0.8151	0.7675	1.0000	0.9184
4	0.8445	0.8580	0.6911	0.9184	1.0000

Table 4: Mean correlation coefficients across data splits (all metrics).

Task	Metirc		Correlation
TP	Accuracy		0.7035
	F1 (True)		0.5986
	F1 (False)		0.5958
	F1 (Macro)		0.7054
	F1 (Weighted)		0.7195
	Informedness		0.7009
	NIT		0.6872
	MCC		0.6999
RE	Accuracy	All	0.8874
	F1 (True)	All	0.8348
	F1 (False)	All	0.8629
	F1 (Macro)	All	0.8895
	F1 (Weighted)	All	0.8891
	Informedness	All	0.8875
	NIT	All	0.8880
	MCC	All	0.8524
	Accuracy	Doc-Level	0.9136
	F1 (Macro)	Doc-Level	0.9193
	F1 (Weighted)	Doc-Level	0.9242

For the RE task, F1 (True) on split 2 exhibits weak correlation with other splits, indicating that a data split suitable for evaluating TP models may not be equally suitable for RE.

We also computed average correlation coefficients for other metrics, shown in Table 4. For definitions of Informedness, NIT, and MCC, see Section 2.4 or Vickers et al. [27].

Overall, the TP task tends to show lower cross-split correlation regardless of the metric used, making it more difficult to evaluate model performance in terms of generalization. In particular, F1 (True) and F1 (False) exhibit low correlation, likely due to their sensitivity to label prevalence and class imbalance. While this does not make them invalid as evaluation metrics, it suggests that they may lead to inconsistent model rankings across different test distributions. Among the metrics we tested, accuracy (which is currently used in the official evaluation), as well as F1 (Macro) and F1 (Weighted), showed slightly better cross-split correlation. However, even advanced metrics such as Informedness, NIT, and MCC do not substantially mitigate the instability in this task. Similar tendencies, though somewhat less pronounced, were observed for the RE task.

5.3 Task Interdependence: Correlation Between TP and RE

The TP and RE tasks are conceptually related, as both are essential for judicial reasoning. The baseline IST model also benefited from multitask learning on both tasks [32]. A natural question is whether models that perform well on the TP task also excel at RE.

To investigate this, we trained 216 models (4 epochs $\times$ 6 learning rates $\times$ 9 α values) using a fixed data split. Each model was jointly trained on TP and RE. Table 5 shows the correlation between TP accuracy and several RE metrics: F1 (True), doc-level accuracy, and doc-level F1 (Macro).

Yamada et al. [32] proposed using *doc-level accuracy* for RE, motivated by the fact that real-world decisions are made at the tort level. This metric is computed by averaging claim-level accuracy within each tort and then taking the macro average across all torts. From this perspective, one might expect doc-level accuracy to align more closely with TP performance. However, our results show that F1 (True) has the highest correlation with TP accuracy (approximately 0.27), and both doc-level metrics exhibit even lower correlation. It remains unclear why these metrics, despite their alignment with real-world decision units, correlate less with TP performance, pointing to a direction worth exploring in future work.

5.4 Inter-Metric Correlations

Using the same 216-model collection, we also computed pairwise correlations between evaluation metrics within each task. Table 6 shows the results for both TP and RE.

In both tasks, F1 (True) and F1 (False) show weak correlations with all other metrics. Notably, their mutual correlation is extremely low: only 0.03 in TP and 0.05 in RE. This suggests that these scores are highly sensitive to subjective label definitions, such as whether `is_accepted` or `is_rejected` is treated as "true". Such sensitivity undermines their usefulness as general-purpose evaluation metrics.

In contrast, most other metrics exhibit strong mutual correlations, suggesting that model rankings are generally stable across metrics that are less sensitive to subjective factors like class labeling.

6 CONCLUDING REMARKS

In this paper, we presented the systems developed by Team KIS for the COLIEE 2025 Pilot Task, LJPJT 2025. Our system adopts a straightforward implementation: all claims are fed into ModernBERT in a unified sequence. Despite its simplicity, the system achieved competitive performance by leveraging model ensembling.

Table 5: Correlation between TP and RE task performance (same dataset).

			TP	RE				
			Accuracy	F1 (True)			Accuracy	F1 (Macro)
				All	Pltf.	Deft.	Doc-Level	Doc-Level
TP	Accuracy		1.0000	0.2711	0.2352	0.2404	0.1910	0.0850
RE	F1 (True)	All	0.2711	1.0000	0.8491	0.9138	0.6317	0.5744
		Pltf.	0.2352	0.8491	1.0000	0.5627	0.3904	0.3148
		Deft.	0.2404	0.9138	0.5627	1.0000	0.6891	0.6608
	Accuracy	Doc-Level	0.1910	0.6317	0.3904	0.6891	1.0000	0.9665
	F1 (Macro)	Doc-Level	0.0850	0.5744	0.3148	0.6608	0.9665	1.0000

Table 6: Cross-task performance correlation (same dataset).

a. TP task

	Accuracy	F1 (True)	F1 (False)	F1 (Macro)	F1 (Weighted)	Informedness	NIT	MCC
Accuracy	1.0000	0.5056	0.8803	0.8872	0.9460	0.9944	0.9532	0.9751
F1 (True)	0.5056	1.0000	0.0389	0.8456	0.7560	0.5614	0.6914	0.6808
F1 (False)	0.8803	0.0389	1.0000	0.5662	0.6835	0.8399	0.7220	0.7536
F1 (Macro)	0.8872	0.8456	0.5662	1.0000	0.9887	0.9118	0.9560	0.9641
F1 (Weighted)	0.9460	0.7560	0.6835	0.9887	1.0000	0.9603	0.9781	0.9910
Informedness	0.9944	0.5614	0.8399	0.9118	0.9603	1.0000	0.9691	0.9877
NIT	0.9532	0.6914	0.7220	0.9560	0.9781	0.9691	1.0000	0.9847
MCC	0.9751	0.6808	0.7536	0.9641	0.9910	0.9877	0.9847	1.0000

b. RE task

		Accuracy	F1 (True)	F1 (False)	F1 (Macro)	F1 (Weighted)	Accuracy	F1 (Macro)	F1 (Weighted)
		All	All	All	All	All	Doc-Level	Doc-Level	Doc-Level
Accuracy	All	1.0000	0.7596	0.6487	0.9182	0.9341	0.9499	0.9050	0.9039
F1 (True)	All	0.7596	1.0000	0.0517	0.5253	0.5692	0.6317	0.5744	0.5633
F1 (False)	All	0.6487	0.0517	1.0000	0.8769	0.8505	0.7175	0.7499	0.7652
F1 (Macro)	All	0.9182	0.5253	0.8769	1.0000	0.9986	0.9153	0.9154	0.9231
F1 (Weighted)	All	0.9341	0.5692	0.8505	0.9986	1.0000	0.9234	0.9199	0.9266
Accuracy	Doc-Level	0.9499	0.6317	0.7175	0.9153	0.9234	1.0000	0.9665	0.9659
F1 (Macro)	Doc-Level	0.9050	0.5744	0.7499	0.9154	0.9199	0.9665	1.0000	0.9980
F1 (Weighted)	Doc-Level	0.9039	0.5633	0.7652	0.9231	0.9266	0.9659	0.9980	1.0000

In addition, we analyzed the performance of a large number of models obtained through hyperparameter search. Our findings indicate that in the LJPJT 2025 dataset, performance varies significantly across data splits, complicating the evaluation of model performance with respect to generalization. We also observed that the performance of a given model can differ substantially depending on the evaluation metric used. This highlights persistent challenges in designing appropriate evaluation metrics. In particular, metrics such as F1 (True) and F1 (False) were found to be highly sensitive to subjective choices, such as how the "true" label is defined or how the dataset is split. These sensitivities raise concerns about their suitability for model evaluation.

For future work, we plan to explore more robust evaluation metrics, and to develop models that can better handle cases that are ambiguous or difficult to resolve, even for human experts [32]. This may involve incorporating additional external information sources into the modeling process.

ACKNOWLEDGMENTS

This research was partially supported by Kakenhi, MEXT Japan (JP22H00804, JP23K22076), JST PRESTO (JPMJPR2461), JST AIP Acceleration Research (JPMJCR22U4), and SECOM Science and Technology Foundation.

REFERENCES

[1] Onur Bilgin, Logan Fields, Antonio Laverghetta, Zaid Marji, Animesh Nighojkar, Stephen Steinle, and John Licato. 2024. Exploring Prompting Approaches in Legal Textual Entailment. *The Review of Socionetwork Strategies* 18, 1 (01 Apr 2024), 75–100. https://doi.org/10.1007/s12626-023-00154-y

[2] Kathrin Blagec, Georg Dorffner, Milad Moradi, Simon Ott, and Matthias Samwald. 2022. A global analysis of metrics used for measuring performance in natural language processing. In *Proceedings of NLP Power! The First Workshop on Efficient Benchmarking in NLP*. Association for Computational Linguistics, Dublin, Ireland, 52–63. https://doi.org/10.18653/v1/2022.nlppower-1.6

[3] Diogo V. Carvalho, Eduardo M. Pereira, and Jaime S. Cardoso. 2019. Machine Learning Interpretability: A Survey on Methods and Metrics. *Electronics* 8, 8 (July 2019), 832. https://doi.org/10.3390/electronics8080832

[4] Ilias Chalkidis, Ion Androutsopoulos, and Nikolaos Aletras. 2019. Neural Legal Judgment Prediction in English. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Florence, Italy, 4317–4323. https://doi.org/10.18653/v1/P19-1424

[5] 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. https://doi.org/10.18653/v1/2022.acl-long.297

[6] Davide Chicco, Niklas Tötsch, and Giuseppe Jurman. 2021. The Matthews correlation coefficient (MCC) is more reliable than balanced accuracy, bookmaker informedness, and markedness in two-class confusion matrix evaluation. *BioData Mining* 14, 1 (04 Feb 2021), 13. https://doi.org/10.1186/s13040-021-00244-z

[7] 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)*. Association for Computational Linguistics, Minneapolis, Minnesota, 4171–4186. https://doi.org/10.18653/v1/N19-1423

[8] Lucas Dixon, John Li, Jeffrey Sorensen, Nithum Thain, and Lucy Vasserman. 2018. Measuring and Mitigating Unintended Bias in Text Classification. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society* (New Orleans, LA, USA) *(AIES '18)*. Association for Computing Machinery, New York, NY, USA, 67–73. https://doi.org/10.1145/3278721.3278729

[9] Masaki Fujita, Takaaki Onaga, Ayaka Ueyama, and Yoshinobu Kano. 2023. Legal Textual Entailment Using Ensemble of Rule-Based and BERT-Based Method with Data Augmentation by Related Article Generation. In *New Frontiers in Artificial Intelligence*. Springer Nature Switzerland, Cham, 138–153. https://doi.org/10.1007/978-3-031-29168-5_10

[10] Randy Goebel, Yoshinobu Kano, Mi-Young Kim, Calum Kwan, Ken Satoh, Hiroaki Yamada, and Masaharu Yoshioka. 2025. Overview of the COLIEE 2025 Competition: Legal Case Law and Statute Law Information Retrieval and Entailment. In *Proceedings of COLIEE 2025 workshop*.

[11] Max Hort, Zhenpeng Chen, Jie M. Zhang, Mark Harman, and Federica Sarro. 2024. Bias Mitigation for Machine Learning Classifiers: A Comprehensive Survey. *ACM J. Responsib. Comput.* 1, 2, Article 11 (June 2024), 52 pages. https://doi.org/10.1145/3631326

[12] Maksims Ivanovs, Roberts Kadikis, and Kaspars Ozols. 2021. Perturbation-based methods for explaining deep neural networks: A survey. *Pattern Recognition Letters* 150 (2021), 228–234. https://doi.org/10.1016/j.patrec.2021.06.030

[13] Mi-Young Kim, Juliano Rabelo, Housam Khalifa Bashier Babiker, Md Abed Rahman, and Randy Goebel. 2024. Legal Information Retrieval and Entailment Using Transformer-based Approaches. *The Review of Socionetwork Strategies* 18, 1 (01 Apr 2024), 101–121. https://doi.org/10.1007/s12626-023-00153-z

[14] 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. arXiv:1907.11692 [cs.CL] https://arxiv.org/abs/1907.11692

[15] Henry Moss, David Leslie, and Paul Rayson. 2018. Using J-K-fold Cross Validation To Reduce Variance When Tuning NLP Models. In *Proceedings of the 27th International Conference on Computational Linguistics*. Association for Computational Linguistics, Santa Fe, New Mexico, USA, 2978–2989. https://aclanthology.org/C18-1252/

[16] Joel Niklaus, Ilias Chalkidis, and Matthias Stürmer. 2021. Swiss-Judgment-Prediction: A Multilingual Legal Judgment Prediction Benchmark. In *Proceedings of the Natural Legal Language Processing Workshop 2021*. Association for Computational Linguistics, Punta Cana, Dominican Republic, 19–35. https://doi.org/10.18653/v1/2021.nllp-1.3

[17] Takaaki Onaga, Masaki Fujita, and Yoshinobu Kano. 2024. Contribution Analysis of Large Language Models and Data Augmentations for Person Names in Solving Legal Bar Examination at COLIEE 2023. *The Review of Socionetwork Strategies* 18, 1 (01 Apr 2024), 123–143. https://doi.org/10.1007/s12626-024-00155-5

[18] Takaaki Onaga and Yoshinobu Kano. 2025. KIS: COLIEE 2025 Task 4 Solver Using Japanese LLM. In *Proceedings of COLIEE 2025 workshop*.

[19] OpenAI. 2024. GPT-4o System Card. arXiv:2410.21276 [cs.CL] https://arxiv.org/abs/2410.21276

[20] Juri Opitz. 2024. A Closer Look at Classification Evaluation Metrics and a Critical Reflection of Common Evaluation Practice. *Transactions of the Association for Computational Linguistics* 12 (2024), 820–836. https://doi.org/10.1162/tacl_a_00675

[21] 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 (01 Apr 2022), 111–133. https://doi.org/10.1007/s12626-022-00105-z

[22] Oona Rainio, Jarmo Teuho, and Riku Klén. 2024. Evaluation metrics and statistical tests for machine learning. *Scientific Reports* 14, 1 (13 Mar 2024), 6086. https://doi.org/10.1038/s41598-024-56706-x

[23] Sunzida Siddique, Mohd Ariful Haque, Roy George, Kishor Datta Gupta, Debashis Gupta, and Md Jobair Hossain Faruk. 2023. Survey on Machine Learning Biases and Mitigation Techniques. *Digital* 4, 1 (Dec. 2023), 1–68. https://doi.org/10.3390/digital4010001

[24] Gemini Team. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. arXiv:2403.05530 [cs.CL] https://arxiv.org/abs/2403.05530

[25] Francisco José Valverde-Albacete, Jorge Carrillo-de Albornoz, and Carmen Peláez-Moreno. 2013. A Proposal for New Evaluation Metrics and Result Visualization Technique for Sentiment Analysis Tasks. In *Information Access Evaluation. Multilinguality, Multimodality, and Visualization*. Springer Berlin Heidelberg, Berlin, Heidelberg, 41–52.

[26] 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*, Vol. 30. Curran Associates, Inc., 11. https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

[27] Peter Vickers, Loic Barrault, Emilio Monti, and Nikolaos Aletras. 2023. We Need to Talk About Classification Evaluation Metrics in NLP. In *Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Nusa Dua, Bali, 498–510. https://doi.org/10.18653/v1/2023.ijcnlp-main.33

[28] 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: Multitask and Ensemble Approaches in Legal Information Processing. *The Review of Socionetwork Strategies* 18, 1 (01 Apr 2024), 145–165. https://doi.org/10.1007/s12626-024-00157-3

[29] Benjamin Warner, Antoine Chaffin, Benjamin Clavié, Orion Weller, Oskar Hallström, Said Taghadouini, Alexis Gallagher, Raja Biswas, Faisal Ladhak, Tom Aarsen, Nathan Cooper, Griffin Adams, Jeremy Howard, and Iacopo Poli. 2024. Smarter, Better, Faster, Longer: A Modern Bidirectional Encoder for Fast, Memory Efficient, and Long Context Finetuning and Inference. arXiv:2412.13663 [cs.CL] https://arxiv.org/abs/2412.13663

[30] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-Art Natural Language Processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Association for Computational Linguistics, Online, 38–45. https://doi.org/10.18653/v1/2020.emnlp-demos.6

[31] Chaojun Xiao, Haoxi Zhong, Zhipeng Guo, Cunchao Tu, Zhiyuan Liu, Maosong Sun, Yansong Feng, Xianpei Han, Zhen Hu, Heng Wang, and Jianfeng Xu. 2018. CAIL2018: A Large-Scale Legal Dataset for Judgment Prediction. arXiv:1807.02478 [cs.CL] https://arxiv.org/abs/1807.02478

[32] 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. https://doi.org/10.1007/s10506-024-09402-0

Leveraging Rhetorical Role-Based Summarization for Legal Case Retrieval

Tebo K. Leburu-Dingalo
leburut@ub.ac.bw
University of Botswana
Gaborone, Botswana

Edwin Thuma
University of Botswana
Gaborone, Botswana
thumae@ub.ac.bw

Gontlafetse Mosweunyane
University of Botswana
Gaborone, Botswana
mosweuny@ub.ac.bw

Nkwebi P. Motlogelwa
University of Botswana
Gaborone, Botswana
motlogel@ub.ac.bw

Monkgogi N. Mudongo
University of Botswana
Gaborone, Botswana
mudongom@ub.ac.bw

Abstract

Legal case retrieval plays a crucial role in the legal research process as it enables law practitioners, such as judges and lawyers, to efficiently identify relevant prior cases or precedents for ongoing cases. However, improving retrieval accuracy remains a challenge due to the complexity, lengthiness, and unstructured nature of legal texts. In this study, we propose approaches that integrate structural representation and summarization based on rhetorical roles to enhance case retrieval performance. In particular, we introduce methods where query sentences are labeled with legal rhetorical roles, and concise versions of the queries built from fact sentences are matched against both similarly summarized candidate cases and un-summarized versions. We also explore score-based filtering of the initial retrieval results. While our approaches do not perform well in the official task, we note that the method that combines filtering with summarized queries and un-summarized cases gives better performance than our other approaches.

CCS Concepts

• **Information systems** → **Retrieval models and ranking**.

Keywords

Legal case retrieval, rhetorical roles, summarization, query representation

ACM Reference Format:
Tebo K. Leburu-Dingalo, Edwin Thuma, Gontlafetse Mosweunyane, Nkwebi P. Motlogelwa, and Monkgogi N. Mudongo. 2025. Leveraging Rhetorical Role-Based Summarization for Legal Case Retrieval. In *Proceedings of COLIEE 2025 workshop, June 20, 2025, Chicago, USA.* ACM, New York, NY, USA, 4 pages.

1 Introduction

As the amount of digital legal text has grown exponentially over the years, there has been growing interest among researchers and institutions to develop efficient retrieval methods to access this vast resource. One such initiative is the Competition on Legal Information Extraction and Entailment (COLIEE)[11]. COLIEE is an annual event that provides an environment for researchers to develop and evaluate innovative systems aimed at improving access to legal text. To facilitate this, the competition avails benchmark datasets that address various aspects of legal text access organized into different tasks. For the current year, the competition consists of five tasks. Task 1 requires the retrieval of a set of existing case law cases that can support the decision of a given query case. For Task 2 the requirement is to identify a paragraph from an existing case that can entail the decision of a new case. Task 3 focuses on retrieving statutory articles that are relevant to a previously unseen query case. Task 4 involves determining whether relevant Civil Law articles retrieved for a legal bar exam question entail it or not. The last task, a Pilot Task, consists of two subtasks that deal with Tort cases. The first subtask, Tort Prediction (TP), involves predicting whether a tort is affirmed given facts as well as arguments from plaintiffs and defendants. The second subtask Rationale Extraction (RE) focuses on predicting which arguments from both plaintiffs and defendants will be accepted or rejected.

In this paper, we present our three approaches to Task 1, case law retrieval. Since the task requires retrieving a set of relevant legal cases that can support the decision of a given query case, we posit that in addition to being lexically similar, retrieved cases should also be structurally identical to the given query cases. We thus explore in one of our approaches the effectiveness of building structure into both the query case and existing candidate cases before retrieval. We define building structure in this instance as identifying and differentiating sentences according to legal rhetorical roles such as facts and arguments and selecting only the most effective role/s, which in our approach is facts, to represent both the query and candidates during retrieval. We further incorporate retrieval score filtering in an attempt to improve our results. This involves removing documents with relevance scores below a predefined threshold after the initial retrieval stage. Our other two approaches are a variation of this approach aimed at testing its validity. Specifically, in an approach that acts as our baseline, we only represent queries as facts and retain candidates in the original form with no filtering of retrieval results. In the last approach, we adopt the baseline but with a further filtering of retrieval results.

Our paper is organized as follows: Section 2 presents related work, Section 3 describes our approaches in more detail, Section

COLIEE 2025, Chicago, USA

4 presents the experimental setup, Section 5 discusses the results, and Section 6 is the conclusion and also proposes future work.

2 Related Work

Automated case law retrieval plays a vital role in legal research, as it allows lawyers, judges, and other stakeholders to efficiently access prior cases that can support the decision applicable to an ongoing case. However, developing an effective case law retrieval system remains a significant challenge due to the length and complex structure of legal documents, as well as the complexity of the legal language. In recent years, in an effort to address these challenges and improve performance, many researchers have been incorporating advanced technologies such as natural language processing, and artificial intelligence into their systems. Specifically, many of the latest COLIEE approaches, including the state-of-the-art have used deep learning methods and Large Language Models(LLMs) in addition to traditional IR methods. The following are some approaches that have performed relatively well in recent versions of the COLIEE competition.

Li et al.[8] deploy a learning to rank based approach using a diverse set of features generated from lexical matching, and pretrained semantic retrieval models. These include features such as the BM25 query-candidate score, and the documents' SAILER[6] and DELTA[7] rank scores. The approach also incorporates preprocessing to remove irrelevant information such as place holder text. Furthermore, post-processing is performed to reduce irrelevant documents such as deleting duplicate query cases from retrieval results.

Curran and Conway[3] develop a pairwise similarity ranking framework by training a feedforward neural network to perform binary classification. The framework uses multiple features derived from each query-candidate case pair, such as the name of the presiding judge and verbatim quotations.

Li et al.[9] implement a learning-to-rank approach that utilizes various features such as query length as well as features generated from a pre-trained structure-aware language model SAILER[6]. The approach also incorporates preprocessing to remove irrelevant terms and phrases. A post-processing strategy is also applied to, for instance, remove query cases from results and also filter out cases with a trial date later than the query case.

Derbama[4] use a query reformulation method that entails scoring query unigram terms to select representative terms for the query. BM25 is used for retrieval and the results are processed to remove retrieved documents with a later year than the query. Furthermore, a threshold-based method is used to select the final set of relevant documents. Preprocessing is also used to remove irrelevant information such as place holders and punctuations.

It is evident from these approaches that the use of hybrid approaches and a combination of various query-document features can improve retrieval model performance. In addition, through post-processing, precision can be improved while maintaining a relatively good recall. Finally, as demonstrated in [9] and [8], utilizing structurally aware methods also has the potential to enhance overall retrieval performance. Based on these observations, we aim to explore a retrieval approach that attempts to build structure into the case documents through the identification of sentences' rhetorical roles. The approach will further incorporate preprocessing of the text to remove uninformative content, quality sentence selection based on a lexical informativeness score, and post-processing to improve precision.

3 Methods

3.1 Task Description

The goal of the COLIEE 2025 Case Law Retrieval Task 1 is to extract and return from a Case Law corpus "noticed cases" (S1, S2…Sn), for a given unseen case Q. A case is considered "noticed" only if it can support the decision of the case Q. Therefore, the task involves searching a collection of legal case documents, and only retrieving those that are relevant to a given query case.

The evaluation metrics for the dataset are precision, recall, and the F1 measure. Precision calculates the proportion of retrieved documents that are actually relevant to a query, while recall calculates the proportion of relevant documents in a collection that are successfully retrieved for a query. The F1 measure is a harmonic mean of precision and recall which provides a single metric that balances both. Micro-average is used for all the metrics, meaning that each measure is computed using the results of all queries. The metrics are defined as follows:

$$\text{Precision} = \frac{\#\text{ of correctly retrieved cases for all queries}}{\#\text{ of retrieved cases for all queries}} \quad (1)$$

$$\text{Recall} = \frac{\#\text{ of correctly retrieved cases for all queries}}{\#\text{ of relevant cases for all queries}} \quad (2)$$

$$F - measure = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \quad (3)$$

3.2 Approach

Our proposed approach generally relies on three key principles that have been shown to enhance legal case retrieval performance. The first is that a query case can be effectively represented using a summarized or shortened version. Secondly, preprocessing, and incorporating structural elements into both query and case documents can enhance their processing and improve retrieval accuracy. Finally, post-processing techniques can refine results and improve precision through the removal of less relevant results.

We thus formulate an approach that incorporates a machine learning component to build structure into the documents, a summarization component, and a retrieval component to identify and return relevant cases. To build structure, we formulate a classification task that identifies the rhetorical role of each sentence in a document. The roles are based on the semantic function that a sentence is associated with in the text such as facts, arguments, or statute. Filtering of results based on an experimentally selected threshold is also explored.

Our overall pipeline comprises of the following main components:

(1) **Preprocessing to remove non-informative content and select informative sentences**
All documents are first preprocessed to remove non-informative content, followed by sentence filtering to retain only the

most informative sentences. To assess sentence informativeness, we employ the Measure of Textual Lexical Diversity (MTLD) [10], which quantifies lexical diversity by computing the number of distinct words within a text.

(2) **Development of a classifier to identify sentence rhetorical role**
A classification model is trained using an external dataset consisting of sentences with rhetorical role labels. For this task, we employ a gradient-boosting ensemble learning technique that builds multiple weak learners, which are typically decision trees, and combines them to build a stronger predictive model [5].

(3) **Sentence Labeling and Summarization**
The classifier is deployed to assign a rhetorical role to each sentence indicating its semantic function in the document. Rhetorical roles used are facts,arguments, statute,precedent, ratio of the decision, ruling by lower court, and ruling by present court. Facts sentences are then extracted and used to generate summary version of each document.

(4) **Retrieval, Ranking and Results Filtering**
For retrieval and ranking, the DPH parameter-free weighting model from the Divergence From Randomness (DFR) framework[1] is used. As a final step post-processing is implemented in an attempt to enhance precision. This process involves removing "noticed" cases that share the same ID as the query case and applying retrieval score-based filtering to refine the results.

4 Experimental Setup

4.1 Dataset

The dataset for the COLIEE 2025 Case Law Retrieval Task 1 consists of judgments drawn from the Federal Court of Canada. The training dataset includes 7350 candidate cases with 1678 identified as query cases. Provided with the training set is a JSON file that maps each query case to its respective set of "noticed" cases. For testing, a dataset containing 2159 candidate cases is provided with 400 identified as query cases. The test JSON file only lists query cases, as the task requires participants to identify "noticed cases" for each query case.

4.2 Experiments

All our experiments are conducted on Google Colab using Python and related libraries.

- Text Preprocessing: Non-informative content such as punctuations, special characters, French text and phrases such as <"Fragment Suppressed"> was removed from text. The text was split into sentences, and each sentence was assigned a lexical diversity score to indicate its informativeness. The diversity score was calculated using the LexicalRichness[1] Python module. Sentences with low scores were then filtered out.
- Development of a classifier to identify sentence rhetorical role: The sklearn Gradient Boosting Classifier[2] was trained and tested on an Artificial Intelligence for Legal Assistance (AILA) dataset consisting of rhetorically role labeled sentences. The dataset created by Bhattacharya et.al [2], utilizes sentences drawn from the Supreme Court of India judgments. Each sentence was represented using a vector of 186 Stylometry features generated using the writeprints[3] package.
- Sentence Labeling and Summarization: A set of writeprints features was extracted for sentences in both candidate documents and query documents and the classifier used to assign each sentence an applicable rhetorical role. Sentences labeled as facts were extracted and used to generate summaries for each candidate document, and each query. The facts role was selected to generate summaries as it demonstrated superior performance when compared to other roles in preliminary experiments.
- Retrieval, Ranking and Results Filtering: Various experiments were conducted to retrieve and rank documents using the PyTerrier (Python Terrier) framework DPH model. The DPH model was selected due to its demonstrated effectiveness during preliminary experiments, where it achieved better performance compared to BM25 and TF-IDF. From these experiments, three were selected as our final runs.
  - In the first run summarized queries were matched against original case documents which had undergone only basic preprocessing. The ranking cut-off was set at 50, and post-processing was applied to remove documents that were duplicates of their respective queries.
  - In the second run, the first run was repeated. However, an additional step was introduced. Specifically, score-based filtering was incorporated in an effort to improve precision by reducing the number of irrelevant documents in the ranked results.
  - In the final run summarized queries are matched against summarized case documents. Low-scoring documents and those identified as duplicates of their respective queries were removed to improve precision.

5 Results and Discussion

Results from the three final runs were submitted to COLIEE for evaluation. Table 1 shows the official evaluation results with our entries labeled UB_2025. It can be observed that our best performing in terms of the F1 score is the second run (run2.txt) where we used summarized queries with un-summarized candidate case documents and applied filtering. This gives an indication that summarizing candidate documents (run3.txt) has a minor impact in terms of improving performance in this task. Hence, it can be inferred that the task can benefit more from finding better methods to summarize query cases, especially when deploying traditional IR methods for retrieval. Despite a high recall score, our first run (run1.txt) is the worst performing in terms of overall performance, highlighting the importance of results post-processing. These results align with findings from preliminary experiments, reinforcing earlier observations that summarizing query cases based on significant rhetorical roles can enhance retrieval performance as opposed to using non-summary versions.

[1] https://pypi.org/project/lexicalrichness/
[2] https://github.com/scikit-learn/scikit-learn/blob/98ed9dc73/sklearn/ensemble/_gb.py#L1126
[3] https://pypi.org/project/writeprints/

Table 1: COLIEE 2025 Case Law Retrieval Task 1 Results

Team	File	F1	Precision	Recall
JNLP	jnlpr&fe2.txt	0.3353	0.3042	0.3735
JNLP	jnlpr&fe1.txt	0.3267	0.2945	0.3667
UQLegalAI	uqlegalair3.txt	0.2962	0.2908	0.3019
UQLegalAI	uqlegalair2.txt	0.2957	0.2903	0.3013
UQLegalAI	uqlegalair1.txt	0.2940	0.2886	0.2996
NOWJ	prerank_dense_bge-rerank_bge_ft_llm2vec_major_vote.txt	0.1984	0.1670	0.2445
AIIR Lab	task1.aiirmpmist5.txt	0.2171	0.2040	0.2319
NOWJ	prerank_dense_bge-rerank_bge_ft.txt	0.1708	0.1605	0.1825
AIIR Lab	task1.aiircombmnz.txt	0.1879	0.2317	0.1580
AIIR Lab	task1.aiirmpmist3.txt	0.1872	0.2308	0.1575
NOWJ	prerank_dense_llm2vec_llama31_8b.txt	0.1580	0.1485	0.1688
JNLP	jnlpfe1.txt	0.1597	0.1307	0.2052
OVGU	task1_ovgu2.txt	0.1498	0.1743	0.1313
UB_2025	run2.txt	0.1363	0.1955	0.1046
UB_2025	run3.txt	0.1171	0.1818	0.0864
UB_2025	run1.txt	0.1051	0.0572	0.6379
SIL	submission_sil_run_results.txt	0.0058	0.0054	0.0063
UA	ua_run3.txt	0.0000	0.0000	0.0000
UA	ua_run2.txt	0.0000	0.0000	0.0000
UA	ua_run1.txt	0.0000	0.0000	0.0000
OVGU	ignore_task1_ovgu1.txt	0.0000	0.0000	0.0000

6 Conclusion

In this paper, we present approaches we submitted towards the COLIEE 2025 Case Law Retrieval Task 1. In an effort to enhance performance, we adopted a strategy that focused on implementing three key principles learnt from the literature, building structure into the documents, text summarization, and post processing of results. While our approaches did not reach expected performance, they have provided valuable insights that can guide future improvements. These include the need to further investigate the effectiveness of adopting structuring, summarization, and advanced filtering strategies, as well as identifying ways to enhance their effectiveness. Hence going forward, we aim to conduct additional experiments that incorporate advanced techniques such as transformer-based models for role detection, and the use of semantic retrieval models. We will further experiment with using different types of features, varying role classes as well as pre and post filtering strategies such as neural re-ranking. A failure analysis will also be needed to examine the impact of different combinations of components and methods within our overall pipeline. This will help us to identify potential weaknesses and thus allow us to refine our approach towards attaining improved retrieval performance.

References

[1] Gianni Amati, Edgardo Ambrosi, Marco Bianchi, Carlo Gaibisso, and Giorgio Gambosi. 2007. FUB, IASI-CNR and University of Tor Vergata at TREC 2007 Blog Track. In *Text Retrieval Conference*. https://api.semanticscholar.org/CorpusID:267819474

[2] Paheli Bhattacharya, Shounak Paul, Kripabandhu Ghosh, Saptarshi Ghosh, and Adam Zachary Wyner. 2019. Identification of Rhetorical Roles of Sentences in Indian Legal Judgments. In *International Conference on Legal Knowledge and Information Systems*. https://api.semanticscholar.org/CorpusID:207930532

[3] 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.

[4] 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)*.

[5] Jerome H Friedman. 2001. Greedy function approximation: a gradient boosting machine. *Annals of statistics* (2001), 1189–1232.

[6] 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.

[7] Haitao Li, Qingyao Ai, Xinyan Han, Jia Chen, Qian Dong, Yiqun Liu, Chong Chen, and Qi Tian. 2024. DELTA: Pre-train a Discriminative Encoder for Legal Case Retrieval via Structural Word Alignment. *arXiv preprint arXiv:2403.18435* (2024).

[8] 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.

[9] 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).

[10] Philip M McCarthy. 2005. *An assessment of the range and usefulness of lexical diversity measures and the potential of the measure of textual, lexical diversity (MTLD)*. Ph. D. Dissertation. The University of Memphis.

[11] 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.

Hierarchical and Referential Structure-Aware Retrieval for Statutory Articles using Graph Neural Networks

Takao Mizuno
FRAIM Inc.
Tokyo, Japan
t.mizuno@fraim.co.jp

Yoshinobu Kano
Shizuoka University
Shizuoka, Japan
kano@inf.shizuoka.ac.jp

Abstract

Statutory Article Retrieval (SAR) is a key technology for enabling legal professionals and the general public to access relevant legal information. However, accurately retrieving statutory articles remains challenging due to the need to interpret the hierarchical organization of legal texts and the referential dependencies among provisions. In particular, Japanese statutes exhibit multi-level hierarchical structures, with lower-level articles often relying on higher-level contextual assumptions, and frequently include explicit references to other articles. To address these challenges, we propose a structure-aware retrieval method based on Graph Neural Networks (GNNs), designed for COLIEE 2025 Task 3, which involves retrieving relevant articles from the Japanese Civil Code. Our model, the Japanese Legal Graph Retriever (JLGR), represents statutory structure as a directed graph and incorporates citation information by recursively inlining referenced article texts into citing articles. A GNN is applied to propagate contextual signals across the graph, enriching article representations with structural information. JLGR follows a two-stage retrieval architecture: a GNN-augmented bi-encoder is used for efficient candidate retrieval, followed by a cross-encoder that re-ranks top candidates via fine-grained query–article interactions. We participated in the COLIEE 2025 formal run as Team INFA and evaluated our system on the official Task 3 dataset using the F_2 score as the primary metric. JLGR ranked 3rd out of 8 teams and demonstrated superior performance over contrastive learning baselines, confirming the effectiveness of incorporating legal structure into article retrieval.

CCS Concepts

• **Information systems** → **Retrieval models and ranking**; • **Computing methodologies** → *Neural networks*; • **Applied computing** → **Law**.

Keywords

Statutory Article Retrieval, Japanese Civil Code, Graph Neural Networks, Contrastive Learning, Reranking, Bi-Encoder, Cross-Encoder

ACM Reference Format:
Takao Mizuno and Yoshinobu Kano. 2025. Hierarchical and Referential Structure-Aware Retrieval for Statutory Articles using Graph Neural Networks. In *Proceedings of COLIEE 2025 workshop, June 20, 2025, Chicago, USA.* ACM, New York, NY, USA, 10 pages.

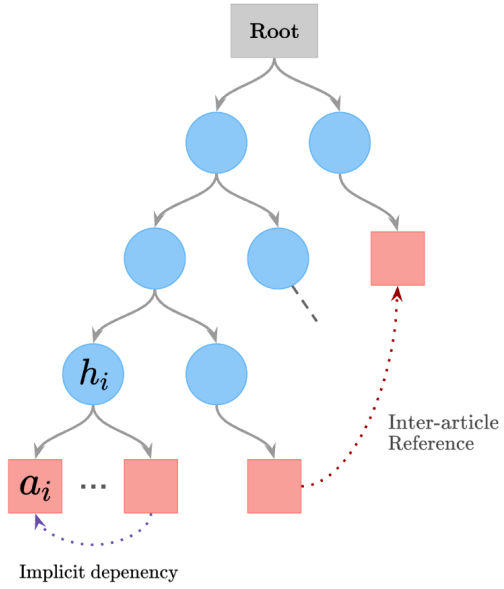


Figure 1: Conceptual illustration of hierarchical and referential dependencies in Japanese statute law. Blue circles represent structural units (h_i), such as `part`, `chapter`, `section`, and `caption`(treated as hierarchical due to its topical relevance). Red squares denote statutory articles (a_i). Solid arrows represent containment; dashed lines indicate omitted substructures. The red dotted arrow represents an inter-article reference; the purple dotted arrow indicates an implicit dependency on a neighboring article. These dependencies illustrate challenges in interpreting articles in isolation. For formal definitions of a_i and h_i, see Sections 3.1 and 4.2.

1 Introduction

The Competition on Legal Information Extraction and Entailment (COLIEE) is an annual shared task series aimed at advancing research in legal information retrieval and textual entailment [7]. COLIEE has served as a benchmark since 2014 and now consists of four subtasks: case law retrieval (Task 1), case law entailment (Task 2), statute law retrieval (Task 3), and statute law entailment (Task 4). Tasks 3 and 4 are based on the Japanese Civil Code and

use questions from the Japanese bar exam. In Task 3, the focus of this paper, participants must retrieve a subset of articles from the Civil Code that are relevant to answering a legal question. Since this retrieval serves as a preprocessing step for textual entailment in Task 4, recall is emphasized, and the macro-averaged F_2 score is used as the official evaluation metric. COLIEE 2025 uses questions from the 2024 bar exam for evaluation. Recent years have seen a variety of approaches to Task 3, including lexical retrieval with BM25, dense retrieval with pretrained encoders, and hybrid systems using large language models (LLMs) for reranking.

Beyond the competition setting, Task 3 corresponds to the broader challenge of statutory article retrieval (SAR), which is a fundamental technology that enables legal professionals and general users to efficiently access relevant legal information when faced with legal issues. In SAR, a central task is to automatically identify all statutory articles relevant to a given natural language query. However, one of the challenges in SAR lies in the difficulty of accurately interpreting legal provisions when read in isolation. Statutory texts, particularly in Japanese statutes, often follow structural and contextual conventions that are not self-contained at the article level. These challenges are rooted in the structural and semantic characteristics of Japanese statutes. Figure 1 illustrates how statutory articles are situated within a layered legal structure and interconnected through both explicit and implicit dependencies. Such dependencies complicate retrieval when articles are interpreted in isolation. Notably, the following three elements are considered potential factors that may affect retrieval performance.

(1) **Dependence on hierarchical structure.** Japanese statutory articles are typically situated at the lowest level of a layered legal structure composed of multiple levels such as parts, chapters, sections, subsections, divisions, and captions; a caption is not formally a structural element, but because it often conveys important topical or contextual cues, we treat it as part of the hierarchy in this work. Each level in this hierarchy is assigned a heading or label that provides a high-level semantic description of the legal content under it. These hierarchical labels often play a crucial role in guiding human interpretation and filtering of relevant articles, and are therefore important context for retrieval models to consider (Figure 2).

(2) **Implicit dependencies on neighboring articles.** Some articles depend on preceding context, even without explicitly referring to other provisions [17]. These implicit dependencies arise when an article relies on definitions or assumptions introduced earlier in the text. For example, Article 89 of the Japanese Civil Code states that "The ownership of natural fruits is acquired by the person entitled to obtain them when they are separated from the original thing[1]." However, the term "natural fruits (fructus naturales)" is not defined within this article itself but in the preceding Article 88; without reference to such contextual definitions, the scope and interpretation of Article 89 remain ambiguous. This illustrates that accurate understanding of certain provisions requires prior contextual information embedded in neighboring articles (Figure 3).

(3) **Explicit inter-article references.** Many articles include direct references to other articles using expressions such as "pursuant to the preceding Article" or "in accordance with Article X." These references form semantic dependencies that span multiple articles. When a retrieval model treats each article as an independent unit, such references may be overlooked, leading to incomplete or incorrect retrieval outcomes (Figure 4).

[1]The English translations of statutory provisions cited in this paper are based on those provided in the COLIEE Task 3 dataset. These translations are intended for research purposes and may not always correspond to official or legally precise versions.

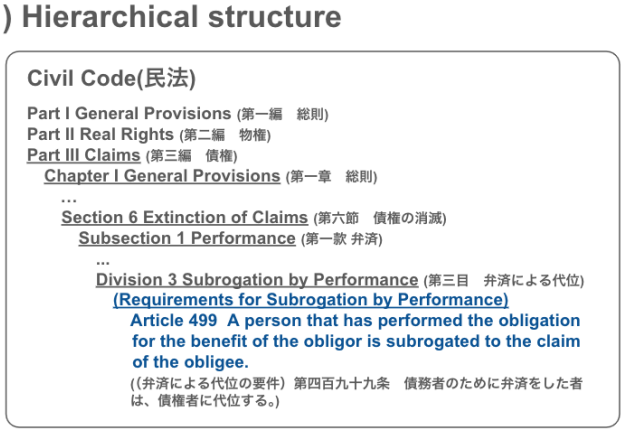


Figure 2: Hierarchical structure of Japanese statutory law. Articles are nested within multiple structural levels such as parts, chapters, sections, subsections, divisions, and captions. This layered structure provides crucial contextual cues for interpretation.

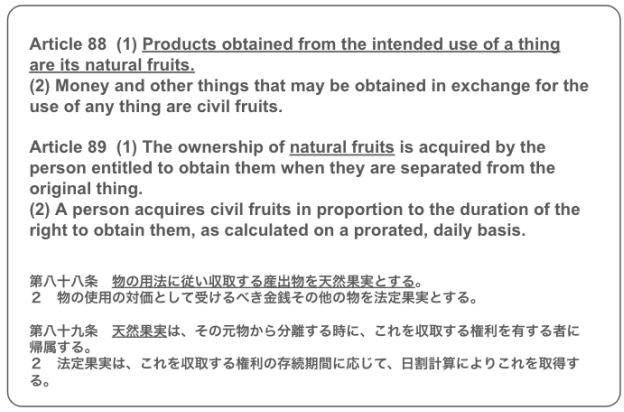


Figure 3: Implicit dependencies between articles. Article 89 assumes the reader understands the term "natural fruits(fructus naturales)," which is defined in the preceding Article 88. Such context-dependent interpretation poses challenges for article-level retrieval.

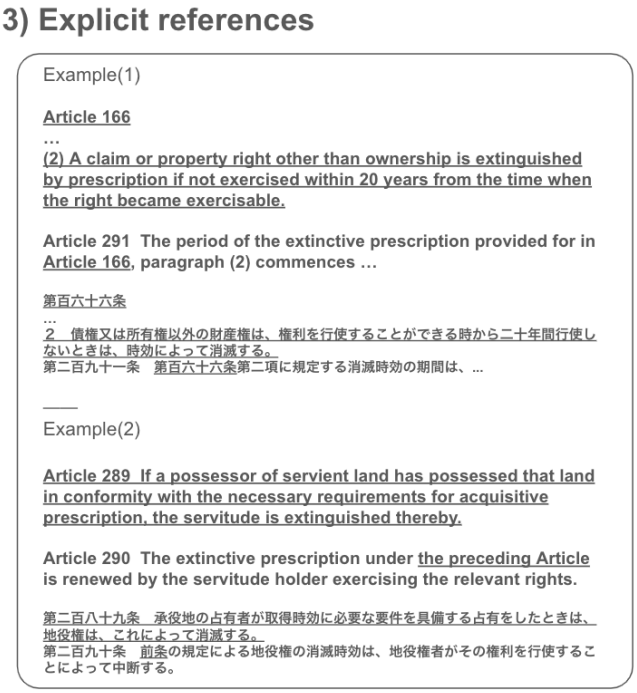


Figure 4: Explicit references across articles. Article 291 directly refers to Article 166, and Article 290 to Article 289, creating explicit semantic links that retrieval systems must account for.

Developing retrieval models that explicitly capture these structural and contextual aspects may help improve semantic alignment between queries and articles, thereby enhancing retrieval performance.

To address these challenges, recent studies have explored various directions that are particularly relevant to this work. One approach involves leveraging graph-based retrieval methods to model structured relationships between legal provisions.

Graph Neural Networks (GNNs) have been employed in retrieval systems to enhance document representations with relational signals. They are particularly effective in modeling complex structured text data by capturing relationships between words, documents, and corpus-level features. For example, Albarede et al. [2] explored using Heterogeneous Graph Attention Networks for passage retrieval, incorporating contextual information to improve relevance estimation. Similarly, Wang et al. [29] provide a survey of GNN applications in text retrieval, highlighting their ability to model document-level and corpus-level structures beyond surface-level semantics. Several approaches have also explored hierarchical retrieval methods to better handle document structure. Liu et al. [15] proposed Dense Hierarchical Retrieval (DHR), which generates passage representations that incorporate both document-level semantics and passage-specific context. Wang et al. [30] introduced a benchmark for Document-Aware Passage Retrieval (DAPR), addressing the limitations of passage retrievers that fail to consider document context, and demonstrated that contextualizing passage representations with document information improves retrieval performance on challenging queries. In the legal domain, G-DSR [17] proposed a graph-augmented bi-encoder for statutory article retrieval, which encodes hierarchical dependencies with a GNN. Specifically, G-DSR constructs a document graph representing the hierarchical structure of legislative texts and employs a GNN to learn structure-aware embeddings that capture both the content of each text unit and its position within the broader legislative framework. This model demonstrated that incorporating statutory structure can significantly improve semantic alignment between legal queries and articles.

As a complementary strategy for improving retrieval performance, many modern systems have adopted two-stage retrieval architectures [14, 20]. Traditional sparse vector space models such as TF-IDF [25] and BM25 [24] rely heavily on lexical overlap and suffer from vocabulary mismatch [8]. To address these limitations, dense retrieval methods have emerged, mapping queries and documents into continuous vector spaces to enable semantically meaningful matching. Unlike sparse vectors whose dimensionality depends on vocabulary size, dense vectors capture semantic information in a fixed-dimensional space.

The two-stage retrieval architecture, which combines the efficiency of bi-encoders with the accuracy of cross-encoders, strikes a balance between retrieval speed and ranking quality. A *bi-encoder* encodes the query and each document separately into dense vectors, typically using shared or analogous encoders, even across heterogeneous inputs. This independent encoding enables efficient retrieval via approximate nearest neighbor search. In contrast, a *cross-encoder* jointly processes the query and each candidate document by concatenating them and computing token-level interactions through attention mechanisms, enabling more accurate reranking at higher computational cost [1]. This retrieve-then-rerank approach has proven effective in large-scale settings.

The bi-encoder paradigm has evolved with advances in neural representation learning, from early models like DSSM [9] to more recent Transformer-based encoders [28] such as BERT [6] and Sentence-BERT [23]. A particularly influential bi-encoder architecture is Dense Passage Retrieval (DPR) [10], which introduced a dual-encoder framework for open-domain question answering. DPR demonstrated that dense retrievers trained solely on question–passage pairs can outperform sparse lexical models such as BM25. It has since laid the foundation for many dense retrieval systems. Building on this, multilingual embedding models such as Multilingual E5 [31] extend DPR-style training to over 100 languages and have achieved strong performance in semantic search, bitext mining, and multilingual retrieval tasks. For Japanese text, Ruri [26] is a general-purpose embedding model that combines contrastive pre-training on LLM-generated QA and NLI data with supervised fine-tuning using high-quality Japanese datasets. It adopts a dual-encoder architecture with knowledge distillation from a cross-encoder reranker and achieves state-of-the-art performance on the Japanese Massive Text Embedding Benchmark (JMTEB), surpassing multilingual models such as Multilingual E5.

To bridge the gap between bi- and cross-encoder paradigms, ColBERT [12] introduces a “late interaction” mechanism that encodes queries and documents into separate token-level embeddings and compares them using a MaxSim operator. This multi-vector design enables efficient token-level matching while allowing pre-computation of document representations, achieving a favorable

balance between retrieval accuracy and scalability. For Japanese text retrieval, JaColBERTv2.5 [5] is a state-of-the-art multi-vector retriever based on the ColBERT architecture. It employs late interaction via token-level MaxSim and is trained with knowledge distillation from strong cross-encoder teachers. Despite being trained on only 40% of the data used by its predecessor, it achieves superior performance across standard Japanese retrieval benchmarks, including MIRACL and JQaRA, while maintaining efficiency suitable for large-scale retrieval.

Building on this background, we propose the Japanese Legal Graph Retriever (JLGR), a statutory article retrieval system submitted to Task 3 of the COLIEE 2025 competition. JLGR explicitly models the hierarchical structure and inter-article references within Japanese statutes using a graph neural network (GNN). It integrates structural context into article embeddings to improve semantic matching. The system employs a two-stage retrieval pipeline, in which a bi-encoder enhanced with GNN-derived representations is used for efficient initial retrieval, followed by a cross-encoder that re-ranks the candidates to refine semantic relevance and improve overall ranking quality. We evaluate JLGR on the COLIEE 2025 Task 3 formal run dataset using the official F_2 score as the main evaluation metric. Through comparison with baseline methods, we demonstrate the effectiveness of incorporating structural information into statutory article retrieval.

2 Related Works

Structure-Aware Retrieval with GNNs. G-DSR [17] is a graph augmented dense retriever for statutory article retrieval (SAR) that explicitly models the hierarchical organization of legislative texts. The motivation stems from the observation that legal articles are rarely self-contained; their interpretation often relies on structural context such as surrounding articles and section headings. G-DSR enhances standard dense retrieval models by incorporating this structural information into article embeddings via a graph neural network (GNN). The model architecture consists of two independently trained components: (1) a dense statute retriever (DSR), and (2) a legislative graph encoder (LGE). The DSR is a bi-encoder model where both queries and statutory articles are embedded into a shared semantic space. Due to article length exceeding typical Transformer limits, articles are split into passages and processed via a hierarchical encoder: each passage is encoded with a BERT-based model, followed by a lightweight Transformer to integrate inter-passage dependencies. The final article embedding is obtained by pooling over contextualized passage representations. Training of DSR follows a contrastive learning framework that maximizes similarity between positive query-article pairs while minimizing it against sampled negatives. Negatives include both in-batch and BM25-retrieved distractors. A domain-adaptive pretraining step is also introduced, where the BERT encoder is further trained on unlabeled statutory texts from the SAR domain. The legislative graph encoder (LGE) enriches article embeddings using a GNN applied to a graph constructed from the hierarchical structure of law. Nodes represent both section headings and articles, and edges reflect parent-child relationships within the statute. Initial node features are computed using the article encoder, and the final node embeddings are learned via a 3-layer GATv2 network [4]. Subgraphs centered around training batch nodes are dynamically sampled to reduce computational overhead. Experiments on the BSARD dataset[2] show that G-DSR achieves state-of-the-art performance and that structural modeling via GNN contributes significantly to retrieval effectiveness. Our work builds on this idea and further incorporates inter-article references not modeled in G-DSR.

Dense Retrieval. Multilingual E5(mE5) [31] is a family of multilingual embedding models trained under a two-stage contrastive learning framework, designed to produce general-purpose text embeddings for tasks such as retrieval, classification, and clustering. Among its variants, we utilize `e5-multilingual-base` in this work as our first-stage retriever, as it offers a well-balanced trade-off between performance and efficiency, supports Japanese, and was publicly available before the COLIEE 2025 cutoff.

In the first stage of training, mE5 undergoes weakly supervised contrastive pre-training on approximately 1 billion multilingual text pairs, constructed from a wide range of sources. The mixture includes Wikipedia (150M), mC4 (160M) [32], CC-News (160M)[3], NLLB translations (160M) [19], Reddit comment-response pairs (160M)[4], S2ORC citation links (50M) [16], StackExchange QA pairs (50M)[5], and others. These pairs are used to optimize an InfoNCE loss [27] with large in-batch negatives (batch size 32k), encouraging semantically similar sentences across languages to be embedded closely in the vector space. In the second stage, mE5 is supervisedly fine-tuned on around 1.6 million labeled query-document pairs drawn from retrieval and QA benchmarks such as MS-MARCO [3], SQuAD [11], and others. Fine-tuning includes mined hard negatives and knowledge distillation from a cross-encoder teacher model. A particularly noteworthy variant, `mE5-large-instruct`, is trained on an extended dataset of 500k synthetic examples generated by GPT-3.5/4 [21, 22]. These examples include natural language instructions describing the task, enabling better generalization in zero-shot and multilingual settings. mE5 models outperform prior multilingual baselines on benchmarks such as MTEB [18], MIRACL [33], and BUCC [34]. In this work, we adopt `mE5-base` to encode legal questions and statutory articles into a shared embedding space, leveraging both multilingual training and instruction tuning.

Japanese-Specific Reranking. To enhance ranking precision beyond the bi-encoder stage, cross-encoder architectures have been widely adopted for reranking due to their ability to model rich interactions between query and document. For Japanese retrieval scenarios, recent work has explored language-specific cross-encoders, such as `japanese-reranker-cross-encoder-large-v1`[6].
This model is based on a 24-layer multilingual BERT encoder with a hidden size of 1024 and is fine-tuned for (query, document) relevance scoring using a full cross-encoding architecture. Unlike prior Japanese rerankers such as JaColBERTv2.5 [5], which adopt late interaction mechanisms, this model performs dense cross-attention over the concatenated input pair to directly compute relevance scores. Training data spans a variety of domains including legal, encyclopedic, and web search contexts, drawn from Japanese QA and

[2] https://huggingface.co/datasets/maastrichtlawtech/bsard
[3] https://commoncrawl.org/blog/news-dataset-available
[4] https://www.reddit.com/
[5] https://stackexchange.com/
[6] https://huggingface.co/hotchpotch/japanese-reranker-cross-encoder-large-v1

retrieval datasets such as JQaRA, JSQuAD, JaQuAD, Mr.TyDi, MIRACL [33], and quiz-style datasets. Positive and hard negative pairs were mined using a combination of BM25 and multilingual embedding models (e.g., mE5-large), selecting semantically similar yet factually incorrect passages via reciprocal rank fusion. The model is optimized with a cross-entropy loss that encourages the correct (query, document) pair to score higher than a set of hard negatives (up to 63 per query). It was trained in two stages: initial training on noisy synthetic and mined positives, followed by fine-tuning on higher-quality curated datasets. This reranker achieves strong performance on Japanese benchmarks such as JQaRA (nDCG@10: 0.7099) and JaCWIR (MAP@10: 0.9364), demonstrating its effectiveness in both open-domain and legal retrieval scenarios.

3 Task Description

3.1 Statute Law Retrieval Task

Task 3 of the COLIEE 2025 competition is a statute law information retrieval task. The objective is to retrieve an appropriate subset of articles from the Japanese Civil Code to support an entailment judgment regarding a given legal query.

Let $Q = \{q_i\}_{i=1}^{N}$ denote the set of legal queries derived from bar exam questions, and let $\mathcal{A} = \{a_j\}_{j=1}^{M}$ be the full set of statutory articles. For each query $q \in Q$, let $\mathcal{A}_q^+ \subset \mathcal{A}$ denote the gold-standard subset of relevant articles such that:

$$\text{Entails}(\mathcal{A}_q^+, q) \quad \text{or} \quad \text{Entails}(\mathcal{A}_q^+, \text{not } q)$$

In other words, the following types of articles are considered relevant:

- Articles that independently entail a Yes/No judgment.
- Articles that contribute jointly with others to such a judgment.
- Articles that appear in at least one subset whose combined meaning entails the query or its negation.

3.2 Evaluation Metrics

The COLIEE formal run evaluation adopts a cross-year setting, where past Japanese legal examination problems are used as training data and the most recent year's problems serve as the test set.

For each query $q \in Q$, we define:

- $\mathcal{R}_q \subset \mathcal{A}$: the set of articles retrieved by the system;
- rank_q: the position of the first relevant article (i.e., any $a \in \mathcal{A}_q^+$) in the ranked list of $\mathcal{R}_q$.

Each metric is computed per query and then macro-averaged across Q.

- **Precision**:
$$\text{Precision}(q) = \frac{|\mathcal{R}_q \cap \mathcal{A}_q^+|}{|\mathcal{R}_q|}$$
This measures the proportion of retrieved articles that are relevant.
- **Recall**:
$$\text{Recall}(q) = \frac{|\mathcal{R}_q \cap \mathcal{A}_q^+|}{|\mathcal{A}_q^+|}$$
This measures the proportion of relevant articles that have been successfully retrieved.
- **F_2 Score**:
$$F_2(q) = \frac{5 \cdot \text{Precision}(q) \cdot \text{Recall}(q)}{4 \cdot \text{Precision}(q) + \text{Recall}(q)}$$
The F_2 score weights recall more heavily than precision, reflecting the importance of retrieving all relevant articles.
- **Mean Reciprocal Rank (MRR)**:
$$\text{MRR} = \frac{1}{|Q|} \sum_{q \in Q} \frac{1}{\text{rank}_q}$$
This metric evaluates the rank position of the first correct result in the system's ranked output.

In practice, all of the above metrics can also be evaluated at a fixed cutoff k, such as Precision@k, Recall@k, F_2@k, or MRR@k, based on the top-k ranked articles. This reflects a realistic setting in which only the top few retrieved results are considered for downstream reasoning.

Top-k retrieval is performed using a bi-encoder with a vector-based similarity measure, such as cosine or dot product. The retrieved candidates are then reranked or filtered using a cross-encoder for more accurate scoring. Additional metrics such as Mean Average Precision (MAP) and R-precision may also be used for further analysis, although they are not officially considered in the formal evaluation of Task 3.

4 Method

This section presents the architecture and training procedure of the Japanese Legal Graph Retriever (JLGR), a two-stage retrieval framework designed to capture the hierarchical and referential structure of Japanese statutory law. JLGR models legal articles within a multi-level legislative hierarchy using a graph neural network (GNN), enabling structural information to be integrated into article representations. In the first stage, a bi-encoder is trained via contrastive learning to generate initial article embeddings, which are then refined through GNN-based propagation for structure-aware retrieval. The second stage re-ranks the top-k_{bi} candidates using a cross-encoder to produce the final top-k_{cross} predictions. An overview of the architecture is shown in Figure 5.

4.1 Dataset Construction

As defined in Section 3, the task requires retrieving a subset of relevant articles $\mathcal{A}_q^+ \subset \mathcal{A}$ for each query $q \in Q$. To this end, we construct query–article pairs for supervised training using both positive and negative labels:

- **Positive samples:** Articles $a^+ \in \mathcal{A}_q^+$ that are labeled as relevant to the query q
- **Negative samples:** Articles $a^- \in \mathcal{A}_q^- \subset \mathcal{A} \setminus \mathcal{A}_q^+$, sampled using:
  - *BM25-based negatives:* top-ranked articles by lexical similarity (i.e., hard negatives), excluded from the ground truth
  - *In-batch negatives:* positive articles from other queries within the same training batch

Here, $\mathcal{A}_q^-$ denotes a subset of non-relevant articles for query q, constructed as a mixture of BM25-based hard negatives and in-batch negatives.

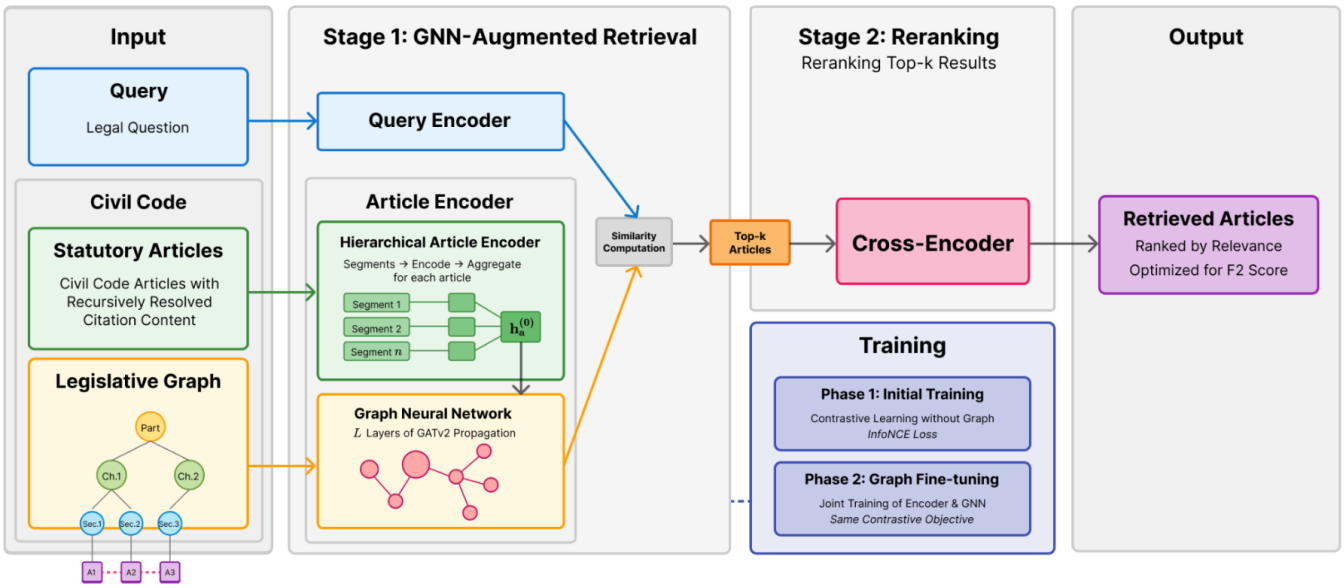


Figure 5: Overall architecture of the Japanese Legal Graph Retriever (JLGR). The system follows a two-stage retrieval framework. Stage 1: GNN-Augmented Retrieval uses a bi-encoder enhanced with graph-based propagation over a legislative graph to independently encode queries and statutory articles. Each article is recursively inlined with citation content and hierarchically encoded to obtain initial embeddings, which are then further refined via GNN-based propagation. Top-$k_{\mathbf{bi}}$ candidates are retrieved based on cosine similarity. Stage 2: Reranking applies a cross-encoder to re-rank the retrieved candidates, and the top-ranked article(s) exceeding a threshold are selected as final predictions, optimized for the $\mathbf{F}_2$ score.

4.2 Graph Construction

To incorporate the hierarchical structure of the Civil Code, we construct a directed graph

$$G = (V, E)$$

where:

- V is the set of nodes representing both statutory articles and structural units:

  $$V = \mathcal{A} \cup \mathcal{H}$$

  Here, $\mathcal{A}$ denotes the set of statutory articles, and $\mathcal{H}$ is the set of structural units defined in the Civil Code, such as:

  $$\mathcal{H} = \left\{ \begin{matrix} \texttt{part},\ \texttt{chapter},\ \texttt{section}, \\ \texttt{subsection},\ \texttt{division},\ \texttt{caption} \end{matrix} \right\}$$

  Each $h_i \in \mathcal{H}$ represents a heading in the statutory hierarchy.
- $E \subseteq V \times V$ is the set of directed edges representing hierarchical relations. Edges are defined based on structural containment and parent–child relationships among nodes. Specifically:
  - For structural nodes $h_i, h_j \in \mathcal{H}$:

    $(h_i, h_j) \in E$ if h_i is the immediate parent of h_j

  - For statutory article nodes $a \in \mathcal{A}$ and their enclosing structural unit $h \in \mathcal{H}$:

    $(h, a) \in E$ if statutory article a is contained within structural unit h

The graph G is constructed by parsing structural metadata associated with each statutory article and instantiating edges that reflect the Civil Code's hierarchical organization.

In this study, we do not explicitly represent citation links as edges in the graph structure. While our framework is in principle capable of modeling such references as directed edges between statutory article nodes, we refrain from incorporating them at this stage due to the lack of a reliable estimation of their structural and computational impact. In particular, it remains an open question whether the citation graph would introduce cycles or significantly increase the graph's connectivity and propagation complexity. Instead, we choose a simpler yet effective strategy: integrating citation information directly into the input text of each article node. For each article a_i, we append the textual content and structural headings of its cited articles $\{a_j\}$, recursively resolving references via topological sorting of the citation graph. This ensures that multi-level references (e.g., "Article 25" referencing "Article 24", which in turn references "Article 23") are consistently inlined in a contextually coherent order.

Each node $v \in V$ is initialized with an embedding vector $\mathbf{h}_v^{(0)} \in \mathbb{R}^d$. For article nodes, embeddings are obtained using the article encoder trained via contrastive learning in Phase 1 (see Section 4.5). For structural nodes, we apply the same encoder to their associated textual labels (e.g., section headings).

4.3 Graph-Augmented Bi-Encoder

JLGR employs a bi-encoder architecture, in which queries and articles are encoded independently into a shared semantic space. The

article encoder is further enhanced with graph-based propagation to incorporate structural context from the legislative graph G, while the query encoder operates independently of graph structure.

Query encoding. Each query $q \in Q$ is mapped to an embedding vector by a transformer encoder:

$$\mathbf{q} = f_{\text{query}}(q) \in \mathbb{R}^d$$

Article encoding. Each article $a \in \mathcal{A}$ is first encoded by a transformer:

$$\mathbf{h}_a^{(0)} = f_{\text{article}}(a) \in \mathbb{R}^d$$

We adopt a hierarchical article encoder to handle long statutory articles that exceed the input limit of base transformer models. Each article is segmented into textual units (e.g., sentences or clauses), which are independently encoded and then aggregated using a transformer-based model over segment-level embeddings. This enables the encoder to capture both intra-segment semantics and higher-level structural coherence.

We apply L layers of graph attention network (GATv2) propagation:

$$\mathbf{h}_v^{(l+1)} = \sigma\left(\sum_{u \in \mathcal{N}(v)} \alpha_{uv}^{(l)} \mathbf{W}^{(l)} \mathbf{h}_u^{(l)}\right)$$

where $\alpha_{uv}^{(l)}$ is a learned attention coefficient, $\mathbf{W}^{(l)}$ is a trainable weight matrix, and σ is a non-linear activation function (e.g., ReLU).

The final article embedding is:

$$\mathbf{z}_a = \mathbf{h}_a^{(L)}$$

Similarity computation. The similarity between a query and article is measured by cosine similarity:

$$s_{\text{bi}}(q, a) = \frac{\mathbf{q}^\top \mathbf{z}_a}{\|\mathbf{q}\| \cdot \|\mathbf{z}_a\|}$$

4.4 Training Strategy

To improve training stability and enable effective integration of structural information, we adopt a two-phase training strategy:

- **Phase 1: Initial Contrastive Training.** We train the bi-encoder without graph structure using contrastive learning with InfoNCE loss (Section 4.5).
- **Phase 2: Graph-Based Fine-Tuning.** Using the article encoder from Phase 1, we initialize node embeddings and jointly train the encoder and GNN over the legislative graph using the same contrastive objective.

4.5 Contrastive Learning

The bi-encoder is trained with contrastive learning using the InfoNCE loss:

$$\mathcal{L} = -\log \frac{\exp(s_{\text{bi}}(q, a^+)/\tau)}{\exp(s_{\text{bi}}(q, a^+)/\tau) + \sum_{a^- \in \mathcal{A}_q^-} \exp(s_{\text{bi}}(q, a^-)/\tau)}$$

where τ is a temperature hyperparameter. The article encoder, including the GNN layers, is trained end-to-end as part of this objective.

After training, the bi-encoder is used to retrieve the top-k_{bi} candidates based on dense similarity scores. We fix $k_{\text{bi}} = 100$ to match the top-100 submission format used in COLIEE and to ensure that reranking with the cross-encoder remains computationally efficient.

4.6 Cross-Encoder Reranking

The top-k_{bi} articles retrieved by the bi-encoder are re-ranked using a cross-encoder:

$$s_{\text{cross}}(q, a) = \text{CrossEncoder}(q, a)$$

An article is predicted as relevant if:

$$\text{predict}(q, a) = \mathbb{I}[a \in \text{Top-}k_{\text{cross}}(q) \wedge s_{\text{cross}}(q, a) \geq \theta]$$

The threshold θ and the number of final outputs k_{cross} were selected based on validation performance. Specifically, we performed a grid search over the $(k_{\text{cross}}, \theta)$ space after re-ranking to maximize the F_2 score.

5 Experiments

5.1 Setting

We follow the official evaluation setting of COLIEE Task 3, in which training and test data are drawn from disjoint years. Specifically, we conducted two patterns of experiments, R06 (formal run) and R05 (previous year's formal run): we train on legal questions and articles from 18 yearly sets spanning from H18 to R04 (1097 questions), use R05 (109 questions) for development evaluation, and evaluate on R06 (105 questions) as the official test set for the COLIEE 2025 formal run. The R06 test set was released with gold annotations, which allows us to perform a fair local evaluation of all baseline models for comparison with our submitted system. The primary evaluation metric is the F_2 score, which emphasizes recall over precision. We also report Precision, Recall, and Mean Reciprocal Rank (MRR).

5.2 Baselines

We compare JLGR against several representative retrieval methods, including both lexical and dense approaches. All dense retrieval baselines use mE5, which refers to `intfloat/multilingual-e5-base`[7], to encode both queries and articles. We evaluate both the pretrained model and variants fine-tuned with contrastive learning, with and without graph structure.

- **BM25**: A lexical retrieval baseline implemented using Elasticsearch[8] with default parameters.
- **mE5 (pretrained)**: Bi-encoder retrieval using mE5 without any task-specific fine-tuning. This represents a strong off-the-shelf baseline.
- **mE5 (contrastive)**: mE5 fine-tuned on the COLIEE training set using contrastive learning with BM25-based and in-batch negatives. No graph structure is used.
- **JLGR (without reranking)**: The bi-encoder model trained with contrastive learning and further enhanced with GNN-based propagation over the legislative graph, but without reranking via cross-encoder. This corresponds to Phase 2 in JLGR, excluding the second-stage reranking.

[7] https://huggingface.co/intfloat/multilingual-e5-base
[8] https://www.elastic.co/elasticsearch

- **JLGR (ours) (INFA)**: A GNN-augmented bi-encoder model that incorporates hierarchical and referential structure from a legislative graph. The encoder is first trained using contrastive learning, then fine-tuned with graph-based propagation via GATv2 layers over the statutory graph. Then, the top-k_{bi} candidates retrieved by the bi-encoder are re-ranked using a cross-encoder, and the top-k_{cross} predictions above a threshold θ are selected as final outputs.
  For reranking, we use the Japanese-specific cross-encoder `japanese-reranker-cross-encoder-large-v1` introduced in Related Work.

5.3 Implementation Details

We use mE5 as our bi-encoder model for encoding both queries and statutory articles, in order to enable a fair comparison with baseline methods that also rely on the same encoder. The associated tokenizer is based on `xlm-roberta-base`[9] and uses `sentencepiece` [13]. Following G-DSR [17], we adopt a hierarchical encoding scheme, where article segments are first encoded using mE5 and then aggregated via mean pooling over segment embeddings. Here, a segment refers to a contiguous span of the article text, such as a sentence or paragraph, that does not exceed the model's maximum input length of 512 tokens. Each segment is independently passed through the same mE5 encoder, and its embedding is defined as the [CLS] token output corresponding to that segment. This hierarchical approach enables encoding of long articles that exceed the input length limitation of Transformer-based models while preserving fine-grained semantic representations. The contrastive variant is trained using the InfoNCE loss with BM25-based hard negatives and in-batch negatives. JLGR augments this model with a GATv2-based GNN applied to a graph constructed from statutory hierarchy and inter-article citations.

Graph nodes are initialized using the encoder trained in Phase 1. Hierarchical edges are derived from structual units such as part, chapter, and section to reflect the statutory structure. Citation information is incorporated by recursively concatenating the texts of referenced articles into the citing article, based on citation mentions detected using regular expressions that match both absolute references (e.g., "Article 25") and relative expressions (e.g., "the previous two articles"), as well as article ranges (e.g., "Articles 25 to 27") and multiple references (e.g., "Articles 12 and 15").

For final prediction, we fixed the number of initial candidates retrieved by the bi-encoder to $k_{\text{bi}} = 100$. We then performed grid search over the number of cross-encoder outputs $k_{\text{cross}} \in \{1, \ldots, 10\}$ and prediction threshold $\theta \in \{0.0, 0.1, \ldots, 1.0\}$ using the R05 validation set. This search indicated that selecting only the top-1 reranked article without applying a threshold ($k_{\text{cross}} = 1$, $\theta = 0.0$) yielded the highest F_2 score.

Training is performed using the AdamW optimizer with a learning rate of 2×10^{-5}, batch size of 16, and early stopping based on validation F_2 score. The temperature parameter τ of the InfoNCE loss is set to 0.1, and ReLU is used as the non-linear activation function in the GATv2 layers. Inputs are truncated to 512 tokens for articles and 256 tokens for queries. All experiments are conducted on an NVIDIA A100 GPU (40GB VRAM).

[9] https://huggingface.co/FacebookAI/xlm-roberta-base

Table 1: Evaluation results on COLIEE Task 3 (R05 test set, 2024)

Model	F_2	Precision	Recall	MRR
BM25	0.5642	0.3716	0.6606	0.6864
mE5 (pretrained)	0.4587	0.4954	0.4541	0.5173
mE5 (contrastive)	**0.6162**	**0.6881**	0.6055	**0.7769**
JLGR (w/o* reranking)	**0.6162**	0.4174	**0.7156**	0.7599
JLGR (ours) (**INFA**)	0.5944	0.6271	0.5902	0.7039

* w/o = without

Table 2: Locally evaluated results on COLIEE Task 3 (R06 test set)

Model	F_2	Precision	Recall	MRR
BM25	0.5901	0.6081	0.5878	0.7782
mE5 (pretrained)	0.5556	0.5676	0.5541	0.7214
mE5 (contrastive)	**0.6592**	0.6892	0.6554	**0.8415**
JLGR (w/o* reranking)	0.6126	0.3920	**0.7230**	0.8066
JLGR (ours) (**INFA**)	0.6474	**0.7179**	0.6389	0.8337

* w/o = without

5.4 Results

Evaluation on R05. In our method, the number of candidates retrieved by the bi-encoder was fixed at $k_{\text{bi}} = 100$. For the cross-encoder reranking stage, we tuned the prediction threshold θ and the number of final outputs k_{cross} on the R05 set. For each model, we report the performance on the R05 set using the hyperparameter configuration that yielded the highest F_2 score on that model, independently tuned per model. In contrast, JLGR used a fixed configuration of $k_{\text{cross}} = 1$ and $\theta = 0.0$ for the R06 formal run, based on the R05 validation results.

Table 1 shows that JLGR significantly outperforms all baselines on this dataset.

Local evaluation on R06. We then evaluate our model and baselines on the R06 test set, which was used in the COLIEE 2025 formal run. The baselines are re-evaluated locally using our implementation, while JLGR corresponds to the formal run submission.

Official evaluation results. We submitted JLGR as our *formal run* to the COLIEE 2025 Task 3 competition. As shown in Table 3, our team **INFA** ranked 3rd out of 8 participating teams based on the official F_2 score, demonstrating the competitiveness of our method in the blind evaluation setting.

6 Discussion

JLGR (without reranking), which applies graph-based propagation without a second-stage cross-encoder, and mE5 (contrastive), a bi-encoder model fine-tuned with contrastive learning on the COLIEE training set, achieved the same F_2 score in the R05 evaluation, though their retrieval behaviors differ notably. While mE5 (contrastive) exhibited strong early ranking performance, as evidenced by the highest MRR, JLGR (without reranking) retrieved a broader range of relevant articles, achieving the highest recall, by

Table 3: Best-performing submitted run per team on COLIEE Task 3 (2025 formal run)

Team	F_2	Precision	Recall
JNLP_RUN1	0.8365	0.8037	0.8744
CAPTAIN.H2	0.8301	0.8333	0.8516
INFA	0.6917	0.7671	0.6826
mpnetUAIIRLab	0.6674	0.3562	0.8858
OVGU3	0.6041	0.6347	0.6142
UIwa	0.5816	0.5856	0.5890
UA-gte	0.2540	0.0.0986	0.4361
NOWJ.H1	0.0137	0.0137	0.0137

leveraging the structural organization of statutes through graph-based propagation. In our ablation experiments on the R05 test set, we found that JLGR (without reranking) yields the highest recall but suffers from a significant drop in precision. Introducing a cross-encoder reranker helps recover precision by reordering top-k candidates based on fine-grained semantic matching, resulting in more balanced performance. Our current reranking module uses a general-purpose Japanese cross-encoder model, specifically `japanese-reranker`[10], which was trained on a mixture of web and QA data and has demonstrated strong performance in general Japanese QA tasks. However, it has not yet been fine-tuned on statutory article retrieval, and its effectiveness in this domain remains to be fully validated. Despite the slight drop in F_2 score compared to the version without reranking, we adopted JLGR (with reranking), which applies a cross-encoder reranking stage and was submitted to the COLIEE 2025 formal run to reduce false positives and improve the semantic quality of final outputs. Although the reranked JLGR shows slightly lower recall and precision than mE5 (contrastive), we argue that its structure-aware design and balanced performance better align with the legal reasoning required for statutory retrieval. We anticipate that in-domain fine-tuning of the reranker will further close this gap while maintaining the interpretability and extensibility benefits of the JLGR framework.

The current model retrieves articles based on semantic embeddings and does not fully address the lexical or conceptual gap that often exists between user queries and statutory expressions. For example, a user query might use colloquial terms such as "apartment", while legal statutes may describe the same concept using formal terminology such as "leased property". JLGR partially mitigates this mismatch through structural propagation via section headers and captions, but surface-level variations are not always captured. To address this, combining JLGR with traditional retrieval models that support lexical signals such as keyword overlap or synonym expansion may be effective. For instance, hybrid systems that incorporate BM25-style scoring could complement the dense model by recovering articles matched through surface-level cues. Building on this, large language models (LLMs) may further enhance such hybrid systems by enabling more flexible, context-aware query expansion. For example, LLMs can dynamically rephrase user queries or generate paraphrases that bridge the abstraction gap between user intent and statutory language.

Another avenue for improvement lies in the representation of referential information. In the current system, references to other articles are incorporated into the citing article's text via recursive inlining, and only the citing article is represented as a node in the graph. However, this approach flattens the structural distinction between citing and cited content, potentially limiting the granularity of information propagation. In future work, we plan to model referenced articles as independent nodes in the legislative graph and connect them explicitly to the citing articles. This will allow the GNN to learn from both hierarchical and referential structures in a unified way, and enable finer-grained control over how inter-article dependencies influence article representations.

A related design question concerns the role of caption nodes in the legislative hierarchy. In JLGR, captions, which are textual headings placed directly above individual articles, are treated as structural units and included as nodes in the graph. This decision is motivated by the observation that captions often convey topical or contextual cues that are directly relevant to interpreting the articles beneath them, especially in statutory texts where individual provisions may be terse or highly abstract. One alternative design would be to concatenate caption text directly into the article content during input preprocessing, thereby allowing transformer encoders to access that information locally. However, such treatment flattens the structural distinction between the article and its contextual heading, and makes it difficult for the model to generalize structural patterns or aggregate signals across articles sharing the same caption. By contrast, representing captions as separate nodes preserves their identity and enables GNN-based propagation of context in a structured, compositional manner. This allows caption information to be shared across multiple connected articles and contributes to the formation of topic-aware clusters in the graph. Nonetheless, caption nodes introduce additional hops between articles, which may dilute signal propagation or complicate attention-based aggregation. Future work should empirically evaluate the trade-offs between textual inlining and explicit structural modeling, for example, by comparing retrieval performance under different configurations or selectively weighting caption-related edges.

Finally, while this study focused on Japanese civil law, the proposed approach is general and could be extended to other statutory systems with similar hierarchical and referential structure.

7 Conclusion

We presented JLGR, a graph-enhanced bi-encoder model for statutory article retrieval. JLGR encodes the hierarchical structure of legal documents as a graph and incorporates inter-article references by recursively concatenating cited texts into the input text of each article. Graph neural networks are applied to enrich article embeddings with structural context derived from statutory organization. Experiments on COLIEE Task 3 show that JLGR (with reranking) combines a structure-aware retriever based on GNNs, capable of high recall, with a cross-encoder reranker to offer a balanced alternative to existing lexical and dense retrieval models. Specifically, the submitted JLGR system achieved an F_2 score of 0.6474 on the

[10] This refers to the model `hotchpotch/japanese-reranker-cross-encoder-large-v1`.

R06 test set in the formal run, and 0.5944 on the R05 test set used for development evaluation. While its precision and recall on R05 were slightly lower than those of a contrastively fine-tuned bi-encoder baseline based on mE5, JLGR provides a more interpretable and recall-oriented retrieval strategy grounded in statutory structure. These results highlight the potential of structure-aware modeling for improving retrieval in legal and other structurally organized text domains. Future work includes in-domain fine-tuning of the reranker, incorporation of LLM-based components such as query rewriting and hybrid scoring, and application to other hierarchically structured document collections.

References

[1] 2020. Retrieve & Re-Rank — Sentence Transformers documentation. https://sbert.net/examples/sentence_transformer/applications/retrieve_rerank/README.html.

[2] Lucas Albarede, Philippe Mulhem, Lorraine Goeuriot, Sylvain MariÃ©, Claude Le Pape-Gardeux, and Trinidad Chardin-Segui. 2023. Heterogeneous graph attention networks for passage retrieval. *Inf. Retr. Boston.* 26, 1-2 (Dec. 2023), 1–25. https://link.springer.com/article/10.1007/s10791-023-09424-3

[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. 2016. MS MARCO: A human generated MAchine Reading COmprehension dataset. *arXiv [cs.CL]* (Nov. 2016). http://arxiv.org/abs/1611.09268

[4] Shaked Brody, Uri Alon, and Eran Yahav. 2021. How Attentive are Graph Attention Networks? *arXiv [cs.LG]* (May 2021). http://arxiv.org/abs/2105.14491

[5] Benjamin ClaviÃ©. 2024. JaColBERTv2.5: Optimising multi-vector retrievers to create state-of-the-art Japanese retrievers with constrained resources. *arXiv [cs.IR]* (July 2024). http://arxiv.org/abs/2407.20750

[6] 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)*. Association for Computational Linguistics, 4171–4186.

[7] 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 *Lecture Notes in Computer Science*. Springer Nature Singapore, Singapore, 109–124. https://coliee.org/documents/waivers/overview_COLIEE2024.pdf

[8] Jiafeng Guo, Yixing Fan, Qingyao Ai, and W Bruce Croft. 2016. A deep relevance matching model for ad-hoc retrieval. In *Proceedings of the 25th ACM international conference on information and knowledge management*. 55–64.

[9] Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry Heck. 2013. Learning deep structured semantic models for web search using clickthrough data. In *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*. 2333–2338.

[10] 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)*. Association for Computational Linguistics, 6769–6781. https://doi.org/10.18653/v1/2020.emnlp-main.550

[11] 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)*. Association for Computational Linguistics, Stroudsburg, PA, USA, 6769–6781. https://aclanthology.org/2020.emnlp-main.550.pdf

[12] 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.

[13] Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for Neural Text Processing. *arXiv [cs.CL]* (Aug. 2018). http://arxiv.org/abs/1808.06226

[14] Jimmy Lin, Rodrigo Nogueira, and Andrew Yates. 2021. Pretrained transformers for text ranking: BERT and beyond. *Synthesis Lectures on Human Language Technologies* 14, 4 (2021), 1–325.

[15] Ye Liu, Kazuma Hashimoto, Yingbo Zhou, Semih Yavuz, Caiming Xiong, and Philip S Yu. 2021. Dense Hierarchical Retrieval for open-domain Question Answering. *arXiv [cs.IR]* (Oct. 2021). http://arxiv.org/abs/2110.15439

[16] Kyle Lo, Lucy Lu Wang, Mark Neumann, Rodney Kinney, and Daniel Weld. 2020. S2ORC: The semantic scholar open research corpus. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Stroudsburg, PA, USA, 4969–4983. https://aclanthology.org/2020.acl-main.447.pdf

[17] Antoine Louis, Gijs van Dijck, and Gerasimos Spanakis. 2023. Finding the law: Enhancing statutory article retrieval via graph neural networks. *arXiv [cs.IR]* (Jan. 2023). http://arxiv.org/abs/2301.12847

[18] Niklas Muennighoff, Nouamane Tazi, Loic Magne, and Nils Reimers. 2023. MTEB: Massive Text Embedding Benchmark. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, Stroudsburg, PA, USA, 2014–2037. https://aclanthology.org/2023.eacl-main.148.pdf

[19] NLLB Team, Marta R Costa-jussÃ , James Cross, Onur ÃElebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, Anna Sun, Skyler Wang, Guillaume Wenzek, Al Youngblood, Bapi Akula, Loic Barrault, Gabriel Mejia Gonzalez, Prangthip Hansanti, John Hoffman, Semarley Jarrett, Kaushik Ram Sadagopan, Dirk Rowe, Shannon Spruit, Chau Tran, Pierre Andrews, Necip Fazil Ayan, Shruti Bhosale, Sergey Edunov, Angela Fan, Cynthia Gao, Vedanuj Goswami, Francisco GuzmÃ¡n, Philipp Koehn, Alexandre Mourachko, Christophe Ropers, Safiyyah Saleem, Holger Schwenk, and Jeff Wang. 2022. No Language Left Behind: Scaling human-centered machine translation. *arXiv [cs.CL]* (July 2022). http://arxiv.org/abs/2207.04672

[20] Rodrigo Nogueira and Kyunghyun Cho. 2019. Passage Re-ranking with BERT. *arXiv preprint arXiv:1901.04085* (2019).

[21] OpenAI. 2023. GPT-4 Technical Report. *arXiv [cs.CL]* (March 2023). http://arxiv.org/abs/2303.08774

[22] Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. *arXiv [cs.CL]* (March 2022). http://arxiv.org/abs/2203.02155

[23] 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.

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

[25] Gerard Salton and Christopher Buckley. 1988. Term-weighting approaches in automatic text retrieval. *Information processing & management* 24, 5 (1988), 513–523.

[26] Hayato Tsukagoshi and Ryohei Sasano. 2024. Ruri: Japanese General Text Embeddings. *arXiv [cs.CL]* (Sept. 2024). http://arxiv.org/abs/2409.07737

[27] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with Contrastive Predictive Coding. *arXiv [cs.LG]* (July 2018). http://arxiv.org/abs/1807.03748

[28] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *arXiv [cs.CL]* (June 2017). http://arxiv.org/abs/1706.03762

[29] Kunze Wang, Yihao Ding, and Soyeon Caren Han. 2024. Graph neural networks for text classification: a survey. *Artif. Intell. Rev.* 57, 8 (July 2024), 1–38. https://link.springer.com/article/10.1007/s10462-024-10808-0

[30] Kexin Wang, Nils Reimers, and Iryna Gurevych. 2023. DAPR: A benchmark on document-Aware Passage Retrieval. *arXiv [cs.IR]* (May 2023). https://aclanthology.org/2024.acl-long.236.pdf

[31] Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. 2024. Multilingual E5 Text Embeddings: A Technical Report. *arXiv [cs.CL]* (Feb. 2024). http://arxiv.org/abs/2402.05672

[32] Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2020. mT5: A massively multilingual pre-trained text-to-text transformer. *arXiv [cs.CL]* (Oct. 2020). http://arxiv.org/abs/2010.11934

[33] Xinyu Zhang, Nandan Thakur, Odunayo Ogundepo, Ehsan Kamalloo, David Alfonso-Hermelo, Xiaoguang Li, Qun Liu, Mehdi Rezagholizadeh, and Jimmy Lin. 2023. MIRACL: A multilingual retrieval dataset covering 18 diverse languages. *Trans. Assoc. Comput. Linguist.* 11 (Sept. 2023), 1114–1131. https://aclanthology.org/2023.tacl-1.63.pdf

[34] Pierre Zweigenbaum, Serge Sharoff, and Reinhard Rapp. 2017. Overview of the Second BUCC Shared Task: Spotting Parallel Sentences in Comparable Corpora. In *Proceedings of the 10th Workshop on Building and Using Comparable Corpora*. 60–67. https://aclanthology.org/W17-2512.pdf

CAPTAIN at COLIEE 2025: Enhancing Legal Text Processing and Structural Analysis with Large Language Models

Dat Nguyen*
nguyendt@jaist.ac.jp
Japan Advanced Institute of Science and Technology
Nomi, Ishikawa, Japan

Minh-Phuong Nguyen*
phuongnm@jaist.ac.jp
Japan Advanced Institute of Science and Technology
Nomi, Ishikawa, Japan

Quang-Huy Chu*
cqhofsns@jaist.ac.jp
Japan Advanced Institute of Science and Technology
Nomi, Ishikawa, Japan

Son T. Luu*
sonlt@jaist.ac.jp
Japan Advanced Institute of Science and Technology
Nomi, Ishikawa, Japan

Nguyen-Hoang Chu*
nguyenhoangchu@jaist.ac.jp
Japan Advanced Institute of Science and Technology
Nomi, Ishikawa, Japan

Trung Vo*
trungvo@jaist.ac.jp
Japan Advanced Institute of Science and Technology
Nomi, Ishikawa, Japan

Le-Minh Nguyen
nguyenml@jaist.ac.jp
Japan Advanced Institute of Science and Technology
Nomi, Ishikawa, Japan

Abstract

The legal domain poses unique challenges for information extraction and reasoning due to the intricate structure and domain-specific language of legal texts. To address these challenges, our team, CAPTAIN, leverages recent advances in Large Language Models (LLMs) to enhance legal information processing within the scope of the COLIEE 2025 competition. We participate in four tasks: Legal Case Entailment (Task 2), Statute Law Retrieval (Task 3), Legal Textual Entailment (Task 4), and Legal Judgment Prediction for Japanese Tort Law (Pilot Task). Our approach harnesses the interpretive power of LLMs to analyze and summarize complex legal documents, uncover semantic relationships between legal cases and relevant statutes, and perform contextual reasoning. By leveraging diverse prompting techniques, our approach effectively uncovers implicit relationships between legal cases and their corresponding statutes, thereby enhancing both interpretability and accuracy. Experimental results demonstrate the strength of our method: it achieved first place in the Tort Prediction sub-task of the Pilot Task, and second place in both the Legal Statute Law Retrieval and Rationale Extraction sub-tasks, confirming the potential of LLM-based approaches in legal AI.

CCS Concepts

• **Computing methodologies** → **Neural networks**; • **Applied computing** → *Law*.

*These authors contributed equally

COLIEE 2025, Chicago, USA

Keywords

COLIEE competition, Legal information processing, Legal text retrieval, Reasoning prompting, Large language models

ACM Reference Format:
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, June 20, 2025, Chicago, USA.* ACM, New York, NY, USA, 10 pages.

1 Introduction

The rapid advancement of artificial intelligence (AI) has opened new frontiers in the legal domain, where the intricate nature of legal data—marked by its complexity, formality, and specialized terminology—poses significant challenges for automated systems. Efforts to bridge this gap have given rise to initiatives like the Competition on Legal Information Extraction and Entailment (COLIEE) competition, an annual event designed to push the boundaries of AI in processing legal documents. By focusing on both case law and statute law, COLIEE encompasses a range of tasks that test the capabilities of machine learning and natural language processing techniques in two core areas: retrieval and entailment. These tasks simulate real-world legal practices, such as identifying precedent cases or statutory provisions and determining their relevance or logical implications for specific legal queries. As AI continues to evolve, competitions like COLIEE highlight the potential for technology to assist legal professionals and underscore the need for systems that can navigate the nuanced reasoning and domain-specific knowledge inherent in the practice of law. COLIEE has become a prominent benchmark for driving progress in legal information processing and retrieval. It features a series of tasks organized into two main categories: legal document retrieval and legal entailment. In detail, Task 1 (Legal Case Retrieval) focuses on a fundamental

aspect of legal work: finding precedent cases that support a given legal argument. This task plays a critical role not only in helping lawyers build persuasive arguments but also in guiding judicial decision-making. Task 2 (Legal Case Entailment) also deals with case law but requires models to identify specific paragraphs that support the outcome of a case, placing greater emphasis on legal reasoning and interpretive skills. Task 3 (Statute Law Retrieval) and Task 4 (Legal Textual Entailment) extend these challenges to the domain of statute law, which involves written laws and codes. Mirroring the structure of Tasks 1 and 2, Task 3 aims to retrieve relevant statutory provisions, while Task 4 tests a model's ability to determine whether a legal statement is entailed or contradicted by those provisions. While Tasks 1 and 3 focus on retrieval and can serve as groundwork for Tasks 2 and 4, they are not strict prerequisites. Each task can be tackled independently, reflecting the complementary yet distinct nature of legal retrieval and entailment.

In 2025 the Pilot Task (LJPJT25) is introduced by COLIEE organizers. This task focuses on civil case judgments related to torts, specifically under Article 709 of the Japanese Civil Code. According to Japanese law, a tort refers to an intentional or negligent violation of someone's rights or legal interests that results in harm or loss to the plaintiff. In today's digital society, torts are increasingly relevant in online disputes, such as defamation or privacy violations on social media platforms. This task offers a controlled environment for researchers to explore techniques using authentic judicial case data from Japanese courts.

We approach the four tasks by leveraging large language models (LLMs) in distinct ways. For Task 2, we employ robust retrieval models, including MonoT5 [19] and BGE [4], experimenting with prompting techniques under both settings instruction fine-tuning and in-context learning LLMs to refine the candidate in the retrieval stage. In Task 3, we propose a three-stage pipeline that leverages advanced zero-shot retrieval, reranking and selection techniques using large language models (LLMs). Task 4's methodology focuses on leveraging LLMs to understand causal relationships in law articles for synthetic data generation. The goal is to enhance both the size and quality of the dataset, thereby improving entailment fine-tuning. Lastly, for the Pilot Task, we propose a method that leverages a fine-tuned large language model (LLM) to perform both Tort Prediction (TP) and Rationale Extraction (RE).

The remainder of the paper is organized as follows: Section 2 provides the background for four tasks. Section 3 details the technical methodology, followed by Section 4, which presents the experimental results. Lastly, Section 5 concludes the paper with a summary of key findings and a discussion of potential future directions.

2 Related works

Task 2. In the 2021 COLIEE competition, the winning team NM [21] explored models such as DeBERTa, MonoT5, and MonoT5 in a zero-shot setting, and submitted an ensemble of MonoT5 and DeBERTa. Meanwhile, the UA team [11] fine-tuned a BERT model using the official training set. In 2022, NM secured first place again by combining outputs from a fine-tuned MonoT5 and a zero-shot MonoT5 approach [20]. The JNLP team [1] placed second with a multi-faceted system involving three runs: score fusion from LegalBERT and BM25, Abstract Meaning Representation (AMR) to match key terms between queries and paragraphs, and an interaction-based method that integrates top candidates from LegalBERT and AMR + BM25. In 2023, the CAPTAIN team [16] achieved state-of-the-art results by fine-tuning MonoT5 using hard negative sampling and applying ensemble techniques. The THUIR team [13] 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 [18] proposed two approaches: (1) fine-tuning a LegalBERT [3] model using triplet loss [22] on the official training data for task 2, and (2) fine-tuning a MonoT5 [19], pre-trained on MSMARCO [17], utilizing hard negative samples, selected via BM25 and an alternative MonoT5 variant.

Task 3. This task involves ranking articles from the Japanese Civil Code and selecting the most relevant statutes for a given legal case. Cases generally fall into two categories: those querying causes/results, or those seeking penalty frames. A variety of methods have been applied in recent years. In 2020, LLNTU [23] led with an ensemble of BERT models. The 2021 winner, OvGU [25], applied several BERT variants and introduced Sentence-BERT-based data augmentation. In 2022, HUKB [29] won by augmenting the statute corpus with modified articles and judicial interpretations to better align with query cases. In 2023, the CAPTAIN team [16] achieved first place by ensembling BERT Japanese and MonoT5, enhanced by diverse data filtering strategies. In COLIEE 2024, the AHMR team [18] proposed approach initially applies a fine-tuned MonoT5 [19] model to rank candidate articles by relevance. This ranked list is then refined using a large language model (LLM), which applies additional post-processing to finalize the selected articles. Moreover, the CAPTAIN team [16] fine-tuned a MonoT5 model for pairwise classification between a legal decision and candidate paragraphs, using hard negative mining to improve the distinction between relevant and irrelevant content. To further boost performance, they apply zero-shot and few-shot prompting with FLAN-T5 on top-ranked candidates, capturing relationships not only between the decision and individual paragraphs but also among candidates themselves.

Task 4. In COLIEE 2021, HUKB [28] used a combination of BERT-based models with data augmentation, focusing on extracting judicial decision sentences and generating labeled training data. OvGU [25] addressed the task using a graph neural network where nodes represented articles or queries, and embeddings were derived from pre-trained BERT models. In 2022, JNLP [1] compared the effectiveness of ELECTRA, RoBERTa, and LegalBERT, and investigated negation-based data augmentation. LLNTU [15] reformulated the data into disjunctive union strings and introduced two models based on longest uncommon subsequence similarity: one excluding stopwords (LLNTUdiffSim) and another including them (LLNTU-deNgram). In 2023, JNLP [2] adopted a zero-shot approach using LLMs like Flan-T5 and Alpaca-T5, generating answers via prompt-based input that includes both queries and relevant articles. This marks a shift from binary classification to generative entailment prediction, capturing more nuanced positive or negative meanings. The authors explored various prompting strategies with Google's Flan-T5-XXL model, selecting top-performing prompts based on

validation data using a ranking-based voting strategy. They also experimented with different thresholds to handle low-agreement situations among prompts. In COLIEE 2024, the CAPTAIN team [16] proposed three strategies to enhance model performance: (1) Few-shot prompting by including three similar training examples in the prompt to guide in-context learning; (2) Automatic Chain-of-Thought (Auto-CoT) prompting, where reasoning chains are generated and retrieved using Dense Passage Retrieval, then used to construct prompts for Flan-T5-XXL; and (3) Data augmentation through LLM-generated summaries and synthetic hypotheses to expand the training set with both positive and negative examples.

Pilot Task. 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 [12]. These earlier methods often required substantial manual effort and struggled to generalize effectively [10]. 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 [27]. 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 Methodology

3.1 Task 2: Legal Case Entailment

The Legal Case Entailment task aims to predict the relevant paragraphs in the base case based on a given text fragment as the query. For each sample in the dataset, let the given text fragment as query q and a list of paragraphs $P = [p1, p2, ..., p_k]$, the aims is to determine the entailment of q on each paragraph p_i s.t. $i = 1...k$. The entailment of the query q and each p_i is determined by two values including *entailment* and *not entailment*. Inherited from the previous year's SOTAs proposed by the CAPTAIN [16] and AMHR [18] for the Legal Case Entailment task, we employed a system for this year's competition, including two main stages as shown in Figure 1:

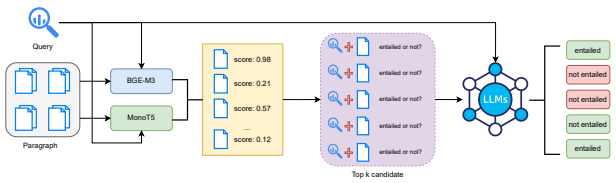


Figure 1: Overview of our proposed system for legal case entailment.

- **Stage 1: Candidate retrieval**: We fine-tuned two robust model for retrieval task including the MonoT5 [19] and BGE [4]. We first construct a negative sample set by choosing the top 10 reverse candidates computed by the BM25 score. Then, we fine-tune the MonoT5 based on these negative samples. Besides, we fine-tune BGE by using pair-wise entailment between the query and the candidate paragraph. Finally, for each query, we ranked the candidates by using the combination score from BGE and MonoT5. Let α and β be the score of BGE and MonoT5, the final score is computed as $0.5 * \alpha + 0.5 * \beta$.
- **Stage 2: Candidate re-ranking**: We choose top-k candidates based on the score from Stage 1. We choose $k = 5$ according to the empirical results shown in the Experiment results section. Then, in order to exploit the entailment label from each pair of query and top-k candidates, we investigate two learning scenarios: in-context learning (zero-shot) with the Qwen2.5-72B-Instruct model [26] and instruction fine-tuning [5] with the Qwen2.5-14B-Instruct model.

```
You are an advanced AI assistant specialized in
 **verifying textual entailment** between a
 query and a candidate document. Your task is to
  analyze the provided query and document
 carefully and determine whether the query is
 logically entailed by the document.
**Instructions:**
   - If the document **supports or confirms**
 the query's statement, respond with: `"entailed
 "`.
   - If the document **contradicts, refutes, or
 does not provide sufficient information** to
 confirm the query, respond with: `"not entailed
 "`.
   **Query:**
      {query}
   **Candidate Document:**
      {candidates paragraphs}
   **Think step by step before deciding.**
 Analyze whether the document provides enough **
 direct** or **inferable** support for the query.

   Your final answer must be only one of these
 two options: "entailed" or "not entailed"
**The answer is:**
```

3.2 Task 3: Statute Law Retrieval

To address Task 3 of the competition, which involves identifying the subset of Japanese Civil Code Articles (A) most relevant to a given legal bar exam question (Q), we propose a three-stage pipeline that leverages advanced zero-shot retrieval, reranking, and selection techniques using large language models (LLMs). Our approach builds on previous work but introduces a combination of models and prompting strategies to enhance retrieval and selection accuracy. Below, we detail each stage of the method:

- **Stage 1: Zero-shot Retrieval**: In the first stage, we employ the gte-Qwen2-7B-instruct model for zero-shot retrieval to identify an initial set of candidate articles from the legal corpus. This model, pre-trained on a diverse range of texts, is capable of understanding semantic relationships between the query (Q) and articles (A) without requiring task-specific fine-tuning. We retrieve the top k=150 articles based on their

relevance scores, calculated by the model's embeddings of the query and article content. This step ensures a broad initial selection of potentially relevant articles, which will be further refined in subsequent stages.

- **Stage 2: Zero-shot Reranking**: To improve the quality of the retrieved candidates, we apply a zero-shot reranking step using the RankingGPT-qwen-7b model. This model is designed to reorder the k=150 articles from Stage 1 by assessing their relevance to the query more precisely. Unlike the retrieval model, RankingGPT-qwen-7b focuses on fine-grained semantic alignment, reordering the articles to prioritize those most likely to address the legal question. We maintain k=150 at this stage to ensure a sufficiently large pool of candidates for the final selection, balancing recall and precision.
- **Stage 3: Final Selection with Fine-tuned LLMs**: In the final stage, we combine the strengths of the fine-tuned Qwen2-72B-Instruct, Qwen2-7B-Instruct, and Llama-3-8B-Instruct by employing a majority voting mechanism to select the most relevant articles from the top k=10 candidates produced by Stage 2. These models have been fine-tuned on legal texts to better understand the nuances of the Japanese Civil Code and legal bar exam questions. We use a relevance verification prompting template to evaluate the match between each article (A) and the query (Q). The prompting template is defined as follows:

```
Can the article match the question?
query: {content of the query}
article: {content of the article}
```

3.3 Task 4: Legal Textual Entailment

We present a method that utilizes the reasoning abilities of pretrained large language models (LLMs) to generate synthetic data. Our approach breaks down legal articles into smaller, more manageable sub-conditions, making their intricate structures easier to understand, minimizing noise, and revealing the complex relationships within legal sentences. This technique produces a more refined synthetic dataset, which, when integrated with the original data, improves the model's accuracy in making entailment judgments.

Our synthetic data generation framework (Figure 2) consists of three main steps: Causal Relationship Extraction for Legal Data Generation, Dataset Combination, and Fine-tuning [8].

3.3.1 Causal Relationship Extraction for Legal Data Generation. Acknowledging the significance of enriching datasets, we introduce a unique data augmentation technique that extracts causal relationships from legal articles. Our initial dataset analysis revealed that legal premises frequently take the form of sub-conditions structured as: `cause(s)` → `effect(s)`. Here, the `cause(s)` denote the conditions, while the `effect(s)` represent the corresponding consequences, judgments, or interpretations. We have identified that when a hypothesis is derived by accurately adhering to the causal relationships within a premise, it will naturally be entailed by that premise. Conversely, if the hypothesis modifies or disrupts

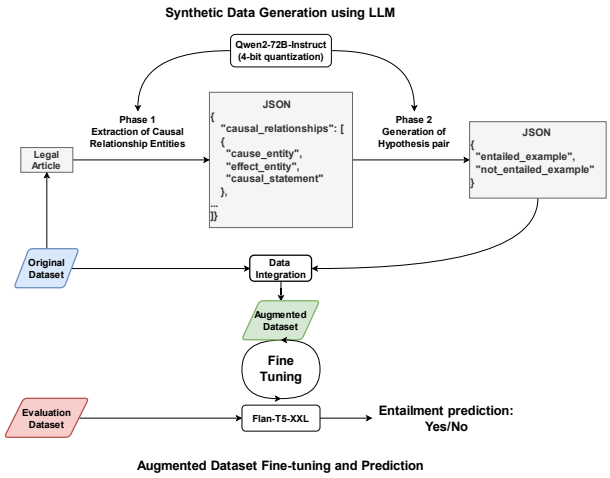


Figure 2: Overview of our proposed framework architecture.

one or more of these causal relationships, it will not be entailed by the premise.

3.3.2 Dataset Combination ratio. We conducted an experiment with three dataset combination strategies to investigate the impact of our synthetic generated data to the entailment improvement: (1) *Original:Augmented = 1:2* (Standard), where each original hypothesis is augmented with two synthetic hypotheses; (2) *Original:Augmented = 1:0.5* (Reduced), where the number of synthetic hypotheses is decreased by a factor of four; (3) *Original:Augmented = 1:1* (Balanced), where the number of synthetic hypotheses is reduced to match the size of the original dataset.

3.3.3 Fine-tuning. Building upon the SOTA methods [16] in the legal textual entailment task, we employ an approach involving instruction fine-tuning with LLMs, leveraging our synthesized dataset. Specifically, we include the *premise* and *hypothesis* data within the prompt input, training the LLMs to generate the desired output text.

3.3.4 Instruction Prompt selection. We apply the prompting method to convert the synthesized data into a question-answering format. After testing multiple prompt templates, we mainly rely on two primary prompts for this purpose.
Prompt 1: *{premise}\nQuestion: {hypothesis} True or False?*
Prompt 2: *{premise}\nQuestion: {hypothesis}? Answer with Yes or No.*

3.3.5 Ensemble Models. Unlike earlier studies [2, 16] that rely on a single LLM (e.g., Flan-T5-XXL, Flan-Alpaca-XXL, BLOOMZ-7B1, etc.) for label prediction, our method employs an ensemble model with a majority voting system. Specifically, we aggregate predictions from three separately fine-tuned Flan-T5-XXL models to determine the final output, improving both robustness and overall accuracy.

3.4 Pilot task: Legal Judgment Prediction for Japanese Tort cases

To address the Pilot Task (LJPJT25) on Legal Judgment Prediction for Japanese Tort Cases, we propose a method that leverages a fine-tuned large language model (LLM) to perform both Tort Prediction (TP) and Rationale Extraction (RE). The task requires predicting whether a tort is affirmed (T) and extracting the accepted arguments (R^P for plaintiffs and R^D for defendants) based on undisputed facts (U) and arguments from both parties (P for plaintiffs and D for defendants). Our approach utilizes the advanced capabilities of the Linkbricks-Horizon-AI-Japanese-Pro-V5-70B model, fine-tuned specifically for this task, to handle the complexity of Japanese legal texts. The model was pre-trained on a diverse dataset encompassing Japanese, Korean, Chinese, and English texts, including a 20-million-document Japanese news corpus and specialized logic judgment data. This broad training enables the model to handle the nuanced language of legal documents while maintaining cross-linguistic consistency. For the Tort Prediction task, the fine-tuned Linkbricks-Horizon-AI-Japanese-Pro-V5-70B model takes as input the undisputed facts (U) and the arguments from both parties (R^P and R^D), leveraging its pre-trained knowledge to understand the relationships between the facts and arguments. We prompt the model to predict a Boolean value (T) indicating whether the tort is affirmed (True) or not (False). The prompt is structured as follows:

```
"Given the undisputed facts (U) and arguments from the
  plaintiffs (P) and defendants (D), determine whether
  the tort is affirmed (T). Output True if the tort is
  affirmed, and False otherwise.
Undisputed facts: {content of U}
Plaintiffs' arguments: {content of P}
Defendants' arguments: {content of D}"
```

The model outputs a single Boolean value (TT) based on its understanding of the legal reasoning and the relative strength of the arguments, as determined by its fine-tuned parameters.

For the Rationale Extraction task, the model identifies the accepted arguments (R^P for plaintiffs and R^D for defendants). Using the same input (U,P,D), the model evaluates each argument in P and D to determine whether it is accepted by the judge (True) or not (False). We use a structured prompt to guide the model in this task:

```
Given the undisputed facts (UU), arguments from the
  plaintiffs (PP), arguments from the defendants (DD),
  identify the accepted arguments ($R^P$) from the
  plaintiffs and ($R^D$) from the defendants. For each
  argument in P and D, output True if the argument is
  accepted by the judge, and False otherwise.
Undisputed facts: {content of U}
Plaintiffs' arguments: {content of P}
Defendants' arguments: {content of D}
```

4 Results

4.1 Experiment preparation

4.1.1 Task 2.

Dataset. : The dataset for task 2 - Legal Case Entailment provided by the COLIEE 2025 organizer consists of two parts: training and evaluation. For the training data, there are a total of 825 legal documents, each document contains a query case (stored in the *entailed_fragment.txt* file) and a list of paragraphs (stored in a *paragraphs* directory, each text file in the folder is a paragraph of the base case). For the test data, the structure is similar to the training set. The organizer gives the ground-truth annotated data for the training set, in which each document has a list of paragraphs that are entailed in the query. We pre-process the text data by the following step: removing special characters like *FRAGMENT_SUPPRESSED*, *FACTUAL*, *BACKGROUND*, and *ORDER*, trimming unexpected space, and pruning the citation number in the text.

Evaluation Method. We used the F1 score to evaluate the performance of the legal case entailment task as described below:

$$F1 = \frac{2 * precision * recall}{precision + recall}$$

$$precision = \frac{\text{\#correctly retrieved cases (paragraphs)}}{\text{\# total retrieved cases}}$$

$$recall = \frac{\text{\#correctly retrieved cases (paragraphs)}}{\text{\# total relevant cases}}$$

Hyper-parameter setting. For the MonoT5, we run the experiment with hyper-parameter search as defined in [16]. For BGE, we run the fine-tuned with 20 epochs using the *BAAI/bge-m3* pre-trained, max length of query equals to 1,024, max length of passage equal to 3,000, and the learning rate is 10^{-5}. For the LLMs, we run the inference on Qwen with 4-bit quantization, do_sample is False, and max_new_token=3. We used the Supervised Fine-tuning Trainer (SFT) to fine-tune the Qwen model for pair-wise entailment between a query and a paragraph.

4.1.2 Task 3.

Dataset: The data is arranged by year, spanning from Heisei 18 (H18, 2006) to Reiwa 05 (R05, 2024), including Reiwa 02 (R02), Reiwa 03 (R03), and Reiwa 04 (R04). For this study, we used the R05 dataset, which includes 109 samples to evaluate the models' performance while the remaining datasets were utilized for training. As a result, the training data consists of 806 original samples, with the R02 dataset containing 81 samples, the R03 dataset comprising 109 samples, and the R04 dataset including 101 samples.

Evaluation Method: For Task 3, the performance will be assessed using precision, recall, and the F2-measure. Given that the information retrieval (IR) process serves as a preliminary step to identify candidate articles for the entailment phases. The evaluation metrics are as follows:

$$F2 = \frac{5 * precision * recall}{4 * precision + recall}$$

$$Pre = \frac{\text{\#average(the number of correctly retrieved articles/query)}}{\text{\#the number of retrieved articles/query}}$$

$$Recall = \frac{\text{\#average(the number of correctly retrieved articles/query)}}{\text{\#the number of retrieved articles/query}}$$

Hyper-parameters setting: we fine-tune Qwen2-72B-Instruct, Meta-Llama-3-8B-Instruct, Qwen2-7B-Instruct using Low-Rank Adaptation (LoRA) [8] to reduce computational costs. We set the learning rate to 1e-5, batch size to {4, 16}, LoRA r and alpha between {4, 32}, {8, 32} and 16, 32} and train for 10 epochs.

4.1.3 Task 4.

Dataset. The dataset is provided as a *Task 4: Legal Textual Entailment* by the COLIEE organizer. The dataset is used to construct Yes/No question-answering systems for legal queries, by entailment from the relevant Civil Law articles.The training data includes triples consisting of a query (hypothesis), relevant article(s) (premise(s)), and a label indicating the correct answer: "Y" for Yes (entail) or "N" for No (not entail). The dataset is organized chronologically from H18 (Heisei 18, 2006) to R05 (Reiwa 05, 2024). For this study, data from R03, R04, and R05 were chosen as evaluation sets, with the remaining years used as training data. This resulted in 806 original samples for training, a development set with 109 samples from R03, and two test sets comprising 101 and 109 samples from R04 and R05, respectively.

Evaluation Method. For the legal textual entailment task, the evaluation measure will be accuracy, concerning whether the yes/no question was correctly confirmed:

$$\text{Accuracy} = \frac{\text{the number of correctly predicted instances}}{\text{the number of all instances}}$$

LLM Selection. Following the COLIEE organizers' guideline— *"To avoid contamination of the test data, you can only use LLMs released before July 9, 2024 (JST)"*—we carried out several test runs using consistent prompts. Considering factors like benchmark dataset performance, output format precision, and the capacity to manage lengthy contexts, we opted for the Qwen2 72B[1] model with 4-bit quantization for generating synthetic data.

Hyper-parameters. To tackle the final task of legal text entailment, we fine-tuned the Flan-T5-XXL model[2] using Low-Rank Adaptation (LoRA) [8], allowing for efficient parameter tuning while minimizing computational demands. Key settings included a learning rate of 3e-5, a batch size of 32, LoRA r values between {16, 32}, LoRA alpha values between {32, 64}, a LoRA dropout rate of 0.25, and a training duration of 15 epochs.

4.1.4 Pilot Task.

Dataset: To assess our proposed approach for the Pilot Task (LJPJT25) on Legal Judgment Prediction for Japanese Tort Cases, we conducted experiments utilizing the dataset supplied by the COLIEE 2025 organizers. The 6,508 tort cases were divided into a training set and a development set at a 90:10 ratio. Specifically, the initial 5,858 cases were allocated for training, while the final 650 cases were set aside for development and evaluation.

Evaluation Method: For the Tort Prediction in the Pilot Task, the performance will be evaluated based on accuracy, determined by whether the True/False label for the court decision was accurately predicted. For the Rationale Extraction in the Pilot Task, the performance will be assessed using the F1-measure, based on the True label for is_accepted. The evaluation metrics are as follows:

$$Acc = \frac{\#\text{instances which were correctly predicted}}{\#\text{the number of all instances}} \quad (1)$$

$$F1 = \frac{2 * prec * recall}{prec + recall} \quad (2)$$

$$prec = \frac{\#\text{the number of claims correctly predicted as True}}{\#\text{the number of claims predicted as True}} \quad (3)$$

$$recall = \frac{\#\text{the number of claims correctly predicted as True}}{\#\text{the number of claims whose gold labels as True}} \quad (4)$$

Hyper-parameters setting: we fine-tune Linkbricks-Horizon-AI-Japanese-Pro-V5-70B model for both Tort Prediction (TP) and Rationale Extraction (RE) using Low-Rank Adaptation (LoRA) [8] to reduce computational costs. We set the learning rate to 1e-5, batch size to 4, LoRA r and alpha to {8, 32} and train for 5 epochs.

4.2 Task 2 results

Table 1 illustrates the results of our system on various COLIEE tests from 2022 to 2024. From the results, we choose the top-k equals 5 since this setting shows higher results in comparison with $k = 10$ (This top-5 is used to retrieve the top relevant candidates in *Stage 1* as described in Section 3). Then, we evaluate our system with different retrieval models, including BGE, MonoT5, and BM25. From the results, it can be seen that the performance of the three retrieval models, including BM25, BGE, and MonoT5, seems similar. In the 2024 test results - the most recent test from the COLIEE competition, BGE shows the best performance with $k = 5$, followed by the MonoT5 models (also with $k = 5$). In the re-ranking stage, the Qwen2.5-72B-Instruct shows its robustness in boosting the accuracy of the system for the Legal case entailment task.

In Table 2, we report the final results by three runs provided by the COLIEE 2025 organizer. We submitted three runs as follows:

- **run1_captain_qwen2572m**: We use the BM25 with MonoT5 for retrieval stage, then re-rank by Qwen-2.5-72B Instruct.
- **run2_captain_qwen2572bm**: We use the BGE with MonoT5 (both are fine-tuned on the training dataset) for retrieving top-5 candidates, then re-rank by Qwen-2.5-72B Instruct.
- **run3_captain_ensv2bge1**: We have two main steps for this run. For the first step, We use the BM25 with MonoT5 to choose the top 5 candidates. Next, in the second step, we first fine-tune the Qwen2.5-14B-Instruct by using instruction tuning to find the "entailment" between a pairwise of query q and i-th candidates. Then the outputs of various models are ensemble by voting strategy. The missing query/case will be filled by a paragraph having top-1 of the BGE score.

Table 2: Official results of our system at COLIEE 2025.

Runs	F1	Precision	Recall
run2_captain_qwen2572bm.txt	0.1882	0.2547	0.1492
run1_captain_qwen2572m.txt	0.1812	0.2453	0.1436
run3_captain_ensv2bge1.txt	0.1712	0.2252	0.1381

[1] https://huggingface.co/Qwen/Qwen2-72B-Instruct
[2] https://huggingface.co/google/flan-t5-xxl

Table 1: Comparison of our approach with other SOTAs. The results are represented in the format as f1/precision/recall

	2022	2023	2024
BM25 (k=10)	20.16 / 11.29 / 94.06	20.09 / 11.26 / 92.50	22.10 / 12.68 / 85.71
MonoT5 (k=10)	22.10 / 12.68 / 85.71	20.09 / 11.26 / 92.50	22.45 / 12.89 / 87.07
BGE-legal (k=10)	19.98 / 11.19 / 93.22	20.81 / 11.67 / 95.83	23.68 / 13.59 / 91.83
BM25 (k=5)	32.03 / 19.80 / 83.89	32.58 / 20.20 / 84.16	34.93 / 22.60 / 76.87
MonoT5 (k=5)	34.95 / 21.60 / 91.52	32.58 / 20.20 / 84.16	36.78 / 23.80 / 80.95
BGE-legal (k=5)	32.68 / 20.20 / 85.59	33.87 / 21.00 / 87.50	37.71 / 24.44 / 82.99
BGE + BM25 + Qwen2.5-72B-Instruct	76.67 / 71.85 / 82.20	79.83 / 78.86 / 80.83	65.75 / 66.20 / 65.30
MonoT5 + BM25 + Qwen2.5-72B-Instruct	69.01 / 64.23 / 74.57	72.06 / 70.07 / 74.16	72.66 / 71.24 / 74.14
BGE + MonoT5 + Qwen2.5-72B-Instruct	75.69 / 71.42 / 80.50	79.01 / 78.04 / 80.00	64.38 / 64.82 / 63.94
Top 1 results	67.83 / 69.64 / 66.10	74.56 / 78.70 / 70.83	65.12 / 63.64 / 66.67

It can be seen that the results decreased significantly in this year's released test. As shown in Table 3, the top 1 team achieved about 31.95% by F1 score, indicating the challenge of the legal case entailment task this year.

Table 3: Official results of Task 2 by teams at COLIEE 2025

Team	F1	Precision	Recall
NOWJ	0.3195	0.3788	0.2762
OVGU	0.2454	0.2759	0.2210
JNLP	0.2412	0.2000	0.3039
AIIR_Lab	0.2368	0.2927	0.1989
CAPTAIN	0.1882	0.2547	0.1492
UA	0.1778	0.2090	0.1547

4.3 Task 3 results

As described in section 3.2, our proposed methods leverage the capabilities of Zero-shot and Finetuned LLMs. The LLMs are fine tuned with datasets coded as H, R01, R02, R03, R04. and we use the R05 dataset to evaluate the performance of the models. We outline the three submission settings:

- **CAPTAIN run1**: GTE-Qwen2-7B-Instruct (Zero-shot Retrieval) + RankingGPT-qwen-7b (Zero-shot Reranking) + Qwen2-72B-Instruct (Finetuned) + Meta-Llama-3-8B-Instruct checkpoint 1 (Finetuned).
- **CAPTAIN run2**: GTE-Qwen2-7B-Instruct (Zero-shot Retrieval) + RankingGPT-qwen-7b (Zero-shot Reranking) + Qwen2-72B-Instruct (Finetuned) + Meta-Llama-3-8B-Instruct checkpoint 2 (Finetuned).
- **CAPTAIN run3**: GTE-Qwen2-7B-Instruct (Zero-shot Retrieval) + RankingGPT-qwen-7b (Zero-shot Reranking) + Qwen2-72B-Instruct (Finetuned) + Qwen2-7B-Instruct (Finetuned).

Table 4: Results of Task 3 on the R05 development set

Runs	F2	Precision	Recall
CAPTAIN_run1	0.7758	0.7477	0.8073
CAPTAIN_run2	0.7643	0.7477	0.7936
CAPTAIN_run3	0.7687	0.7401	0.8028

With the performance of our proposed methods as presented in Tables 3 and 4, CAPTAIN run1 achieved an F2 score of 0.7758, a precision of 0.7477, and a recall of 0.8073 on the R05 set (Table 3). This setting outperformed the other two CAPTAIN runs, CAPTAIN run2 recorded an F2 score of 0.7643, a precision of 0.7477, and a recall of 0.7936 (Table 3), while this run maintained the same precision as CAPTAIN run1, the slight drop in recall. CAPTAIN run3 yielded an F2 score of 0.7687, a precision of 0.7401, and a recall of 0.8028 (Table 3). This run achieved a higher recall than CAPTAIN run2 but at the cost of lower precision, demonstrating the effectiveness of combining zero-shot retrieval and reranking with fine-tuned models. When compared to the top 10 official test results at COLIEE 2025 (Table 4), CAPTAIN runs rank competitively, surpassing several strong baselines such as mstralRerank (F2=0.5962) and OVGU3 (F2=0.6041), though it falls short of the top performer, JNLP run1 (F2=0.8365).

Table 5: Top 10 Official Test Results at COLIEE 2025.

Runs	F2	Precision	Recall
JNLP_run1	0.8365	0.8037	0.8744
CAPTAIN_run2	0.8301	0.8333	0.8516
CAPTAIN_run3	0.8204	0.8002	0.8584
CAPTAIN_run1	0.8103	0.8196	0.8311
JNLP_run2	0.7863	0.7272	0.8402
JNLP_run3	0.7861	0.7420	0.8265
INFA	0.6917	0.7671	0.6826
mpnetAIIRLab	0.6674	0.3562	0.8858
OVGU3	0.6041	0.6347	0.6142
mistralRerank	0.5962	0.3196	0.7900

4.4 Task 4 results

Table 6 highlights the top-performing experimental setups evaluated on the test sets ranging from R03 to R05. Compared to existing SOTA techniques, our method yields competitive results:

Applying In-Context Learning (ICL) leads to meaningful performance gains, particularly in the R03 dataset, where it boosts results by an average of 8.72 %. However, its impact is far less pronounced on R04, with only a 0.5 % increase, and nearly negligible in Prompt 2 (0.99 %). In contrast, R05 sees a significant drop of −7.8 %, suggesting that while ICL offers benefits in some cases, it lacks

Table 6: Comparison of Our Approach with COLIEE Top 3 Results from Other Competitors

Year	Method (#original : #augmented data)	Accuracy		
		R03 (2022)	R04 (2023)	R05 (2024)
2022	KIS2 [6]	0.6789(*)	-	-
	HUKB-1 [30]	0.6697	-	-
2023	JNLP3 [2]	-	0.7822(*)	-
2024	CAPTAIN 2 [16]	-	-	0.8257(*)
	JNLP1 [7]	-	-	0.8165
2025	1. Our Zero shot (P1) *(no aug. data)*	0.7431	0.7822	0.7798
	2. Our Zero shot (P2) *(no aug. data)*	0.7890	0.7921	0.7156
	3. Our ensemble models (1:2)	0.7706	**0.8317**	0.8073
	4. Our ensemble models (1:1)	0.7706	0.8218	0.8257
	5. Our ensemble models (1:0.5)	**0.7900**	0.8218	**0.8440**

(*) indicates the winner's performance in each year. The notations P1 and P2 represent prompt 1 and prompt 2, respectively, as referred to in the Method section. Bold scores denote the best performance.

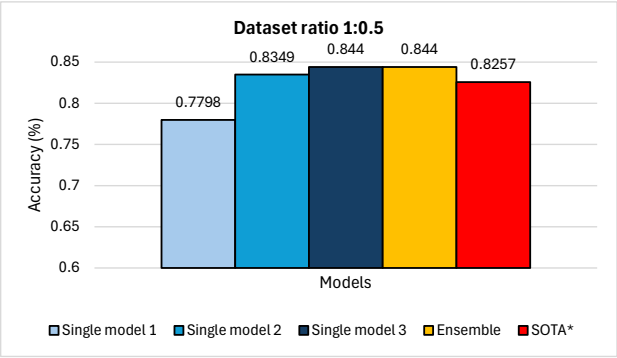


Figure 3: A performance comparison between standalone models, their ensembles, and current SOTA models.

reliability for complex, domain-specific tasks, reinforcing the need for targeted fine-tuning.

Compared to previous state-of-the-art (SOTA) methods, our data augmentation strategy consistently delivers superior performance across all datasets. Notably, it exceeds KIS Team's results by [6] by 9.82 %±0.91 % on R03 and improves upon JNLP Team's 2023 [2] performance on R04 by 4.29 %±0.47 %. The performance on R05 is modest, however, the 1:0.5 dataset integration ratio exceeds the SOTA performance by 1.83 %, as well as maintains strong performance across R03 and R04.

Focusing on the ensemble model trained with the 1:0.5 ratio, Figure 3 shows that Models 2 and 3 consistently outperform the SOTA benchmarks. Although Model 1 underperforms, the ensemble as a whole achieves stronger overall results than existing SOTA methods. Despite the fact that the 1:1 and 1:2 dataset combinations fall short of SOTA, the 1:0.5 configuration proves to be more effective for legal entailment tasks.

Overall, these findings emphasize the strength of leveraging large language models (LLMs) for synthetic data generation and reasoning, offering clear advantages over traditional rule-based or semi-automated augmentation techniques.

Table 7: Top 10 Official Test Results for Task 4 - COLIEE 2025. The setting numbers indicate the corresponding setting number in the development results in Table 6

Team	Correct/All	Accuracy
KIS3	67/74	0.9054
LUONG01	64/74	0.8649
UIRunCot	63/74	0.8514
CAPTAIN2 (setting 5)	60/74	0.8108
JNLP002	60/74	0.8108
CAPTAIN1 (setting 4)	58/74	0.7838
CAPTAIN3 (setting 6)	58/74	0.7838
UA2	58/74	0.7838
KLAP.H2	57/74	0.7432
NOWJ.run1	55/74	0.7432
OVGU1	55/74	0.6622
RUG_V1	45/74	0.6081

Despite these promising results, our approach may be overspecialized due to fine-tuning on a narrowly defined prompt set during the hypothesis generation phase. This limitation is evident in the official results of COLIEE 2025 (Table 7), where we ranked 4th, highlighting areas for improvement. We hypothesize that this discrepancy stems from the prompt design in phase 2 and the complexity of causal relationships within the premise. Additionally, generating only a single entailed and not-entailed hypothesis may be insufficient for premises with multiple conditions, limiting the quality and scope of the hypotheses. As a result, this constraint affects fine-tuning, reducing the model's ability to generalize to new cases, and potentially leading to misinterpretations and incorrect predictions. Future work will focus on diversifying synthetic data generation and optimizing prompt engineering to enhance model generalization and performance.

4.5 Pilot task results

To evaluate our proposed method for the Pilot Task (LJPJT25) on Legal Judgment Prediction for Japanese Tort Cases, we conducted experiments using the dataset provided by the COLIEE 2025 organizers. We split the 6,508 tort cases into a training set and a development set in a 90:10 ratio. Specifically, the first 5,858 tort cases were used for training, while the last 650 tort cases were reserved for development and evaluation. This split ensures a robust training corpus while providing a sufficiently large development set to assess model performance. For simplicity, we only use Accuracy to evaluate both the Tort Prediction (TP) and Rationale Extraction (RE) tasks in our experiments. Our approach leverages the fine-tuned *Linkbricks-Horizon-AI-Japanese-Pro-V5-70B* model for both Tort Prediction (TP) and Rationale Extraction (RE). We fine-tuned the *Linkbricks-Horizon-AI-Japanese-Pro-V5-70B* model on the training set of 5,858 tort cases, optimizing its parameters for the nuances of Japanese legal texts. The model's performance was then evaluated on the development set of 650 tort cases, with results reported in Table 8. For the official submission, we used the same fine-tuned model and evaluated its performance on the COLIEE 2025 test set, with results presented in Table 9.

Table 8: Results of Pilot Task on the development set including Tort Prediction (TP) and Rationale Extraction (RE).

Runs	TP (Accuracy)	RE (Accuracy)
CAPTAIN_run1	72.8%	70.7%
CAPTAIN_run2	71.4%	68.4%
CAPTAIN_run3	70.2%	66.5%

Table 9: Official Pilot Task Results at COLIEE 2025 including Tort Prediction (TP) and Rationale Extraction (RE).

Runs	TP (Accuracy)	RE (F1-All)
CAPTAIN_run1	**76.5%**	70.6%
KIS5	71.3%	**71.2%**
KIS6	71.3%	67.3%
KIS4	69.7%	68.2%
NOWJ2	67.1%	69.2%
modernbert	66.6%	69.1%
NOWJ1	63.8%	68.1%
NOWJ3	59.7%	55.9%
OVGU2	55.3%	48.6%
OVGU3	53.2%	31.6%
OVGU1	51.5%	65.7%

With the performance of our proposed method for the Pilot Task (LJPT25) on Legal Judgment Prediction for Japanese Tort Cases, as shown in Table 8 and 9. On the development set (Table 8), , the best performance is observed with CAPTAIN run1, achieved a Tort Prediction (TP) accuracy of 72.8% and a Rationale Extraction (RE) accuracy of 70.7%. Subsequent runs, CAPTAIN run2 and CAPTAIN run3, yielded TP accuracies of 71.4% and 70.2%, respectively, with corresponding RE accuracies of 68.4% and 66.5%. On the official COLIEE 2025 test set (Table 9), CAPTAIN recorded a TP accuracy of 76.5% and an RE accuracy of 70.6%. This result positions CAPTAIN as the top performer in TP accuracy, outperforming KIS5 (TP: 71.3%) and KIS6 (TP: 71.3%). The improvement in TP accuracy from the development set (72.8%) to the test set (76.5%) suggests that our model generalizes well to unseen data, likely due to the robust fine-tuning on a large training set of 5858 tort cases. For RE, CAPTAIN's accuracy of 70.6% is competitive but slightly lower than KIS5 (71.2%). Comparing CAPTAIN to other submissions in Table 9, our method significantly outperforms lower-ranking runs such as OVGU1 (TP: 51.5%, RE: 65.7%) and OVGU3 (TP: 53.2%, RE: 31.6%), highlighting the effectiveness of our fine-tuning strategy and prompt design. However, the gap in RE performance between CAPTAIN and KIS5 suggests that further improvements in argument extraction are needed. This could involve refining the prompt to better guide the model in distinguishing accepted arguments or incorporating additional legal knowledge into the fine-tuning process.

4.6 Discussion

For task 2, the combination of the retrieval model consists of MonoT5 and BGE with Qwen2.5-72-Instruct helps improve the performance of the system for legal case entailment tasks. We finally achieved a 0.18882 by F1 score on the original test released by COLIEE 2025. In comparison with previous years' results, there is a significant decrease in the performance of models. According to the results in COLIEE 2024 [7], the top 1 result on task 2 is 65.12% by F1, while this year's best result is only about 31.95% by F1 (as shown in Table 3), indicating the challenge for the legal entailment task. Obviously, it can be seen that the system seems to overfit the training dataset and cannot generate the generalized result for the practical test cases.

Next, our proposed method demonstrates strong performance on Task 3, which achieves a competitive F2 score of 0.7758 on the R05 test set and ranks among the top submissions with a test F2 score of 0.7769. The combination of zero-shot retrieval, reranking, and fine-tuned LLMs (GTE-Qwen2-7B-Instruct, RankingGPT-qwen-7b, Qwen2-72B-Instruct, and Meta-Llama-3-8B-Instruct) effectively balances precision and recall, though CAPTAIN run1 slightly outperforms the other runs due to its higher recall. While our approach surpasses several baselines like mstralRerank (F2: 0.5672) and OVGU3 (F2: 0.5654), the gap with the top performer, JNLP run1 (F2: 0.7829), highlights the need for further refinement in the final selection stage.

Besides, we found that LLMs can effectively extract causal relationships (sub-conditions) defined in law articles when solving Task 4. Through a two-step synthetic data generation process, we observed that our generated dataset—when combined with a specific training-to-synthetic ratio—enhanced fine-tuning performance by nearly 2 % compared to conventional synthetic methods. While this approach proved effective, we recognize its limitations in handling nested causal relationships and long articles that reference other legal texts. In addition, the Pilot Task results demonstrate the strength of the Linkbricks-Horizon-AI-Japanese-Pro-V5-70B model in handling Japanese tort cases, particularly for Tort Prediction. The competitive RE performance of CAPTAIN indicates that our approach is promising but requires further optimization to match the best-performing submissions in rationale extraction.

Overall, it can be seen that Large-language models have a robust ability to process and understand legal text data, even with non-English languages like Japanese. However, there are still several challenges to the application of LLMs in the legal text, especially the ability to adapt to the diversity and change in the legal domain. We propose several future directions to enhance the performance of LLMs in Legal Text Processing tasks including using advanced retrieval methods like GRAG [9] to better exploit the semantic relations among paragraphs that represent the relevant with the query, enhancing the fine-tuning process and exploring advanced prompting strategies such as Self-prompting [14] or Plan-and-Solve [24] to improve overall retrieval and selection accuracy, and augmenting the dataset in training and fine-tuning the model to avoid overfiting.

5 Conclusion

We described our proposed method and system for Task 2, Task 3, Task 4, and Pilot task in the COLIEE 2025 Competition. From the competition results, we found that the LLMs have potential results in legal engineering tasks since they show robust performance in processing and understanding legal documents. Our team - CAPTAIN, achieved 2^{nd} in Task 3, 1^{st} in Tort Prediction (TP), and 2^{nd} in Rationale Extraction (RE) from the Pilot task.

Future works focus on robust prompting techniques and advanced retrieval methods to enhance the performance of LLMs in legal text processing systems. In addition, the LLMs show potential application in practical intelligent systems that assist humans in the legal domain.

References

[1] MQ Bui, C Nguyen, DT Do, NK Le, DH Nguyen, and TTT Nguyen. 2022. Using deep learning approaches for tackling legal's challenges (COLIEE 2022). In *Sixteenth International Workshop on Juris-informatics (JURISIN)*.

[2] Quan Minh Bui, Dinh-Truong Do, Nguyen-Khang Le, Dieu-Hien Nguyen, Khac-Vu-Hiep Nguyen, Trang Pham Ngoc Anh, and Minh Nguyen Le. 2023. 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)*.

[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* (2020).

[4] 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

[5] Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2022. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416* (2022).

[6] Masaki Fujita, Takaaki Onaga, Ayaka Ueyama, and Yoshinobu Kano. 2023. Legal Textual Entailment Using Ensemble of Rule-Based and BERT-Based Method with Data Augmentation by Related Article Generation. In *New Frontiers in Artificial Intelligence*, Yasufumi Takama, Katsutoshi Yada, Ken Satoh, and Sachiyo Arai (Eds.). Springer Nature Switzerland, Cham, 138–153.

[7] 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.

[8] 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*. https://openreview.net/forum?id=nZeVKeeFYf9

[9] Yuntong Hu, Zhihan Lei, Zheng Zhang, Bo Pan, Chen Ling, and Liang Zhao. 2024. Grag: Graph retrieval-augmented generation. *arXiv preprint arXiv:2405.16506* (2024).

[10] Daniel Katz, Michael Bommarito, and Josh Blackman. 2017. A general approach for predicting the behavior of the Supreme Court of the United States. *PLOS ONE* 12 (04 2017). doi:10.1371/journal.pone.0174698

[11] MY Kim, J Rabelo, and R Goebel. 2021. Bm25 and transformer-based legal information extraction and entailment. In *Proceedings of the COLIEE Workshop in ICAIL*.

[12] BENJAMIN LAUDERDALE and TOM CLARK. 2012. The Supreme Court's Many Median Justices. *American Political Science Review* 106 (11 2012). doi:10.1017/S0003055412000469

[13] 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]

[14] Junlong Li, Jinyuan Wang, Zhuosheng Zhang, and Hai Zhao. 2024. Self-Prompting Large Language Models for Zero-Shot Open-Domain QA. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, Kevin Duh, Helena Gomez, and Steven Bethard (Eds.). Association for Computational Linguistics, Mexico City, Mexico, 296–310. doi:10.18653/v1/2024.naacl-long.17

[15] M Lin, SC Huang, and HL Shao. 2022. 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)*.

[16] 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 *JSAI International Symposium on Artificial Intelligence*. Springer, 125–139.

[17] Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. Ms marco: A human-generated machine reading comprehension dataset. (2016).

[18] 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 *JSAI International Symposium on Artificial Intelligence*. Springer, 200–211.

[19] Rodrigo Nogueira, Zhiying Jiang, and Jimmy Lin. 2020. Document ranking with a pretrained sequence-to-sequence model. *arXiv preprint arXiv:2003.06713* (2020).

[20] Guilherme Moraes Rosa, Luiz Bonifacio, Vitor Jeronymo, Hugo Abonizio, Roberto Lotufo, and Rodrigo Nogueira. 2022. Billions of parameters are worth more than in-domain training data: A case study in the legal case entailment task. *arXiv preprint arXiv:2205.15172* (2022).

[21] 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.

[22] Matthew Schultz and Thorsten Joachims. 2003. Learning a distance metric from relative comparisons. *Advances in neural information processing systems* 16 (2003).

[23] Hsuan-Lei Shao, Yi-Chia Chen, and Sieh-Chuen Huang. 2021. 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*. Springer, 226–239.

[24] Lei Wang, Wanyu Xu, Yihuai Lan, Zhiqiang Hu, Yunshi Lan, Roy Ka-Wei Lee, and Ee-Peng Lim. 2023. Plan-and-Solve Prompting: Improving Zero-Shot Chain-of-Thought Reasoning by Large Language Models. 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, 2609–2634. doi:10.18653/v1/2023.acl-long.147

[25] Sabine Wehnert, Viju Sudhi, Shipra Dureja, Libin Kutty, Saijal Shahania, and Ernesto W De Luca. 2021. 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*. 285–294.

[26] 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).

[27] Hai Ye, Xin Jiang, Zhunchen Luo, and Wenhan Chao. 2018. Interpretable Charge Predictions for Criminal Cases: Learning to Generate Court Views from Fact Descriptions. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, Marilyn Walker, Heng Ji, and Amanda Stent (Eds.). Association for Computational Linguistics, New Orleans, Louisiana, 1854–1864. doi:10.18653/v1/N18-1168

[28] Masaharu Yoshioka, Yasuhiro Aoki, and Youta Suzuki. 2021. 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*. 278–284.

[29] Masaharu Yoshioka, Youta Suzuki, and Yasuhiro Aoki. 2023. 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*. Springer, 109–124.

[30] Masaharu Yoshioka, Youta Suzuki, and Yasuhiro Aoki. 2023. HUKB at the COLIEE 2022 Statute Law Task. In *New Frontiers in Artificial Intelligence*, Yasufumi Takama, Katsutoshi Yada, Ken Satoh, and Sachiyo Arai (Eds.). Springer Nature Switzerland, Cham, 109–124.

NOWJ@COLIEE 2025: A Multi-stage Framework Integrating Embedding Models and Large Language Models for Legal Retrieval and Entailment

Hoang-Trung Nguyen
VNU University of Engineering and Technology
Hanoi, Vietnam
20020083@vnu.edu.vn

Tan-Minh Nguyen
Japan Advanced Institute of Science and Technology
Ishikawa, Japan
minhnt@jaist.ac.jp

Xuan-Bach Le
VNU University of Engineering and Technology
Hanoi, Vietnam
22024506@vnu.edu.vn

Tuan-Kiet Le
VNU University of Engineering and Technology
Hanoi, Vietnam
22024546@vnu.edu.vn

Khanh-Huyen Nguyen
VNU University of Engineering and Technology
Hanoi, Vietnam
22026502@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

Le-Minh Nguyen
Japan Advanced Institute of Science and Technology
Ishikawa, Japan
nguyenml@jaist.ac.jp

Abstract

This paper presents the methodologies and results of the NOWJ team's participation across all five tasks at the COLIEE 2025 competition, emphasizing advancements in the Legal Case Entailment task (Task 2). Our comprehensive approach systematically integrates pre-ranking models (BM25, BERT, monoT5), embedding-based semantic representations (BGE-m3, LLM2Vec), and advanced Large Language Models (Qwen-2, QwQ-32B, DeepSeek-V3) for summarization, relevance scoring, and contextual re-ranking. Specifically, in Task 2, our two-stage retrieval system combined lexical-semantic filtering with contextualized LLM analysis, achieving first place with an F1 score of 0.3195. Additionally, in other tasks–including Legal Case Retrieval, Statute Law Retrieval, Legal Textual Entailment, and Legal Judgment Prediction–we demonstrated robust performance through carefully engineered ensembles and effective prompt-based reasoning strategies. Our findings highlight the potential of hybrid models integrating traditional IR techniques with contemporary generative models, providing a valuable reference for future advancements in legal information processing.

CCS Concepts

• **Applied Computing** → **Law**; • **Computing Methodologies** → **Natural Language Processing**.

COLIEE 2025, Chicago, USA

Keywords

Legal Information Processing, Document Retrieval, Textual Entailment, Multi-stage, Embedding Models, LLMs

ACM Reference Format:
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 COLIEE 2025 workshop, June 20, 2025, Chicago, USA.* ACM, New York, NY, USA, 10 pages.

1 Introduction

Legal text processing is a specialized field that requires knowledge of both law and information science. The use of artificial intelligence (AI) and large language models (LLMs) as supporting tools in judicial processes has become more prevalent [9, 15, 20]. COLIEE [5] is an annual event organized to support the research of legal information processing. The competition covers various challenges, including document retrieval, textual entailment, and judgment prediction. COLIEE is a valuable opportunity for researchers to explore and evaluate various advanced techniques in complex real-world judicial problems.

The COLIEE 2025 competition comprises five tasks that span two major legal systems: case law and statute law. Tasks 1, 2, and 5 focus on case law, drawing on legal cases from the Federal Court of Canada and Japanese court decisions. Task 1 (Legal Case Retrieval) involves identifying relevant precedents that support the decision of a given case, serving as a foundational step for Task 2 (Legal Case Entailment), which aims to determine whether specific paragraphs within the retrieved cases entail the decision. Task 5 (Legal Judgment Prediction) targets the prediction of judicial outcomes

in real-world civil cases from Japanese courts. In contrast, Tasks 3 and 4 are based on the statute law system, which is followed by countries such as Japan, Vietnam, and many European nations. Task 3 (Statute Law Retrieval) requires participants to retrieve statutory articles from the Japanese Civil Code corpus that support a given legal query. Building upon this, Task 4 (Legal Textual Entailment) challenges systems to provide a binary answer—"Yes" or "No"—based on the retrieved legal content.

COLIEE 2025 is the third year the NOWJ team has participated. We propose multi-state frameworks based on state-of-the-art models such as monoT5, Llama-3, Qwen-2, DeepSeek, and prompting techniques for handling complex legal tasks of the competition. Specifically, a four-state framework involving meticulous pre-processing, LLM-based summarization, and different retrieval approaches is proposed for Task 1. For Task 2, we combined the advantages of lexical matching, semantic ranking methods, and recent LLMs (e.g., QwQ, DeepSeek) to model the entailment relationships between judicial factors. For Task 3, we leveraged pre-trained language models to develop a retrieval method based on bi-encoder and cross-encoder models. Various LLMs and prompting techniques, including zero-shot and few-shot prompts, are explored in Task 4. Finally, we employed a hierarchical language model, clustering approach, and heuristic post-processing phase for addressing real-world judgment prediction problems in Task 5. Our proposed methods achieved promising results, including first place in Task 2, and a third rank on the Tasks 3 and 5 leaderboard. Our findings highlight the potential of hybrid models integrating traditional IR techniques with contemporary generative models, providing a valuable reference for future advancements in legal information processing.

The remainder of this paper presents our methods used in the competition, with each section dedicated to a specific task. The final section concludes the paper and outlines directions for future research. The source code will be released shortly to support the reproducibility of our work.

2 Task 1: Legal Case Retrieval

2.1 Task Overview

Many countries, including Canada and the United States, follow the common law system, where case law is a fundamental component of judicial practice. Judges and legal professionals rely on precedents when handling new cases. To advance research in legal processing, the Legal Case Retrieval task aims to identify noticed cases—previous cases that support a given case decision. Specifically, this task requires extracting all supporting cases $\{d_1, d_2, \ldots, d_n\}$ that are semantically or logically similar to a given query case q. A standard case document consists of four main sections: background, facts, reasoning, and decision. However, the lack of a unified document structure makes case presentation and extraction challenging. Additionally, case documents typically range from 4,000 to 10,000 words, exceeding the input limits of many pre-trained language models (e.g., BERT, T5), making efficient processing difficult.

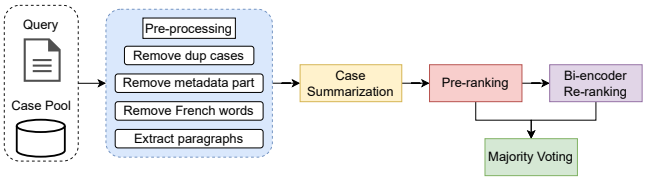


Figure 1: Overall architecture of multi-stage framework for Legal Case Retrieval.

Last year, the winning team, TQM [7], combined lexical and dense models to generate features and improve case relevance understanding. They also applied meticulous pre-processing and post-processing to filter out irrelevant information. The runner-up team, UMNLP [2], proposed a pairwise similarity ranking framework at multiple levels, including paragraphs, sentences, and "propositions". They trained a multilayer perceptron model to assess case relevance using various features extracted from each query-candidate pair.

2.2 Methodology

To overcome the challenges of Legal Case Retrieval, which are excessive length and logical structure of case documents, we proposed a four-stage framework involving LLMs for summarization and massive text embedding models for computing case relevance. The detailed architecture of the framework is presented in Figure 1.

Data Pre-processing. The case pool contained noise, including duplication and line break errors. Therefore, we performed a data processing step before training or computing case relevance. This process involved removing duplicate files and filtering out metadata, which included procedural details such as related parties and locations. Non-English sentences and passages were also removed using the Langdetect tool. Finally, paragraphs containing special placeholders were extracted. Through this pre-processing, we aimed to minimize irrelevant information in the case documents, ensuring that only content relevant to the judgment remained.

Case Summarization. To enhance the representation of case law documents, we applied abstractive summarization using an LLM with zero-shot prompting. Specifically, Qwen-2.5 is instructed to generate a concise summary of each case in the predefined format shown in Listing 1. This LLM is trivial for processing long context due to its context length of up to 131,072 tokens and generation of 8,192 tokens. By leveraging the in-context learning ability of LLMs, we compressed legal cases while preserving key facts and essential information for relevance comparison.

Listing 1: Zero-shot prompting for structured-based case summarization.

```
Summarize the following Federal Court decision containing these
parts:
Introduction, Facts, Relevant provisions, Analyses, and Court's
conclusion
Legal case:
{INPUT_CASE}
Generated summary:
```

Pre-ranking Model. The pre-ranking step serves as the first filter to select the most relevant case within a large number of candidates

in the database. Therefore, a pre-ranking model should satisfy both performance and recall score, which measure the ability to find all the relevant cases. The relevance score between a query case q and a candidate case d is calculated as:

$$s_p(q,d) = Sim(\mathbf{h}_q, \mathbf{h}_d) \tag{1}$$

where $\mathbf{h}_q$ and $\mathbf{h}_q$ are the representation of query q and candidate d from the pre-ranking model. Specifically, we employed a pre-trained model, BGE-m3, due to its strong generalization and context length of up to 8,192 tokens. Top-k candidates with the highest relevance score are selected as the input for the re-ranking step.

Re-ranking Model. We employed two methods for the re-ranking phase: (1) fine-tuned a text embedding model–BGE-m3 on the COLIEE training set and (2) utilized a recent decoder-only model for text encoding–LLM2Vec. The BGE-m3 model is trained to distinguish relevant cases from irrelevant ones. Each retrieval method is expected to assign a higher score to a query's positive samples compared to its negative ones. To achieve this, the training process minimizes the InfoNCE loss, defined as follows:

$$\mathcal{L}_{InfoNCE} = -\log \frac{\exp(s(q,p^+)/\tau)}{\sum_{p\in\{p^+,p^{easy-},p^{hard-}\}} \exp(s(q,p)/\tau)} \tag{2}$$

where p^+, p^{easy-}, and p^{hard-} stand for positive, easy negative, and hard negative samples to the query q; $s(\cdot)$ is a similarity function (e.g., the dot product or cosine similarity), τ is the temperature coefficient. In this process, the positive samples are the ground truth labels, while the easy negative samples are selected randomly from the case pool. Hard negative cases are selected using a pre-ranking relevance score for effective model training.

Another re-ranking model is LLM2Vec, a decoder-only model for text encoding by performing three steps: enabling bi-directional attention, masked next token prediction, and unsupervised contrastive learning. The relevance score of a query q and candidate d is computed as:

$$s_r(q,d) = Sim(\mathbf{h}'_q, \mathbf{h}'_d) \tag{3}$$

where $\mathbf{h}'_q$ and $\mathbf{h}'_d$ are the representation of query q and candidate d from the re-ranking model, which could be fine-tuned BGE-m3 or LLM2Vec.

Post-processing. Finally, we combined the top-m retrieved candidates from pre-ranking and re-ranking steps following the majority voting method to improve the performance and recall score. The voting operator can be defined as:

$$Final\_case = mode(TopM(Pre), TopM(FtBGE), TopM(LLM)) \tag{4}$$

where $TopM(\cdot)$ is the function that returns top cases with the highest relevance score.

2.3 Experimental Setup

The training set of COLIEE 2025 contains 7,350 cases, with an average of 4.1 relevant cases per query. The testing set consists of 2,159 files in the case pool and 400 cases for querying. The proposed method is evaluated on the COLIEE 2024 benchmark comprising 400 query cases and 1,734 documents in the pool. The official evaluation metrics for Legal Case Retrieval are precision, recall, and micro-F1.

All models were implemented using Python and HuggingFace platform. Qwen-2.5-14B-Instruct[1] is utilized for case summarization due to its strengths in handling long context input. Massive text encoder model BGE-m3 was deployed for both pre-ranking and re-ranking phases. LLM2Vec-Meta-Llama-8B-Instruct[2] is another model used for the re-ranking phase. For the fine-tuning process, BGE-m3 was trained for 4 epochs, with a batch size of 8 and an initial learning rate of $1e^{-5}$. The number of hard and easy negative samples is 3. The majority voting step combines the top 10 cases from each model.

Based on the evaluation results, we submitted three settings as follows:

- Run 1: Top-5 candidates from re-ranking using fine-tuned BGE-m3.
- Run 2: Top-5 candidates from re-ranking using pre-trained LLM2Vec
- Run 3: Majority voting of pre-ranking, fine-tuned BGE-m3, and LLM2Vec outputs.

2.4 Result and Discussion

Table 1 presents the recall performance of the pre-ranking stage under two settings: with and without the case summarization step. Retrieval at top-200 and top-500 shows strong performance, successfully retrieving between 78% and 89% of the ground truth cases. The summarization step further improves recall across most metrics, though a slight drop is observed at $R@1$. Balancing retrieval effectiveness and computational efficiency, we select the top 200 returned cases from the pre-ranking stage as candidates for the subsequent retrieval process.

Table 2 reports the performance of the proposed methods on the COLIEE 2024 benchmark. Among them, the ensemble run yields the best results across all metrics, with a particularly notable improvement in recall. Compared to the baseline, defined as the top-5 results from the pre-ranking phase, the proposed method improves the F1 score by 5%, demonstrating the effectiveness of combining ranking and representation techniques. Notably, the pre-trained LLM2Vec model performs comparably to the fine-tuned BGE-m3, highlighting the potential of general-purpose language models in the legal case retrieval domain, even without task-specific fine-tuning.

Table 3 presents the official results of the Legal Case Retrieval task, which saw participation from seven teams with a total of 21 submitted runs. Our proposed framework, combining the strengths of a pre-ranking phase and fine-tuned text embedding models, secured fourth place on the leaderboard. As anticipated, the run using fine-tuned BGE-m3 (Run 1) outperformed the one based on the pre-trained LLM2Vec model (Run 2) by approximately 2% across all metrics. The ensemble run, which integrates outputs from both approaches, achieved the highest performance among our submissions, demonstrating the benefit of combining complementary methods.

[1] https://huggingface.co/Qwen/Qwen2.5-14B-Instruct
[2] McGill-NLP/LLM2Vec-Meta-Llama-3-8B-Instruct-mntp

Table 1: The recall performance of pre-ranking phases on the evaluation set.

	R@1	R@2	R@5	R@10	R@100	R@200	R@500	R@1000
w/o summary	0.0755	0.1133	0.1939	0.2797	0.6280	0.7387	0.8687	0.9507
summary	0.0723	0.1165	0.2099	0.2970	0.6824	0.7784	0.8950	0.9654

Table 2: Performance of the proposed methods on the evaluation set.

Model	F1	Precision	Recall
Pre-ranking	0.1701	0.1515	0.1939
BGE-m3-ft	0.2262	0.2015	0.2580
LLM2Vec	0.2167	0.1930	0.2471
Majority Voting	**0.2611**	**0.2153**	**0.3316**

Table 3: Leaderboard of the Legal Case Retrieval task.

Team	F1	Precision	Recall
JNLP	0.3353	0.3042	0.3735
UQLegalAI	0.2962	0.2908	0.3019
AIIR Lab	0.2171	0.2040	0.2319
NOWJ_run3	0.1984	0.1670	0.2445
NOWJ_run1	0.1708	0.1605	0.1825
NOWJ_run2	0.1580	0.1485	0.1688
OVGU	0.1498	0.1743	0.1313
UB_2025	0.1363	0.1955	0.1046
SIL	0.0058	0.0054	0.0063

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 aims to identify the specific paragraph $p \in R$ that entails the decision d. This task presents a fine-grained challenge in legal text understanding, as multiple paragraphs may reference related legal issues without directly supporting the decision. Unlike Task 1, which operates at the case level, Task 2 evaluates textual entailment at the paragraph level, using the same metrics.

The dominance of monoT5-based models among top-performing teams in COLIEE 2024 [5] underscores their effectiveness for legal entailment tasks. The AMHR team [13] achieved the highest performance by fine-tuning a monoT5 model (pre-trained on MS-MARCO), enhanced with hard negatives selected via BM25 and further refined using a score-ratio threshold tuned via grid search. Similarly, CAPTAIN [11] fine-tuned monoT5 with hard negative sampling, then selected top-k candidate paragraphs to construct zero-shot and few-shot prompts for FlanT5-based in-context learning. The JNLP team [10] also fine-tuned monoT5 on the task dataset and incorporated FlanT5 and Mixtral models for prompting-based inference.

3.2 Methodology

Our system follows a three-stage pipeline to identify the most relevant paragraphs that entail a given decision.

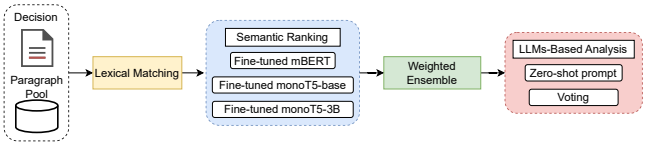


Figure 2: The three-stage framework based on lexical matching, language models, and LLMs for Legal Case Entailment.

Lexical Pre-ranking: In the first stage, we employ BM25 to retrieve paragraphs with high lexical overlap with the decision efficiently. This serves as a computationally inexpensive filtering step, which helps reduce the candidate pool while maintaining high recall of potentially entailing paragraphs.

Semantic Re-ranking: Next, we apply PLMs such as BERT and monoT5 to re-rank the paragraphs retrieved by BM25. These models assess each (q, p_i) pair to capture deep semantic relationships beyond lexical matching. To adapt PLMs for legal text, both BERT and monoT5 are fine-tuned using the Cross-Entropy (CE) loss function, defined as:

$$\mathcal{L}_{CE} = -\sum_{i=1}^{N} [y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)] \tag{5}$$

During training, each query-paragraph pair (q, p_i) is labeled as either positive relevant, $y_i = 1$ or negative non-relevant, $y_i = 0$. Negative labels are obtained by randomly sampling paragraphs from the set of paragraphs not relevant to the given query. The models learn to differentiate these labels by minimizing the CE Loss, thus effectively enhancing their capability to distinguish between relevant and non-relevant paragraphs. After training, we combine the lexical and semantic scores to produce a refined ranking and select the top-k paragraphs for further analysis. The value of k is empirically determined to balance retrieval performance with computational efficiency.

LLM-Based Analysis: In the final stage, we use LLMs to identify which paragraphs entail the decision. Rather than evaluating each (q, p_i) pair separately, we construct a single prompt that includes the decision q and all k candidate paragraphs with a clear instruction asking the model to determine which paragraphs support the decision. This holistic setup enables the LLM to consider inter-paragraph relationships and make more informed entailment judgments. The specific instruction is shown in Listing 2, where we told the LLM to pick just one paragraph that best explains the decision's reasoning. They could only choose two paragraphs in cases where both were strictly necessary.

Listing 2: Zero-shot prompting for paragraph entailment.

```
You are a legal expert tasked with identifying which paragraph(s)
from a noticed case contain the reasoning or analysis that supports
the decision of a new case. Below is the decision of the new case
(query) and the paragraphs from the noticed case. Your task is to
identify the single best paragraph that provides the reasoning or
analysis leading to the decision. Only if there are two paragraphs
that are both significantly important, equally critical, and
absolutely necessary, you may return two. Otherwise, return only
one. Do not select the paragraph that states the final decision or
order. Instead, focus on the paragraph(s) that contain the
reasoning or analysis that supports the decision.

Query (Decision of the New Case):
{query}

Paragraphs from the Noticed Case:
{paragraphs}

Which paragraph(s) contain the reasoning or analysis that supports
the decision (entails the decision)?
```

3.3 Experimental Setup

We fine-tune three Transformer-based language models: mBERT[3], monot5-base[4] and monot5-3b[5]. During training, we randomly select 5 negative samples from the top 20 BM25 results to help the models learn more stably, rather than immediately using potentially confusing hard negatives. The fine-tuning configuration consists of 3 epochs, a learning rate of $1e^{-5}$, and a batch size of 8. The scores from these language models and BM25 are combined using weights optimized via grid search to produce the re-ranked list. Finally, we experiment with state-of-the-art LLMs such as DeepSeek-V3[6] and QwQ-32B[7].

We submitted three settings as follows:

- **Run 1**: Starting from the top 20 BM25-retrieved paragraphs, we re-rank them using the combined relevance scores. A threshold tuned on the validation set is applied to this list, and all paragraphs with scores above the threshold are predicted as entailing.
- **Run 2**: Instead of applying a threshold, this run feeds the 20 re-ranked paragraphs along with the decision query into QwQ-32B to get the final prediction.
- **Run 3**: We expand the input to the top 35 candidates and prompt both DeepSeek-V3 and QwQ-32B for entailment prediction. A voting strategy is then applied: only paragraphs selected by both models are kept. In cases of complete disagreement, both predictions are retained to ensure coverage.

3.4 Result and Discussion

Following Table 4, the results suggest a progression in effectiveness based on the methodology employed. Initially, relying solely on combining reranking models based on relevance correlation appears to have limitations. These methods often evaluate passages individually, observing only a single paragraph in relation to the entailment during their core inference. This can lead to difficulties in distinguishing between paragraphs that are merely topically similar versus those that directly support or refute an entailment. Furthermore, such approaches may struggle when multiple distinct paragraphs are truly relevant, as they might overly focus on the single highest-scoring match rather than identifying a comprehensive set of evidence.

Following this initial phase, the introduction of an LLM for re-evaluating a curated list of potentially relevant passages (like the top 20 in Run 2) demonstrated a clear improvement, particularly in precision. By processing multiple candidate sentences together, the LLM can leverage broader context, leading to a more nuanced assessment of relevance and filtering out some spurious correlations identified by the initial re-rankers.

The most successful strategy involved both expanding the candidate pool (top 35) and implementing a stricter validation mechanism using two distinct LLMs in a voting agreement (Run 3). Expanding the pool likely increased the chances of capturing all necessary evidence while requiring consensus between two LLMs to act as a strong filter against false positives. This combination significantly boosted precision, indicating a higher confidence in the selected evidence and ultimately leading to the best overall F1 performance observed. This highlights the power of combining a wider search with multi-perspective LLM-based verification to enhance both the reliability and accuracy of the final results.

Table 4: Leaderboard of the Case Textual Entailment task.

Team	F1	Precision	Recall
NOWJ (run3)	**0.3195**	**0.3788**	0.2762
NOWJ_(run2)	0.2865	0.2976	0.2762
NOWJ_(run1)	0.2782	0.2650	0.2928
OVGU	0.2454	0.2759	0.2210
JNLP	0.2412	0.2000	**0.3039**
AIIR	0.2368	0.2927	0.1989
CAPTAIN	0.1882	0.2547	0.1492
UA	0.1778	0.2090	0.1547

4 Task 3: Statute Law Retrieval

4.1 Task Overview

For each input legal question Q, sourced from the Japanese Bar Examination, participating systems are required to automatically retrieve from the Japanese Civil Code the complete set of articles $\{a_1, a_2, \ldots, a_n\}$ deemed relevant. Relevance is defined by the condition that an article, either individually or in combination with others, entails a Yes/No answer to question Q. Both the legal questions and the Civil Code corpus are provided in Japanese, accompanied by English translations.

Both the JNLP [10] and CAPTAIN [11] utilized BERT-base-Japanese models fine-tuned for employing ensemble strategies across their submissions. JNLP initially generated a ranked list by ensembling predictions from multiple BERT checkpoints. Their subsequent runs involved distinct LLMs2 for post-processing: one run used Mistral with prompting for final selection, another employed RankLLaMA to re-score top candidate pairs, and the third utilized Orca and Qwen for list refinement, also incorporating results from the Mistral run to improve recall. Similarly, the CAPTAIN approach

[3]https://huggingface.co/google-bert/bert-base-multilingual-cased
[4]https://huggingface.co/castorini/monot5-base-msmarco-10k
[5]https://huggingface.co/castorini/monot5-3b-msmarco-10k
[6]https://huggingface.co/deepseek-ai/DeepSeek-V3
[7]https://huggingface.co/Qwen/QwQ-32B

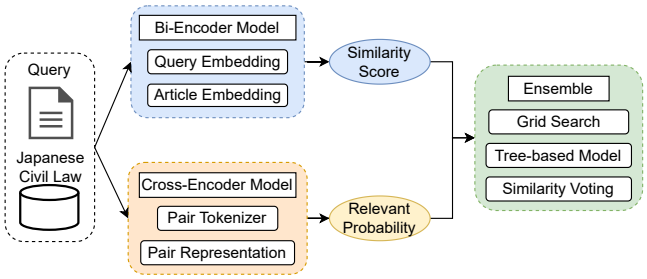


Figure 3: Overall architecture of the proposed framework for Statute Law Retrieval.

involved ensembling, initially using top BERT checkpoints. However, their refinement primarily focused on filtering results using LLM prompting with Flan T5, applied either to the BERT ensemble output or to results generated by a fine-tuned MonoT5 re-ranker. CAPTAIN's final submissions often resulted from ensembling these differently filtered result sets.

4.2 Methodology

Given the remarkable success and demonstrated semantic proficiency of contemporary open-source LLMs on diverse information retrieval and question-answering benchmarks, we posit that strategically combining these powerful pre-trained models presents a highly promising approach for addressing the specific demands of this legal retrieval task. Consequently, our methodology prioritizes the effective utilization and ensemble of existing publicly available LLM architectures instead of creating specialized models from scratch. The illustration of our proposed method is presented in Figure 3.

Core Retrieval Architecture: Our approach utilizes two main categories of models: Bi-Encoders and Cross-Encoders. For the Bi-Encoder group, models including bge, e5, stella, and NV-Embed are employed to generate independent vector representations (embeddings) for questions and legal articles separately. Subsequently, cosine similarity is calculated between these respective embedding pairs to produce an initial relevance ranking score. The Cross-Encoder group, using models like bge-reranker and gte-reranker variants, processes question-article pairs jointly, inputting them simultaneously into the model to directly compute a score reflecting their correlation or relevance.

Tree-based Ensemble: Trains a LightGBM (LGBM) model using relevance scores from multiple base retrieval models as input features. The trained LGBM then predicts a final relevance score for each query-article pair, capturing non-linear relationships between the base model outputs.

Grid Search Weighted: Identifies the top N performing base models on a development set. Performs a grid search to find optimal weights $(w_1, ..., w_N)$ that maximize performance when linearly combining the scores of these top models.

$$Score_{final} = \sum_{i=1}^{N} w_i * Score_{Mi} \quad (6)$$

Table 5: Results of Statute Law Retrieval task in COLIEE 2025.

Team	F2	Precision	Recall
JNLP_RUN1	**0.8365**	0.8037	0.8744
CAPTAIN.H2	0.8301	**0.8333**	0.8516
INFA	0.6917	0.7671	0.6826
mpnetAIIRLab	0.6674	0.3562	**0.8858**
OVGU3	0.6041	0.6347	0.6142
UIwa	0.5816	0.5856	0.5890
UA-mpnet	0.2540	0.0986	0.4361
Our runs			
NOWJ.H2	0.7702	0.7572	0.8086
NOWJ.H1	0.7311	0.7352	0.7511
NOWJ.H3	0.7069	0.7534	0.7100

Similarity-Informed Voting Ensemble: For a given test query Q, first retrieves the most similar queries Q_{sim} from historical data. Then, evaluates the performance of each base model (M_i) on Q_{sim}. Finally, aggregates the predictions of the base models for Q, weighting each M_i contribution based on its measured historical accuracy on similar queries Q_{sim}.

4.3 Experimental Setup

Our approach utilizes a diverse set of pre-trained Bi-Encoder and Cross-Encoder models from the Hugging Face Hub. Bi-Encoders are used for efficient initial candidate retrieval, while Cross-Encoders refine these results through more precise re-ranking. To enhance overall performance, we employ three ensemble strategies that combine the relevance scores produced by these models for our final submissions.

We submitted three runs based on different ensemble techniques:

- **Run 1:** Employed a LightGBM model trained on scores derived from *all* base models to predict final relevance.
- **Run 2:** Applied optimized linear weights, found via grid search, to combine scores from three selected top-performing base models: *NV-Embed-v1, multilingual-e5-large-instruct, bge-reranker-large.*
- **Run 3:** Used query similarity, calculated via bi-encoder model, *multilingual-e5-large*, to dynamically weight base model predictions based on historical performance on similar queries.

4.4 Result and Discussion

The results of our ensemble method are presented in the Table 5. Analysis of the results reveals that the Run 2 method achieved the highest efficacy F2, which is 0.7702, leveraging a strategy centered on the linear combination of scores from only three top-performing base models, with weights optimized via grid search. This suggests that the selection of high-quality input signals coupled with meticulous optimization, even when employing a straightforward methodology, proved pivotal for success, particularly with the high Recall score of 0.8086, making a substantial contribution to the F2 metric. Run 1, which employed a more sophisticated LightGBM model across all base models, yielded suboptimal performance, F2 is 0.7311, potentially attributable to the introduction of noise from less

effective models or inherent challenges in optimizing this potent ensemble model over the larger set of inputs. Run 3 proposed a more advanced concept involving dynamic weight adjustment predicated on query similarity, yet exhibited the lowest performance, F2, which is 0.7069. The inherent complexity of this approach, compounded by potential difficulties in precisely quantifying query similarity or in effectively leveraging historical performance data, seemingly impeded its practical efficacy relative to the more straightforward ensemble strategies.

5 Task 4: Legal Textual Entailment

5.1 Task Overview

This task aims to develop a Yes/No question-answering system given a legal question q and relevant articles $A = \{a_1, ..., a_n\}$, in which $n \geq 1$. The training set contains triplets $\{q, A, label\}$, in which $label \in \mathbb{R}^2$, $label = \{Y, N\}$. In the inference phase, the system is assessed by answering unseen queries. The official evaluation metric for this task is Accuracy, which is computed by the number of correct answer queries divided by the total number of queries.

Last year, most teams utilized the in-context learning capabilities of various LLMs (e.g., FlanT5, Qwen, Llama, GPT-3.5) and combined their outputs for post-processing. The CAPTAIN team[11] secured first place with data augmentation and LLM fine-tuning using LoRA. The runner-up team, JNLP[10], experimented with prompting across multiple LLMs—Qwen 14B, Mistral 7B, Flan-Alpaca, and FlanT5—employing a voting ensemble approach. Therefore, to prevent data leakage, i.e., test queries appearing in the model's training set, only open-source LLMs released before July 2024 are allowed in this year's competition.

5.2 Methodology

Inspired by recent advancements in legal text processing [4, 5, 12], the proposed framework leverages the in-context learning capability of open-source LLMs (i.e. Qwen-2, Llama-3, Mixtral) to address the problem of textual inference in legal text. The framework contains four main phases: prompt construction, LLMs deployment, answer processing, and majority voting as presented in Figure 4.

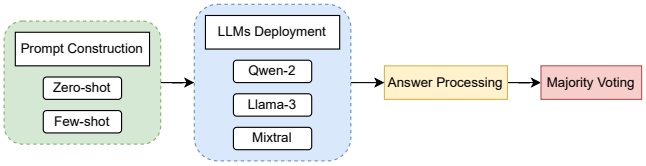


Figure 4: The overall LLMs-based framework for Legal Textual Entailment.

Prompt Construction. : We construct the prompt collection following both zero-shot and few-shot, parallel prompting. For few-shot prompting, article-shared questions are selected as examples for the LLMs. If the query shares no article with others, the examples are identified based on semantic similarity, computed by a bi-encoder architecture.

Finally, the LLM input can be defined as follows:

$$Input = [System, Inst, Premise, Hypothesis, Examples] \quad (7)$$

in which *System* is the system prompt (e.g "You are an overthinking legal assistant who always gives the best advice."), *Inst* is the step-by-step guidance to solve the task, *Examples* are similar samples extracted from the training set, *Premise* and *Hypothesis* are replaced by relevant articles and question respectively.

LLMs Deployment. Qwen-2, Llama-3, and Mixtral are chosen as the backbone models since they are the best LLMs that meet the release date constraint. In the prompt template, the placeholders premise and hypothesis are replaced with the question and article content. If multiple relevant articles exist, they are concatenated to form a single hypothesis base. Finally, LLM follows instructions and examples in the input to generate responses, which may include simple binary answers or explanations.

Answer Processing. A scanning function is designed to extract binary answers based on specific patterns (e.g., "TRUE", "FALSE"). Since responses include explanations, reasoning paths, and noise, the function first identifies the "CONCLUSION" section in the text before extracting the answer. If the extracted answer matches positive patterns, it returns "Y"; otherwise, it returns "N".

Majority Voting. Finally, the extracted answers are combined following the majority voting method to improve performance and reliability. The voting operator can be defined as follows:

$$Final\_answer = mode(Ans_{Llama}, Ans_{Qwen}, Ans_{Mix}) \quad (8)$$

5.3 Experimental Setup

The COLIEE 2025 benchmark includes 1,206 training samples and 74 testing samples, extracted from the Japanese Bar Exam. The proposed method is evaluated on the COLIEE 2023 and 2024 benchmarks containing 101 and 109 samples, respectively. The official evaluation metric for Legal Textual Entailment is accuracy.

We implemented the 4-bit quantized version of Qwen-2.5-72B-Instruct[8], Llama-3-70B-Instruct[9], and Mixtral-8x7B-Instruct-v0.1[10]. The temperature is set at 0 to ensure consistency among generations. The maximum length of responses is 800 tokens. Based on the evaluation results, we prepare three settings for submissions as follows:

- Run 1: Qwen-2 and legal few-shot prompting.
- Run 2: Majority voting of Qwen-2, Llama-3, Mixtral and legal zero-shot prompting.
- Run 3: Majority voting of Qwen-2, Llama-3, Mixtral and legal few-shot prompting.

5.4 Result and Discussion

Table 6 presents the evaluation results of the proposed method on the development set. The highest accuracy is achieved using legal few-shot prompting, scoring 0.8217 on COLIEE 2023 and 0.8623 on COLIEE 2024. However, the effectiveness of few-shot prompting is inconsistent and unreliable. Additionally, LLM performance remains similar across most experiments. To improve robustness, we combined LLM outputs using the majority voting method and submitted both few-shot and zero-shot prompting runs.

[8] https://huggingface.co/Qwen/Qwen2-72B-Instruct
[9] meta-llama/Meta-Llama-3-70B-Instruct
[10] mistralai/Mixtral-8x7B-Instruct-v0.1

Table 6: Results of the proposed method on the evaluation set.

Model	Few-shot	COLIEE 2023	COLIEE 2024
Qwen-2-72B	✓	0.7227	**0.8623**
	✗	0.7326	0.8073
Llama-3-70B	✓	0.7326	0.7522
	✗	0.7920	0.7889
Mixtral-8x7B	✓	**0.8217**	0.7339
	✗	0.7722	0.7981

Table 7: The official leaderboard of the Legal Textual Entailment task.

Team	Correct	Accuracy
KIS3	66	0.9041
CAPTAIN2	60	0.8219
JNLP002	59	0.8082
UA2	57	0.7808
KLAP.H2	56	0.7671
NOWJ.run1	54	0.7397
NOWJ.run2	54	0.7397
NOWJ.run3	54	0.7397
OVGU1	54	0.7397
RUG_V1	48	0.6575
AIIRLLaMA	44	0.6027
BaseLine	37	0.5068

The official ranking for the Legal Textual Entailment task is shown in Table 7. This year, 10 teams participated, submitting a total of 30 runs. Unexpectedly, all of our runs achieved an accuracy score of 0.7397, securing 6th place on the leaderboard. Despite their strong in-context learning abilities, LLMs still struggle with real-world challenges such as Legal Textual Entailment in COLIEE. Future work should focus on enhancing the performance and robustness of LLM-based methods through data augmentation and fine-tuning, particularly in handling complex legal reasoning and reducing inconsistencies in model predictions.

6 Pilot Task: Legal Judgment Prediction

6.1 Task Overview

This task addresses legal judgment prediction in Japanese tort cases, where plaintiffs claim that defendants' actions constitute a tort, while defendants contest these claims. It consists of two sub-tasks:

- **Tort Prediction (TP)**: Given undisputed facts (U) and arguments from plaintiffs (P) and defendants (D), the goal is to predict whether the judge affirms the tort (T, a Boolean value). TP is evaluated using *accuracy*, measuring the proportion of correctly predicted cases.
- **Rationale Extraction (RE)**: Identifies which arguments were accepted by the judge. The task predicts Boolean sequences (R^P, R^D), indicating accepted arguments in P and D. RE is evaluated using the *micro-F1 measure*.

In short, *Pilot Task* requires predicting (T, R^P, R^D) from (U, P, D).

Table 8: Statistics of data in the Legal Judgment Prediction task.

	Train set	Test set
No. samples	6508	812
Average facts / Case	1.33	1.37
Max facts / Case	134	26
Average plaintiff claims / Case	3.87	3.82
Max plaintiff claims / Case	111	130
Average defendant claims / Case	3.45	3.49
Max defendant claims / Case	86	50
No. samples without facts	3764	414
No. samples without plaintiff claims	454	33
No. samples without defendant claims	1321	106
No. samples without both claims	144	1
No. samples without facts and both claims	1	0
No. samples with facts and both claims	2215	335

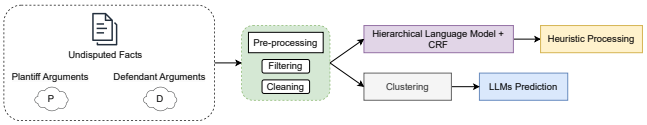


Figure 5: The overall architecture for Legal Judgment Prediction.

6.2 Methodology

A brief data analysis was conducted to understand the dataset. In this task, a training set and a test set are provided by the organizers. The dataset is provided in JSONL format, with each case consisting of undisputed facts, claims from both parties and the final court decision. In the test set, the accepted status of claims and the final court decision are hidden and require predictions.

As shown in Table 8, the training set is more detailed, containing a notably higher maximum number of facts per case, with 134 facts, compared to the test set, which only has 26. Additionally, the training set includes a larger proportion of cases with missing attributes, with 3764 cases missing undisputed facts, representing approximately 58% of the training dataset. Only 2215 cases, or 34% of the training dataset, are complete with all three attributes, which may impact the model's ability to generalize.

Figure 5 illustrates the overall architecture of our proposed framework for Legal Judgment Prediction. We adopt two different approaches. The first utilizes a hierarchical language model combined with a Conditional Random Field (CRF) layer, followed by a heuristic-based post-processing step designed to refine predictions. The second approach leverages the reasoning capabilities of advanced large language models (LLMs), which are prompted to perform judgment prediction on clustered case inputs. .

Pre-processing: The dataset contains samples that may be missing one, two, or all three attributes: undisputed facts, plaintiff claims, or defendant claims. In the training set, we remove samples that are missing two or more of these attributes and retain the rest. This results in 1246 samples being discarded due to missing two or more attributes, which accounts for about 19% of the original training

set. The remaining samples are used for further processing using our methods.

Our primary approach employs a hierarchical language model, following the Inter-Span Transformer (IST) architecture [19], which captures both word-level and span-level representations of legal texts. At the word-level encoder, we use ModernBERT-Ja-310M [11], a large variant of ModernBERT [18] trained on a high-quality corpus of Japanese and English texts. ModernBERT [18], an improved version of BERT[3], integrates local and global attention mechanisms to efficiently handle long sequences while maintaining computational efficiency. It also incorporates Rotary Positional Embeddings (RoPE) [16], further improving performance across various NLP tasks. Claims and facts of each tort are first processed through the word-level encoder to generate contextual embeddings. The span-level Transformer [17] then captures interactions among claims, where each claim representation is enriched with fact, party-type, and positional embeddings. The TP task is framed as a binary classification problem, while RE is treated as a sequence labeling task. To model dependencies between closely related claims, particularly those from the same side, we incorporate a Conditional Random Field (CRF) layer [6], ensuring that claim predictions remain consistent within a party's argument.

To refine predictions, we introduce post-processing heuristics for both sub-tasks. In TP, if one party has both more accepted and fewer unaccepted claims than the other, the predicted decision is reversed to favor the dominant side. Once the final TP decision is established, RE predictions are refined by ensuring consistency within each party: if the number of accepted claims exceeds unaccepted ones by at least x times (optimized via grid search), the unaccepted claims are adjusted to accepted, and vice versa.

To promote explainability, we propose a clustering-based approach that organizes claims into semantically coherent subarguments. First, claims from both sides are embedded using the paraphrase-multilingual-MiniLM-L12-v2 model[12], a pretrained sentence transformer [14]. The embeddings are then clustered using HDBSCAN [1], a density-based algorithm that groups similar claims while leaving outliers unclustered. Undisputed facts are incorporated into these clusters to provide additional context. Each subargument is independently assessed by DeepSeek-V3[13], a large language model, to generate predictions for both TP and RE. The final TP decision of the tort is determined through a voting mechanism, where the side with the most winning subarguments prevails. For RE, unclustered claims inherit the majority stance of their respective party to maintain consistency. The entire case is treated as a single cluster if no clusters are formed.

6.3 Experimental Setup

Our primary approach employed a hierarchical transformer architecture using ModernBERT-Ja-310M as the word-level encoder, followed by a span-level Transformer with 8 layers and 8 attention heads. The model was trained for 18 epochs using the AdamW[8] optimizer with a learning rate of $6e^{-6}$ and a linear warmup schedule over 10% of the total training steps. We combined binary cross-entropy loss for TP and CRF loss for RE, weighted by a factor $\alpha = 0.4$. Each input case was processed with a maximum of 64 claims, where individual claims were truncated to 64 tokens and the aggregated undisputed facts were limited to 512 tokens. The decision threshold for TP classification was optimized via grid search on the validation set with an optimal value of 0.3838. The RE post-processing threshold was also determined through grid search, with $x = 2$ selected as the optimal value. For our clustering-based approach, we generated claim embeddings using *paraphrase-multilingual-MiniLM-L12-v2* and performed HDBSCAN clustering with at least two claims per cluster, using cosine similarity as the distance metric.

We submitted three settings as follows:

- **Run 1**: We utilize the hierarchical language model architecture with ModernBERT-Ja-310M and a span-level Transformer, predicting TP as binary classification and RE with a CRF layer, without additional post-processing.
- **Run 2**: We enhance Run 1 by incorporating the outlined post-processing heuristics.
- **Run 3**: We implement the proposed clustering-based method, embedding claims with *paraphrase-multilingual-MiniLM-L12-v2*, clustering via HDBSCAN, and assessing clusters with DeepSeek-V3 for TP and RE predictions, followed by our proposed voting mechanism to determine the final TP decision.

Table 9: The official leaderboard of the Legal Judgment Prediction task.

Team	Accuracy
CAPTAIN	**76.5%**
KIS	71.3%
NOWJ (run 2)	67.1%
omega	66.6%
NOWJ (run 1)	63.8%
NOWJ (run 3)	59.7%
OVGU	55.3%
LLNTU	54.1%

(a) Tort Prediction

Team	F1 score
KIS	**71.2%**
CAPTAIN	70.6%
NOWJ (run 2)	69.2%
omega	69.1%
LLNTU	68.2%
NOWJ (run 1)	68.1%
OVGU	65.7%
NOWJ (run 3)	55.9%

(b) Rationale Extraction

6.4 Result and Discussion

Our heuristic post-processing step in run 2 yielded notable improvements, achieving 67.1% accuracy in TP and 69.2% F1 score in RE, representing gains of 3.3 and 1.1 percentage points, respectively, over run 1. These improvements validate our hypothesis that maintaining consistency between claim patterns and final decisions can enhance model performance. Our clustering-based approach (run 3) underperformed with 59.7% TP accuracy and 55.9% RE F1 score, suggesting that while semantic clustering provides explainability, it may oversimplify the complex relationships in legal argumentation. This performance gap highlights the importance of structural dependencies in legal reasoning. Consequently, our best approach (run 2) secured third place in both sub-tasks, with a 9.4 percentage point gap in TP accuracy and a 2.0 percentage point gap in RE F1

[11] https://huggingface.co/sbintuitions/modernbert-ja-310m
[12] https://huggingface.co/sentence-transformers/paraphrase-multilingual-MiniLM-L12-v2
[13] https://huggingface.co/deepseek-ai/DeepSeek-V3

score compared to the top-performing team. Future work could consider alternative model architectures, integrating legal-reasoning prompts with LLMs, and developing structured clustering methods to enhance both performance and explainability.

7 Conclusion

In COLIEE 2025, the NOWJ team successfully developed and deployed innovative methodologies across all five competition tasks, notably securing the highest performance in Task 2 (Legal Case Entailment) by integrating hierarchical retrieval methods with contextualized re-ranking using Large Language Models (QwQ-32B and DeepSeek-V3). Our multi-stage ensemble framework, combining embedding-based models and advanced LLM-based techniques, consistently showed effectiveness in managing complex legal reasoning and retrieval scenarios. Results demonstrate the clear advantage of combining embedding precision, transformer-based summarization, and deep contextual reasoning through generative language models. Future research should continue exploring model fine-tuning, improved ensemble strategies, and enhanced prompt engineering to further advance the interpretability, accuracy, and generalizability of legal information systems.

References

[1] Ricardo J. G. B. Campello, Davoud Moulavi, and Joerg Sander. 2013. Density-Based Clustering Based on Hierarchical Density Estimates. In *Advances in Knowledge Discovery and Data Mining*, Jian Pei, Vincent S. Tseng, Longbing Cao, Hiroshi Motoda, and Guandong Xu (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 160–172.

[2] 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, Singapore, 156–166.

[3] 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

[4] 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

[5] 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.

[6] John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *Proceedings of the Eighteenth International Conference on Machine Learning (ICML '01)*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 282–289.

[7] 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 *New Frontiers in Artificial Intelligence: JSAI International Symposium on Artificial Intelligence, JSAI-IsAI 2024, Hamamatsu, Japan, May 28–29, 2024, Proceedings* (Hamamatsu, Japan). Springer-Verlag, Berlin, Heidelberg, 212–227. doi:10.1007/978-981-97-3076-6_15

[8] Ilya Loshchilov and Frank Hutter. 2019. Decoupled Weight Decay Regularization. arXiv:1711.05101 [cs.LG] https://arxiv.org/abs/1711.05101

[9] Julián Moreno Schneider, Georg Rehm, Elena Montiel-Ponsoda, Víctor Rodríguez-Doncel, Patricia Martín-Chozas, María Navas-Loro, Martin Kaltenböck, Artem Revenko, Sotirios Karampatakis, Christian Sageder, Jorge Gracia, Filippo Maganza, Ilan Kernerman, Dorielle Lonke, Andis Lagzdins, Julia Bosque Gil, Pieter Verhoeven, Elsa Gomez Diaz, and Pascual Boil Ballesteros. 2022. Lynx: A knowledge-based AI service platform for content processing, enrichment and analysis for the legal domain. *Information Systems* 106 (2022), 101966. doi:10.1016/j.is.2021.101966

[10] 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 *New Frontiers in Artificial Intelligence*, Toyotaro Suzumura and Mayumi Bono (Eds.). Springer Nature Singapore, Singapore, 167–182.

[11] 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*, Toyotaro Suzumura and Mayumi Bono (Eds.). Springer Nature Singapore, Singapore, 125–139.

[12] 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*, Toyotaro Suzumura and Mayumi Bono (Eds.). Springer Nature Singapore, Singapore, 183–199.

[13] 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*, Toyotaro Suzumura and Mayumi Bono (Eds.). Springer Nature Singapore, Singapore, 200–211.

[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)*, Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (Eds.). Association for Computational Linguistics, Hong Kong, China, 3982–3992. doi:10.18653/v1/D19-1410

[15] Dong Shu, Haoran Zhao, Xukun Liu, David Demeter, Mengnan Du, and Yongfeng Zhang. 2024. LawLLM: Law Large Language Model for the US Legal System. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management* (Boise, ID, USA) *(CIKM '24)*. Association for Computing Machinery, New York, NY, USA, 4882–4889. doi:10.1145/3627673.3680020

[16] Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. 2024. RoFormer: Enhanced transformer with Rotary Position Embedding. *Neurocomputing* 568 (2024), 127063. doi:10.1016/j.neucom.2023.127063

[17] 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

[18] Benjamin Warner, Antoine Chaffin, Benjamin Clavié, Orion Weller, Oskar Hallström, Said Taghadouini, Alexis Gallagher, Raja Biswas, Faisal Ladhak, Tom Aarsen, Nathan Cooper, Griffin Adams, Jeremy Howard, and Iacopo Poli. 2024. Smarter, Better, Faster, Longer: A Modern Bidirectional Encoder for Fast, Memory Efficient, and Long Context Finetuning and Inference. arXiv:2412.13663 [cs.CL] https://arxiv.org/abs/2412.13663

[19] 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* (May 2024). doi:10.1007/s10506-024-09402-0

[20] 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*, Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault (Eds.). Association for Computational Linguistics, Online, 5218–5230. doi:10.18653/v1/2020.acl-main.466

Received 7 April 2025; accepted 5 May 2025

JNLP at COLIEE 2025: Hybrid Large Language Model-based Framework for Legal Information Retrieval and Entailment

Hai Nguyen*
Hiep Nguyen*
Trang Pham*
nthehai01@jaist.ac.jp
hiep.nkv@jaist.ac.jp
trangpna@jaist.ac.jp
Japan Advanced Institute of Science and Technology
Nomi, Ishikawa, Japan

Minh Nguyen
minh.nn@jaist.ac.jp
Japan Advanced Institute of Science and Technology
Nomi, Ishikawa, Japan

An Trieu
antrieu@jaist.ac.jp
Japan Advanced Institute of Science and Technology
Nomi, Ishikawa, Japan

Dinh-Truong Do
truongdo@jaist.ac.jp
Japan Advanced Institute of Science and Technology
Nomi, Ishikawa, Japan

Nguyen-Khang Le
lnkhang@jaist.ac.jp
Japan Advanced Institute of Science and Technology
Nomi, Ishikawa, Japan

Le-Minh Nguyen+
nguyenml@jaist.ac.jp
Japan Advanced Institute of Science and Technology
Nomi, Ishikawa, Japan

Abstract

This paper presents the JNLP team's approaches for the COLIEE 2025 competition, addressing all four legal information processing tasks: case law retrieval, case law entailment, statute law retrieval, and statute law entailment. Our systems leverage a hybrid framework that synergistically combines classical information retrieval (IR) pipelines, fine-tuned Transformer-based models, and instruction-tuned Large Language Models (LLMs) for deep legal reasoning. For case law retrieval (Task 1), we enhance a proposition-based ranking model by integrating lexical and structural-semantic features. For case law entailment (Task 2), we adopt a two-stage pipeline: we fine-tune re-rankers with hard-negative sampling and refine predictions using few-shot prompted LLMs. In statute law retrieval (Task 3), we implement a three-stage pipeline consisting of embedding-based pre-retrieval, LoRA/QLoRA-based fine-tuning, and model ensembling. For statute law entailment (Task 4), we explore zero-shot, few-shot, and reasoning ensemble prompting using models like Qwen2-72B to generate well-justified yes/no answers. Experimental results show that our methods achieve top-tier performance across multiple tasks in the official COLIEE 2025 evaluation. Our findings highlight the practicality and effectiveness of integrating lightweight IR models with large-scale LLMs for high-stakes legal NLP applications.

CCS Concepts

• **Information systems** → *Expert systems.*

COLIEE 2025, Chicago, USA

Keywords

Legal Information Retrieval and Entailment, Natural Language Processing in Law, Large Language Model for Legal, COLIEE 2025

ACM Reference Format:
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 COLIEE 2025 workshop, June 20, 2025, Chicago, USA.* ACM, New York, NY, USA, 10 pages.

1 Introduction

The Competition on Legal Information Extraction/Entailment (COLIEE) is an annual benchmark that drives research in legal natural language processing by defining challenging tasks on case law and statute law [23, 27]. Four tasks are featured: legal case retrieval (Task 1), which asks systems to find prior cases supporting a new case; legal case entailment (Task 2), which requires identifying specific paragraph(s) in a precedent case that entails the decision of a query case; statute law retrieval (Task 3), which involves retrieving relevant statutory articles for a yes/no legal question; and statute law entailment (Task 4), a yes/no question answering task determining if a given statute resolves the query. These tasks address real-world needs in the legal domain: retrieving on-point precedents is essential for attorneys and judges to ensure consistent, well-founded arguments, and accurately answering legal questions against statutes can assist in legal decision support (e.g., automating parts of the bar exam). However, automating these tasks is difficult because legal documents have intricate structures, domain-specific terminology, and often implicit reasoning steps [22].

Large Language Models (LLMs) have recently opened a new frontier for legal NLP [11, 23]. In the last two years, COLIEE participants began to harness the few-shot reasoning capabilities of generative LLMs. Notably, Nguyen et al. [24] (CAPTAIN team) demonstrated

*Equal Contribution
+Corresponding author

that prompting an instruction-tuned LLM can capture the complex relations between a query case and statutes, helping win the statute entailment task in COLIEE 2024. Similarly, other teams integrated LLMs into their pipelines: for instance, Nighojkar et al. [25] used a Flan-T5-XXL model with carefully designed prompts and voting to achieve strong performance in statute law QA. These successes highlight the growing role of foundation models in legal reasoning tasks, as LLMs can parse and summarize legal texts, then apply general reasoning to determine relevance or entailment.

In this paper, we present the JNLP team's approach for COLIEE 2025, addressing all four tasks with tailored methodologies that combine the strengths of classical retrieval systems, fine-tuned transformers, and instruction-following large language models (LLMs). For case law retrieval (Task 1), we extend a proposition-based similarity ranking framework with enhanced lexical and semantic features, incorporating BM25 and SAILER scores [14]. For case law entailment (Task 2), we employ hard-negative sampling to fine-tune re-ranking models and apply a two-stage filtering pipeline that leverages few-shot prompting with LLMs. In statute law retrieval (Task 3), we construct a multi-stage pipeline combining dense retrieval, gradient-boosted relevance classification, RankLLama re-ranking, and LoRA/QLoRA-based fine-tuning. For statute law entailment (Task 4), we design a reasoning-augmented prompting framework using zero-shot, few-shot, and reasoning ensemble strategies with models like Qwen2-72B [1]. Our system achieves top-tier performance across tasks, demonstrating the effectiveness of hybrid pipelines that strategically integrate efficient retrieval techniques with powerful LLM-based legal reasoning. The following sections review related work and outline how our approach advances the state of the art in legal information retrieval and entailment.

2 Related Work

2.1 Case Law Retrieval and Entailment

Over recent years, legal case retrieval and entailment have evolved significantly through the adoption of neural ranking architectures. In COLIEE 2022, the UA team [28] leveraged paragraph-level embeddings to assess semantic similarity between queries and candidate cases. Other teams incorporated external knowledge [9] or fused lexical matching with semantic understanding [5], highlighting the trend toward richer representations. The 2023 competition introduced more advanced pipelines. THUIR [16] developed a legal-domain encoder-decoder model, while JNLP [4] and IITDLI [7] adopted a two-step strategy: initial document retrieval using classical models (e.g., BM25), followed by re-ranking using fine-tuned language models or LLMs. In COLIEE 2024, the TQM team [15] advanced this further by combining lexical and dense retrieval features—along with simple heuristics like document length—and feeding them into a learning-to-rank framework with customized preprocessing and postprocessing filters.

In Task 2 (case law entailment), ensemble techniques have become dominant. In COLIEE 2022, NM [30] topped the leaderboard by integrating scores from a fine-tuned MonoT5 and a zero-shot version. JNLP [5] achieved second place by enhancing retrieval with abstract meaning representation (AMR) and combining it with LegalBERT and BM25-based signals. The CAPTAIN system [22] refined this in 2023 by applying hard-negative mining with MonoT5 and aggregating multiple model checkpoints. THUIR [16] implemented a multi-stage pipeline—starting with BM25 and QLD, followed by contrastive learning and ensemble integration. In COLIEE 2024, AMHR [25] achieved top performance using a MonoT5 model fine-tuned on the MSMARCO dataset with hard negatives mined by both BM25 and previous MonoT5 predictions.

2.2 Statute Law Retrieval and Entailment

Task 3 in COLIEE focuses on retrieving relevant Japanese Civil Code articles. In 2022, HUKB [35] fine-tuned BERT on legal texts and applied data augmentation to better align article and query representations. CAPTAIN [22] led the 2023 challenge with an ensemble of BERT-Japanese and MonoT5, combined with data filtering to improve retrieval precision. In 2024, the same team [24] extended this by introducing prompting with Flan-T5 to filter low-quality outputs and enforce consistency across retrieved statutes.

For Task 4 (statute law entailment), early systems focused on binary classification using pre-trained language models. JNLP [5] benchmarked models like ELECTRA, RoBERTa, and LegalBERT, and introduced negation-aware data augmentation. LLNTU [19] proposed a novel input formatting strategy using concatenation and a longest uncommon subsequence metric. By 2023, generative LLMs emerged: JNLP [4] employed zero-shot prompting with Flan-T5 and Alpaca-T5 to generate direct answers from query-article pairs. The 2024 champion, CAPTAIN [24], used LLM-based summarization of statute articles with filtering heuristics to guide Flan-T5-XXL in producing reliable entailment predictions.

3 Problem Formalization

3.1 Task 1

Legal Case Retrieval is a foundational task in legal practice, enabling practitioners to identify previous cases that support a given legal case. This capability is significant for legal professionals in collecting well-supported and persuasive arguments, and equally important for judges and courts in ensuring that decisions are consistent, fair, and grounded in established case law. A robust retrieval system not only promotes transparency but also enhances the accuracy and efficiency of legal proceedings.

Formally, given a query legal document q, the objective is to retrieve a set of relevant precedent cases, or notice cases, denoted as $R = \{r_1, r_2, ...\}$, from a shared pool of candidate cases $D = \{d_1, d_2, ...\}$. The candidate set D remains the same for all queries, while the number of relevant notice cases for each query may vary.

3.2 Task 2

The goal of this task is to automatically determine which paragraph(s) in a noticed (cited) case text entail a given decision extracted from a new query case. Let Q denote a query, which corresponds to the "decision" of a new legal case. Although referred to as a "decision", Q is a textual segment representing a judicial conclusion rather than a final ruling. Let N denote the noticed case that contains multiple paragraphs. Denote the set of paragraphs within N as $P = \{p_1, p_2, ..., p_m\}$, where each p_i is a unique paragraph.

In the COLIEE settings, a legal textual entailment system aims to identify the specific set of paragraphs $\tilde{P} \subset P$ that entails Q. Entailment in this context indicates that $p_i \in \tilde{P}$ semantically justifies or supports the proposition stated in Q. Although a noticed case might contain multiple paragraphs that partially overlap with Q, the dataset identifies a single paragraph (or occasionally a small set of paragraphs) that is deemed central to entailing the decision. The principal objective is to maximize the accuracy of identifying the paragraph p_i in the noticed case that correctly entails the decision Q. In practice, this task serves as a simplified version of judicial decision prediction, focusing on the step where a judge's conclusion Q can be traced back to specific paragraph(s) in a cited precedent (N).

3.3 Task 3

Automatically identifying and retrieving the most relevant legal provisions for a given query becomes essential in the legal domain to ensure accurate and efficient access to statutory information. An article A is considered "Relevant" to a query Q if the information contained within A allows this query Q to be answered with a simple "Yes" or "No", meaning that this article logically entails the query's meaning. A major challenge arises from the necessity to evaluate a large set of legal articles $\{A_1, A_2, \ldots, A_n\}$, where n can be in the hundreds. Besides, certain queries Q may require information aggregated from multiple articles, further introducing additional complexity to the retrieval process.

Under these conditions, this task involves building systems capable of legal reasoning and understanding the entailment relationship. Assuming a scenario with one query article and a set of legal articles, these systems must predict which candidate articles from the corpus entail the decision of the query case, indicating a supporting legal precedent. As a result, the output is a list of articles that entail the query case, reflecting a valid legal precedent relationship.

3.4 Task 4

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.

To address this, we designed 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 $Y \in \{$"Yes", "No"$\}$. The objective is to maximize the likelihood of generating a correct answer, formalized as maximizing the conditional probability $P(Y \mid X)$.

Top-k	Precision	Recall	F1-score
100	0.030	0.7663	0.058
200	0.017	0.8547	0.033

Table 1: Performance of BM25 in retrieving top-k candidate cases.

4 Methodology

4.1 Task 1

In this task, we develop a framework that based on the proposed model of team UMNLP [6] in Task 1 of the COLIEE 2024 competition. The UMNLP team proposed a pairwise similarity ranking framework by training a feedforward neural network to perform a binary classification task, based on a multitude of features from each query-candidate case pair. This approach first extracts statistical and semantic features from each case after performing preprocessing tasks, then comparing these case-level features to obtain a set of numerical features for each query-candidate case pair.

Rather than utilizing the full content of legal documents, the features are derived from propositions—statements that encapsulate the claims supported by citations. These propositions are concise, third-person summaries presented in an objective tone. Their extraction is guided by specific placeholders (e.g., "FRAGMENT_SUPPRESS") found within the query documents. The authors employed a custom dataset along with sequence-to-sequence Transformers models to identify and generate these propositions. Subsequently, a fully connected feedforward neural network was trained to perform a binary classification task, assigning a label of 1 to notice cases and 0 to non-relevant cases. This trained model was then applied to test case pairs, ranking them based on the likelihood of being a notice case, followed by heuristic filtering to produce the final predicted results.

To further improve the performance of the framework, we extend the feature set with two new features: BM25 scores and SAILER [14] scores. BM25 is a ranking function that is used in information retrieval to estimate the relevance of a document to a given search query while the SAILER model incorporates the structural information inherent in legal documents and emphasizes key legal elements, mirroring the approach of legal professionals when reviewing cases. With applying two features, the model can capture lexical-matching information from BM25 scores and the semantic and structural information from the SAILER scores. Moreover, unlike the original framework, which directly applies to all the candidate documents, we first filter the relevant candidates based on BM25 scores due to an observation that by filtering with top candidates based on BM25 scores, the system can achieve competitive recall scores, creating a good foundation for the following re-ranking steps. The table 1 shows the performance of BM25 on Task 1 COLIEE 2024 test dataset. By retrieving the top 100 candidates, BM25 achieves a recall of 76%, effectively reducing the search space for the subsequent re-ranking process.

Figure 1 illustrates an overview of our system. In summary, our proposed framework includes the following components:

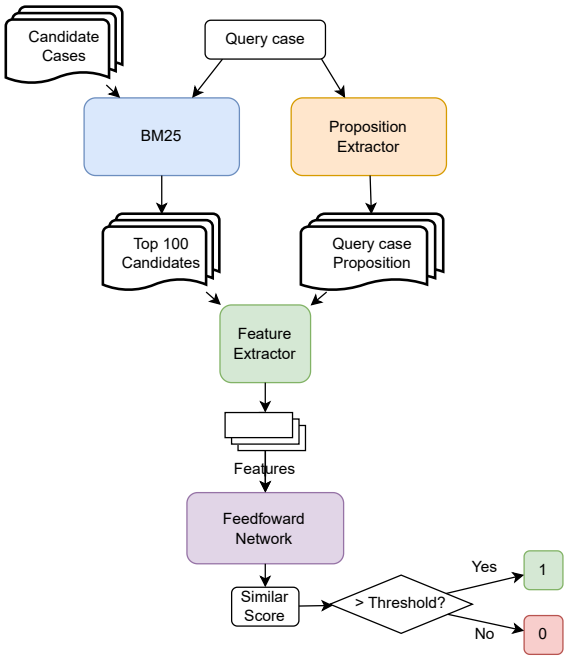


Figure 1: Task 1 System Overview

- ***Retrieval*** We rank the candidate cases via BM25 scores and select top candidates
- ***Proposition Extraction*** We re-produced the proposition extraction model in the work of the UMNLP team by fine-tuning a T5-transformers sequence-to-sequence model. Then we extract the proposition for each query legal case.
- ***Re-ranking*** We train a re-ranker as a binary classifier to assess the relevance between a query case's proposition and a candidate case. During training, feature vectors are extracted for each proposition–candidate pair and input into a feed-forward neural network to predict whether the pair is relevant. The training data is divided into training and validation sets, and the model is optimized using early stopping based on F1-score performance on the validation set. A relevance threshold is predefined: if the model's output probability exceeds this threshold, the pair is labeled as relevant (1); otherwise, it is labeled as non-relevant (0). The optimal threshold is selected via grid search on the validation set.

4.2 Task 2

Three distinct runs (jnlp_001, jnlp_002, jnlp_003) were submitted for Task 2: Legal Case Entailment. All runs utilized the official COLIEE dataset, and the methodology emphasis on two key components: (1) fine-tuning model with hard negative sampling and (2) two-stage filtering with LLM.

4.2.1 Fine-Tuning with Hard Negative Sampling. Figure 2 depicts the fine-tuning process for our models. Given a set of triplets

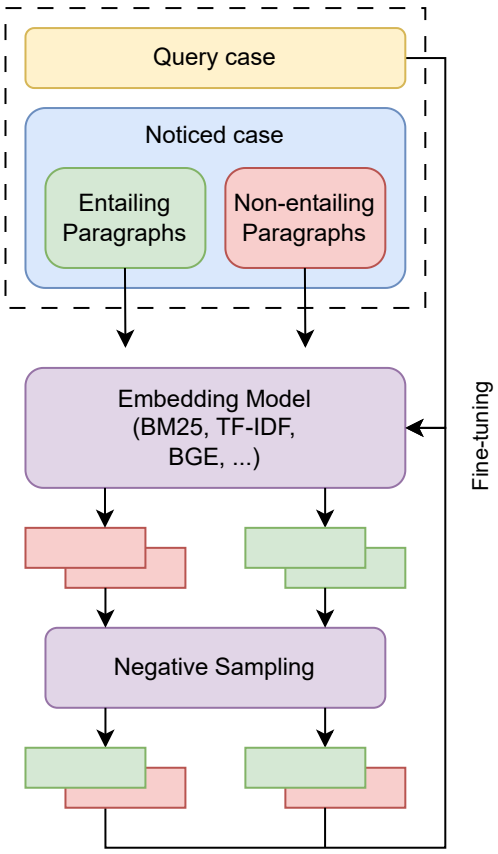


Figure 2: Hard-negative Sample Fine-tuning

(Q, N, p^*), where Q denotes the query case, N is the noticed case, and p^* are the gold paragraphs that support the query. During the model fine-tuning process, hard negative samples were employed. Specifically, for each query, the negative samples were drawn from paragraphs in N that were non-entailing yet exhibited contextual similarity to the gold paragraphs. The context similarity is based on an intermediate representation (embedding), which can be the model itself if the model is an embedding model or an external representation such as BM25, TF-IDF, NVEmbed, etc. By using these closely matched (but incorrect) paragraphs as negative instances, the fine-tuning process compelled the models to focus on fine-grained semantic cues that separate valid from invalid entailments.

4.2.2 Two-Stage Filter with Large Language Model. Our third submission integrated a two-stage filtering process, combining two models to further refine the set of candidate paragraphs. First, a fine-tuned re-ranking model applied a relatively lenient threshold to filter out obviously irrelevant paragraphs, thereby retaining plausible candidates. Next, an LLM applied a second filtering step. In this step, we explicitly designed few-shot prompts (figure 3) to query a large instruction-tuned model, enabling it to determine if the candidate paragraph entailed the query. By incorporating this multi-model approach, the workflow aimed to reduce false

Few-shot filtering prompt.

Given a query and a document, determine if the document is really relevant to the query. Here are some examples:
Example 1:
Query: ...
Relevant document: ...

Example 2:
Query: ...
Relevant document: ...

Example 3:
Query: ...
Relevant document: ...

Following the above examples, answer the question.
Query: ...
Document: ...

Question: does the document really relevant to query? Please provide the answer as 'yes' or 'no'.

Figure 3: Few-shot filtering prompt.

negatives through a permissive first filter while improving precision through the final re-ranking performed by the large language model.

4.3 Task 3

Inspired by prior works [23, 24], this work adopts a multi-stage retrieval pipeline designed to enhance both effectiveness and efficiency. In particular, we emphasize the utilization of large language models as the foundation for constructing this pipeline. Our proposed framework consists of three distinct stages: pre-retrieval, model fine-tuning, and result ensemble. Figure 4 describes the overview of our pipeline.

Stage 1: Pre-retrieval. Directly identifying relevant articles from a large legal corpus poses a significant challenge due to the expansive search space. One effective strategy for addressing this issue is to remove the articles that have low relevance levels to the query. Motivated by this approach, we establish a set of filtering steps aimed at eliminating the least important articles and retaining only top-k (e.g. $k = 50$) candidates. This methodology ensures a high recall rate while substantially reducing the volume of data for subsequent stages. Experimental results indicate that our proposed method can archive recall rates higher than 95% while narrowing down the search space for the next phases of the pipeline.

Building upon the previous work [29], we adopt a binary classification approach using Gradient Boosting [10] applied on multi-dimensional text embeddings to filter the top 100 most relevant articles for a given query. Specifically, we use BGE-M3 [1] [21] to extract the dense text vectors for both the articles and the query. We then compute the L1 distance between the query vectors and

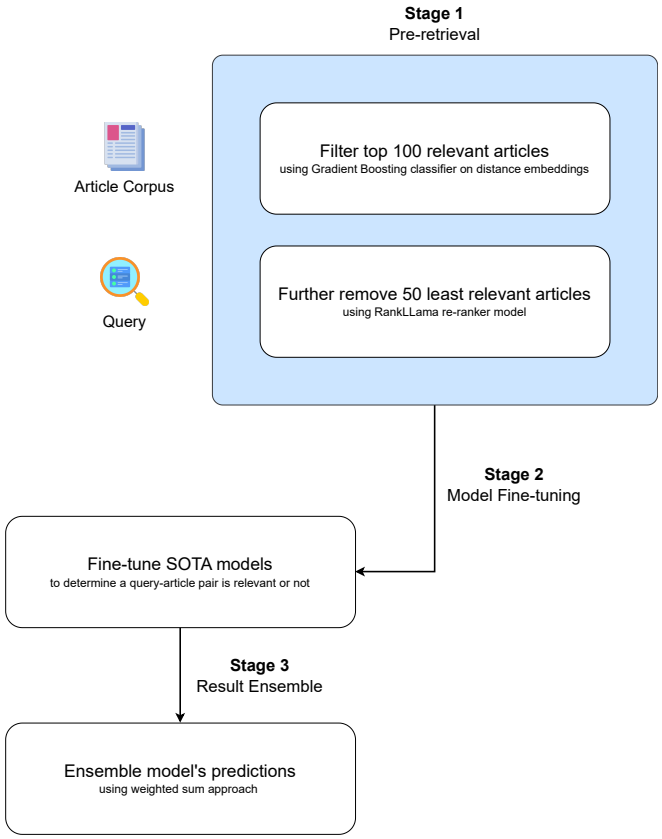


Figure 4: A multi-stage retrieval pipeline to extract relevant articles for a given query input in Task 3.

all of the provided articles in the corpus. These distance vectors are subsequently organized into bins, with a maximum of 76 bins. A gradient-boosting classifier is then trained on these binned distance features to determine the relevance of query-article pairs. Due to the inherent class imbalance phenomenon, where the irrelevant (negative) pairs dominate relevant (positive) ones, we apply oversampling to the relevant pairs by replicating them 300 times to make the dataset more balanced. Finally, we keep only 100 articles that have the highest predicted probability of being relevant to a query.

Following the initial filtering step that identifies the top 100 relevant articles, we further refine the selection by removing 50 articles contributing the least to the recall score. In this step, we leverage RankLLama [2] [20], a model originally fine-tuned for passage re-ranking, which demonstrated the effectiveness in the previous competition [23]. RankLLama is employed to assign the relevance scores for the query-article pairs. Based on these scores, only the top 50 articles with the highest scores are retained for the subsequent stages. Experiments show that this model can effectively discard the less relevant articles produced from the previous step without a catastrophic performance drop in overall performance.

[1] https://huggingface.co/BAAI/bge-m3

[2] https://huggingface.co/castorini/rankllama-v1-7b-lora-passage

Stage 2: Model Fine-tuning. In this stage, we fine-tune a batch of state-of-the-art large language models to perform binary classification, determining whether a given query-article pair is relevant or not. To enable the fine-tuning process of these billion-parameter models while minimizing computational overhead, we employ the use of parameter-efficient adapter fine-tuning techniques, namely LoRA [12] and QLoRA [8].

The dataset used for this stage is derived from the output of Stage 1, consisting of approximately 50 candidate articles per query. To achieve more robust fine-tuned models, we augment this dataset by including all ground-truth relevant articles for each query, even though they may be discarded from the previous stage in our pipeline. Due to the nature of the task, the resulting dataset exhibits a significant class imbalance: on average, each query is associated with only one or two positive samples, compared to roughly 50 negative samples. To alleviate this problem, we simply apply the up-sampling strategy again, replicating each positive sample three times for a query instance while maintaining the original number of negative samples. This approach helps to improve the model's ability to correctly identify relevant documents.

Stage 3: Result Ensemble. To enhance the robustness and overall performance of the final result, we employ an ensemble strategy that combines the outputs of the models fine-tuned in the previous stage. Particularly, a weighted sum approach is utilized to aggregate the individual model predictions. The optimal weights for these models are automatically tuned using the Optuna [3] framework.

4.4 Task 4

In this task, for each legal query, we begin by constructing a set of input sequences that combine the query with its relevant legal articles. These sequences are then provided to a Large Language Model (LLM), which is prompted to generate an answer accompanied by an explanation. Finally, we collect these responses and prompt the LLM again to determine the final binary answer — either `"Yes"` or `"No"` — based on the reasoning provided. For the answer generation phase, we apply three different prompting strategies: **Zero-shot**, **Few-shot**, and **Reasoning Ensemble**.

Zero-shot: In this setting, we construct input sequences by pairing each legal query with its relevant articles, with different prompts. The LLM is then asked to produce an answer along with an explanation for each sequence. These generated responses are stored for both the final decision step and for use as reference samples in the other settings.

Few-shot: In this setting, we guide the LLM using k example samples drawn from the **Zero-shot** responses. These samples help steer the LLM toward generating more accurate answers for new queries. This process involves four steps, as in the figure 5:

(1) Select correct answers and corresponding explanations from the Zero-shot setting to construct an example pool.
(2) Use Dense Passage Retrieval to find k examples whose legal queries are most similar to the current query.
(3) Construct an input sequence by combining the retrieved examples (queries, relevant articles, answers, and explanations) with the current query and its relevant articles.
(4) Feed the final input sequence into the LLM and collect the generated response.

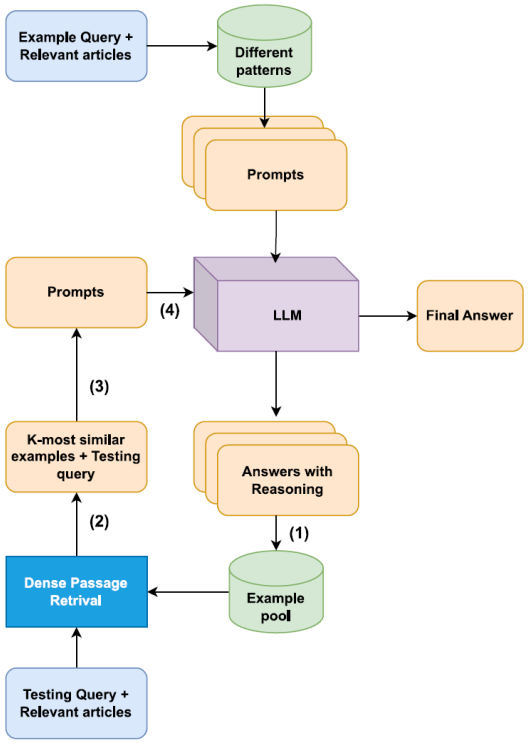


Figure 5: Overview of few-shot setting.

Reasoning Ensemble: In this setting, we aggregate the answers and explanations from k different Zero-shot input sequences for a given legal query. These are combined into a single input sequence, which is then passed to the LLM. The model is asked to synthesize the collected reasoning and generate a single, final answer.

5 Experimental Results

5.1 Task 1

We use the dataset in Task 1 of the COLIEE 2024 Competition. We utilize the train dataset as our training dataset and the test dataset is used as our evaluating dataset. For baselines, we apply the following models

- **BM25**: Utilizing BM25 scores with Python implementation from pyserini [18]
- **Histogram-BGE**: A framework that uses the BGE model [21] to obtain the histogram of similarities between the paragraphs of the query document and the candidate document. We use the BGE model to embed each paragraph in a legal document, and then for each pair of a query document and a candidate document, we calculate the cosine similarity among the paragraphs of the above two documents. From the obtained cosine similarities, we group the scores into a

Model	Precision	Recall	F1-score
BM25	0.1496	0.2702	0.1926
Histogram-BGE	0.1533	0.3924	0.2204
UMNLP	0.3786	0.4373	0.4058
JNLP&fe1 (ours)	0.4339	0.4078	0.4205
JNLPr&fe1 (ours)	0.4336	0.4135	0.4233
JNLPr&fe2 (ours)	0.4310	0.4238	0.4274

Table 2: Evaluation results on the dataset.

10-bins histogram and use this as features for a binary classifier to predict whether the candidate document is similar to the query document.

- **UMNLP**: We use the reported result of the UMNLP [6] team in Task 1 of the COLIEE 2024 Competition.

In task 1, *precision*, *recall* and *F1-score* are used to evaluate performance. In this evaluation, micro-average is prefer rather than macro-average.

$$Precision = \frac{\text{Correctly retrieved cases(paragraphs) for all queries}}{\text{Retrieved cases(paragraphs) for all queries}}$$

$$Recall = \frac{\text{Correctly retrieved cases(paragraphs) for all queries}}{\text{Relevant cases(paragraphs) for all queries}}$$

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

For Task 1 of the COLIEE 2025 Competition, we submited three runs

- **jnlpr&fe1**: Apply BM25 to filter candidates first with top-100 candidates, then applying the our customized UMNLP framework.
- **jnlpr&fe2**: Apply BM25 to filter candidates first with top-200 candidates, then applying the our customized UMNLP framework/
- **jnlpfe1**: Apply our customized UMNLP framework directly.

The table 2 illustrates the performance of the models on the evaluating dataset. Our proposed framework, enhanced with additional features, outperforms the baseline methods. Incorporating a candidate filtering step based on BM25 scores yields an improvement over applying the framework without prior filtering. Our framework claims the first and second ranks in Task 1 of the COLIEE 2025 Competition.

5.2 Task 2

We used the highest ranking in the previous year COLIEE 2024 competition as our baseline [AMHR COLIEE 2024 entry: legal entailment and retrieval]. This group employed 2 methodologies for this task. The best performed approach entailed finetuning a monoT5 model pretrained on the MSMARCO dataset, supplemented with hard negative mining through BM25 and another monoT5 model. From the resulting predictions, the top two paragraphs were selected if their ratio of similarity score fell below a grid searched threshold of 6.619; otherwise, only the highest-scoring paragraphs was chosen. The training data consists of triplets (Q, N, p^*), where

Team	File	F1	Precision	Recall
JNLP	**jnlpr&fe2.txt**	**0.3353**	**0.3042**	**0.3735**
JNLP	**jnlpr&fe1.txt**	**0.3267**	**0.2945**	**0.3667**
UQLegalAI	uqlegalair3.txt	0.2962	0.2908	0.3019
UQLegalAI	uqlegalair2.txt	0.2957	0.2903	0.3013
UQLegalAI	uqlegalair1.txt	0.2940	0.2886	0.2996
NOWJ	prerank_dense_bge -rerank_bge_ft_llm2vec _major_vote.txt	0.1984	0.1670	0.2445
AIIR Lab	task1.aiirmpmist5.txt	0.2171	0.2040	0.2319
NOWJ	prerank_dense_bge -rerank_bge_ft.txt	0.1708	0.1605	0.1825
AIIR Lab	task1.aiircombmnz.txt	0.1879	0.2317	0.1580
AIIR Lab	task1.aiirmpmist3.txt	0.1872	0.2308	0.1575
NOWJ	prerank_dense_ llm2vec_llama31_8b.txt	0.1580	0.1485	0.1688
JNLP	**jnlpfe1.txt**	**0.1597**	**0.1307**	**0.2052**
OVGU	task1_ovgu2.txt	0.1498	0.1743	0.1313
UB_2025	run2.txt	0.1363	0.1955	0.1046
UB_2025	run3.txt	0.1171	0.1818	0.0864
UB_2025	run1.txt	0.1051	0.0572	0.6379
SIL	submission_sil_−_run_results.txt	0.0058	0.0054	0.0063
UA	ua_run3.txt	0.0000	0.0000	0.0000
UA	ua_run2.txt	0.0000	0.0000	0.0000
UA	ua_run1.txt	0.0000	0.0000	0.0000
OVGU	ignore_task1_ovgu1.txt	0.0000	0.0000	0.0000

Table 3: Performance of submitted systems on Task 1 COLIEE 2025

Q represents a query case, N is the set of multiple paragraphs from the noticed(cited) case, and p^* is the set of gold paragraphs those entail Q. Each model employed hard negative sampling during training: for a given query, negative samples were retrieved from the paragraphs those are non-entailing but contextually similar to the gold paragraph. This strategy increases the the model's ability to distinguish subtle semantic differences between correct and incorrect paragraphs. Following standard practice, the models were then fine-tuned on the 2024 training data. Table 4 depicted the official test results.

Table 4: Evaluation results for Task 2 submissions.

Team	Run	F1	Precision	Recall
NOWJ	nowj003	0.3195	0.3788	0.2762
NOWJ	nowj002	0.2865	0.2976	0.2762
NOWJ	nowj001	0.2782	0.2650	0.2928
OVGU	task2_ovgu2	0.2454	0.2759	0.2210
JNLP	**jnlp_002**	0.2412	0.2000	0.3039
JNLP	**jnlp_003**	0.2400	0.2708	0.2155
AIIR_Lab	task2crossaiirlab	0.2368	0.2927	0.1989
AIIR_Lab	task2mergeaiirlab	0.2229	0.2632	0.1934
OVGU	task2_ovgu3	0.1965	0.2692	0.1547
AIIR_Lab	task2mt5aiirlab	0.1930	0.2050	0.1823
CAPTAIN	run2	0.1882	0.2547	0.1492
CAPTAIN	run1	0.1812	0.2453	0.1436
JNLP	**jnlp_001**	0.1779	0.2500	0.1381
UA	submission3	0.1778	0.2090	0.1547
CAPTAIN	run3	0.1712	0.2252	0.1381
UA	submission1	0.1712	0.2252	0.1381
OVGU	task2_ovgu1	0.1708	0.2400	0.1326
UA	submission2	0.1736	0.2077	0.1492

For the first submission (**jnlp_001**), two transformer-based models were employed: **castorini/monot5-large-msmarco-10k** a large T5 variant specialized for re-ranking tasks; and **google/flan-t5-xxl** - a large instruction-tuned T5 model. The monoT5 model was fine-tuned on the 2024 training dataset with hard negative sampling. After generating candidate paragraphs with the fine-tuned monoT5 model, Flan-T5-XXL was prompted with a few demonstration examples (namely few-shot prompting) enabling it to rank the results based on the probability of generating the "yes" token. The final submission was done through hyperparameter search based on the 2024 test data.

For the second submission (**jnlp_002**), we employed a single re-ranking model: **BAAI/bge-reranker-v2-minicpm-layerwise**. This is a re-ranking model designed to handle bilingual (or multilingual) generative embeddings (BGE). As in **jnlp_001**, the model was fine-tuned on the 2024 training data with hard negative sampling. Following fine-tuning, the system performed inference on candidate paragraph. The selection threshold was determined through a hyperparameter search on the 2024 test data.

For the final run (**jnlp_003**), we combined two models in a multi-stage filtering and re-ranking workflow. The re-ranking model was **BAAI/bge-reranker-v2-minicpm-layerwise**, and identical to the **jnlp_002** run in terms of fine-tuning on the 2024 training data with hard negative sampling. This model was used as the first step to filter out paragraphs deemed irrelevant, using a relatively low threshold to retain plausible candidates. From the paragraphs shortlisted by the bge-reranker model, a second filter step used the **Qwen2.5-32B-Instruct** model. Few-shot prompts were manually designed and applied accross all queries.

5.3 Task 3

Dataset Settings. In this study, we utilize only the English-translated version of the queries and the corresponding civil law articles. The queries whose ID prefixes are R04 and R05 are chosen for validation and the public test sets, respectively. All of the remaining data provided is used for training purposes across the various stages of our pipeline.

Evaluation Metrics. F2 measure is employed for this task. The F2 measure is a variation of the traditional F1 score, commonly used in classification and information retrieval tasks to balance precision and recall. However, unlike the F1 score, the F2 measure weights the recall score twice as important as the precision score. As a result, systems that retrieve a greater number of relevant documents—albeit with the inclusion of some irrelevant ones—will achieve a higher F2 score. Therefore, the primary objective of this task is to maximize recall while maintaining an acceptable level of precision.

Models to Fine-tune in Stage 2. We employ a diverse set of models with varying sizes and underlying architectures for the experiments. Detailed specifications of these models are presented in Table 5. The selected models can be broadly divided into two groups: (1) Text embedding models which are designed to generate the dense high-dimensional representation for a given text input and (2) Casual language models which are trained on the next-token prediction objective.

Model Name	Model Type
e5-mistral-7b-instruct[3] [33, 34]	Text embedding model
gte-Qwen2-7B-instruct[4] [17]	Text embedding model
SFR-Embedding-2_R[5] [31]	Text embedding model
gemma-2-9b-it[6] [32]	Casual language model
gemma-2-27b-it[7] [32]	Casual language model
Phi-3-medium-4k-instruct[8] [2]	Casual language model
Mistral-7B-Instruct-v0.2[9] [13]	Casual language model
monot5-3b-msmarco-10k[10] [26]	Casual language model
Qwen2-7B-Instruct[11] [1]	Casual language model

Table 5: Base models for fine-tuning in Stage 2 of Task 3.

Official Test Results. The results of our three official submissions can be found in Table 6. Our best-performing submission, JNLP_RUN1, achieves the highest ranking in this Task 3. In this configuration, rather than ensembling all of the nine fine-tuned models listed in Table 5, we selectively ensemble only four models, i.e., e5-mistral-7b-instruct, gemma-2-9b-it, gemma-2-27b-it, and Phi-3-medium-4k-instruct, because this subset yields the highest F2 score on the validation set. Regarding JNLP_RUN2, we ensemble all of the nine fine-tuned models. However, this approach results in a noticeable decline in the F2 score, likely due to the abundance of models causing the overfitting problem. Finally, for JNLP_RUN3, the model configuration remains the same as JNLP_RUN1 but the data settings are different. Specifically, R05 is used as both the validation and public test sets. The remaining data, i.e., including the R04 set, is used for fine-tuning the four models. This configuration produces the lowest F2 score among all submissions, suggesting that incorporating the R04 set for fine-tuning the models is not optimal, hence the performance is hurt.

Development Results. Table 7 presents a comparative analysis between our proposed framework and the top-ranked submissions from the competitions of the previous years. Overall, our pipeline continuously outperforms the top solutions on both the R04 and R05 sets by a large margin. Furthermore, our method also demonstrates a high degree of consistency between the validation and the test sets with neat performance gaps across the runs.

Experimental Results of the Stage-1 Module. The effectiveness of our proposed method for the pre-retrieval stage is demonstrated through empirical evaluation. Specifically, our approach can manage to archive the recall rate of 98.46% on the R04 set and 95.38% on the R05 set while keeping only 50 articles for a query. These outcomes not only highlight the efficacy of combining a text

[3] https://huggingface.co/intfloat/e5-mistral-7b-instruct
[4] https://huggingface.co/Alibaba-NLP/gte-Qwen2-7B-instruct
[5] https://huggingface.co/Salesforce/SFR-Embedding-2_R
[6] https://huggingface.co/google/gemma-2-9b-it
[7] https://huggingface.co/google/gemma-2-27b-it
[8] https://huggingface.co/microsoft/Phi-3-medium-4k-instruct
[9] https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.2
[10] https://huggingface.co/castorini/monot5-3b-msmarco-10k
[11] https://huggingface.co/Qwen/Qwen2-7B-Instruct

Submission	F2	Precision	Recall
JNLP_RUN1 (Ours)	**0.8252**	**0.7928**	**0.8626**
CAPTAIN.H2	0.8189	0.8221	0.8401
CAPTAIN.H3	0.8093	0.7894	0.8468
CAPTAIN.H1	0.7993	0.8086	0.8198
JNLP_RUN2 (Ours)	**0.7757**	**0.7173**	**0.8288**
JNLP_RUN3 (Ours)	**0.7755**	**0.7320**	**0.8153**
INFA	0.6824	0.7568	0.6734
mpnetAIIRLab	0.6584	0.3514	0.8739
mistralRerank	0.5978	0.3198	0.7928
OVGU3	0.5959	0.6261	0.6059
OVGU2	0.5878	0.6014	0.5946
NVAIIRLab	0.5854	0.3018	0.7883
UIwa	0.5738	0.5777	0.5811
UImeta	0.5715	0.5709	0.5811
UIthr	0.5646	0.5946	0.5608
OVGU1	0.4609	0.4572	0.4730
UA-gte	0.2558	0.1000	0.4369
UA-mpnet	0.2506	0.0973	0.4302
UA-bm25_allMini	0.2085	0.0784	0.3649

Table 6: Performance comparison on the official test set R06 of Task 3.

Run	R04	R05
CAPTAIN.Missq [22] (COLIEE 2023)	0.7569	-
JNLP.constr-join [23] (COLIEE 2024)	-	0.7408
JNLP_RUN1 (Ours)	0.8017	0.8114
JNLP_RUN2 (Ours)	0.7943	0.7997
JNLP_RUN3 (Ours)	-	0.7780

Table 7: Comparison of F2 scores against top solutions from previous competitions of Task 3.

embedding model, a classical machine learning classifier, and a re-ranker model but also present a compelling strategy for developing a high-performance per-retrieval system in the legal domain.

5.4 Task 4

For Task 4, we conducted experiments using only the English-translated versions of the queries and civil law articles provided by the COLIEE 2025 organizers. This decision was made to ensure consistent language inputs and avoid the variability introduced by machine translation.

In the few-shot setting, we constructed an example pool consisting of 231 queries, each accompanied by its corresponding answer and reasoning. This pool served as in-context demonstrations to guide the model's predictions during inference. For each input query, we collect 3 examples from the example pool to guide LLM on how to answer the query. Across all experimental configurations, we utilized the Qwen2-72B-Instruct model to generate answers for the legal queries. To support retrieval-augmented generation, we employed the all-MiniLM-L6-v2 model as our dense passage retriever, enabling efficient retrieval of relevant legal articles for each query.

The performance of each method on the private test of the previous COLIEE competition is shown in 8. We conducted experiments on the 3 public test sets, which are R03, R04, and R05. According to the experimental results, we see that the few-shot method gives the most stable and high-performance result among the 3 methods.

Table 9 presents the official results of our three submitted runs **JNLP001**, **JNLP002**, and **JNLP003** on the Task 4 test set. Each run was configured with a different setting: **JNLP001** used the zeroshot setting, **JNLP002** employed the few-shot setting, and **JNLP003** utilized the reasoning ensemble approach.

Table 8: Performance comparison of Task 4 across R03, R04, and R05

	R03	R04	R05
Zeroshot (ours)	77.06	**79.21**	80.73
FewShot (ours)	**81.65**	76.24	**83.49**
Reasoning Ensemble (ours)	80.73	71.29	78.90

Table 9: Performance of participating systems on the official Task 4 test set

Team	Correct	Accuracy
KIS3	66	0.9041
KIS1	64	0.8767
LUONG01	63	0.8630
UIRunCoT	62	0.8493
KIS2	62	0.8493
CAPTAIN2	60	0.8219
UIRunLang	60	0.8219
JNLP002	59	0.8082
JNLP003	59	0.8082
CAPTAIN1	58	0.7945
CAPTAIN3	58	0.7945
UA2	57	0.7808
UA3	57	0.7808
JNLP001	56	0.7671
KLAP.H2	56	0.7671
UA1	55	0.7534
NOWJ.run1	54	0.7397
NOWJ.run2	54	0.7397
NOWJ.run3	54	0.7397
BaseLine	37	0.5068

6 Conclusions

In this work, we presented our approaches to address the retrieval and entailment tasks for both legal case law and statute law in

the context of the COLIEE 2025 competition. By combining classical machine learning techniques with fine-tuned large language models and advanced prompting strategies, our systems are able to perform both efficient information retrieval and sophisticated legal reasoning. The strong performance of our proposed pipelines on the official test sets underscores the value of integrating lightweight retrieval frameworks with the interpretive strength of modern Transformer-based models. Furthermore, extensive experimentation using automated hyper-parameter tuning contributes to the optimization process of each component within our frameworks. Our work highlights a promising direction for future legal support systems: adaptable and fine-grain multi-stage pipelines that balance performance and scalability for real-world legal applications.

References

[1] 2024. Qwen2 Technical Report. (2024).

[2] Marah Abdin, Jyoti Aneja, Hany Awadalla, Ahmed Awadallah, Ammar Ahmad Awan, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Jianmin Bao, Harkirat Behl, et al. 2024. Phi-3 technical report: A highly capable language model locally on your phone. *arXiv preprint arXiv:2404.14219* (2024).

[3] 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 and Data Mining*.

[4] Minh-Quan Bui, Dinh-Truong Do, Nguyen-Khang Le, Dieu-Hien Nguyen, Khac-Vu-Hiep Nguyen, Trang Pham Ngoc Anh, and Minh Le Nguyen. 2024. Data augmentation and large language model for legal case retrieval and entailment. *The Review of Socionetwork Strategies* 18, 1 (2024), 49–74.

[5] Quan Minh Bui, Chau Nguyen, Dinh-Truong Do, and Nguyen-Khang Le. 2023. JNLP Team: Deep Learning Approaches for Tackling Long and Ambiguous Legal. 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*, Vol. 13859. Springer Nature, 68.

[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)*.

[8] 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.

[9] Tobias Fink, Gabor Recski, Wojciech Kusa, and Allan Hanbury. 2023. Statute-enhanced lexical retrieval of court cases for COLIEE 2022. *arXiv preprint arXiv:2304.08188* (2023).

[10] Jerome H Friedman. 2001. Greedy function approximation: a gradient boosting machine. *Annals of statistics* (2001), 1189–1232.

[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 *JSAI International Symposium on Artificial Intelligence*. Springer, 109–124.

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

[13] Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7B. arXiv:2310.06825 [cs.CL] https://arxiv.org/abs/2310.06825

[14] 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.

[15] 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.

[16] 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).

[17] Zehan Li, Xin Zhang, Yanzhao Zhang, Dingkun Long, Pengjun Xie, and Meishan Zhang. 2023. Towards general text embeddings with multi-stage contrastive learning. *arXiv preprint arXiv:2308.03281* (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* (Virtual Event, Canada) *(SIGIR '21)*. Association for Computing Machinery, New York, NY, USA, 2356–2362. doi:10.1145/3404835.3463238

[19] M Lin, SC Huang, and HL Shao. 2022. 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)*.

[20] Xueguang Ma, Liang Wang, Nan Yang, Furu Wei, and Jimmy Lin. 2024. Fine-tuning llama for multi-stage text retrieval. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2421–2425.

[21] Multi-Linguality Multi-Functionality Multi-Granularity. 2024. M3-Embedding: Multi-Linguality, Multi-Functionality, Multi-Granularity Text Embeddings Through Self-Knowledge Distillation. (2024).

[22] Chau Nguyen, Phuong Nguyen, Thanh Tran, Dat Nguyen, An Trieu, Tin Pham, Anh Dang, and Le-Minh Nguyen. 2024. Captain at coliee 2023: Efficient methods for legal information retrieval and entailment tasks. *arXiv preprint arXiv:2401.03551* (2024).

[23] 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.

[24] 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 *JSAI International Symposium on Artificial Intelligence*. Springer, 125–139.

[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 *JSAI International Symposium on Artificial Intelligence*. Springer, 200–211.

[26] 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*, Trevor Cohn, Yulan He, and Yang Liu (Eds.). Association for Computational Linguistics, Online, 708–718. doi:10.18653/v1/2020.findings-emnlp.63

[27] Juliano Rabelo, Randy Goebel, Yoshinobu Kano, Mi-Young Kim, Masaharu Yoshioka, and Ken Satoh. 2021. Summary of the competition on legal information extraction/entailment (coliee) 2021. *Proceedings of the Eighth International Competition on Legal Information Extraction/Entailment (COLIEE 2021)* (2021), 1–7.

[28] 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.). Springer Nature Switzerland, Cham, 84–95.

[29] Juliano Rabelo, Mi-Young Kim, and Randy Goebel. 2023. Transformer-based legal information extraction. In *Workshop of the tenth competition on legal information extraction/entailment (COLIEE'2023) in the 19th international conference on artificial intelligence and law (ICAIL)*.

[30] 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.

[31] Shafiq Rayhan Joty Caiming Xiong Yingbo Zhou Semih Yavuz Rui Meng*, Ye Liu*. 2024. SFR-Embedding-2: Advanced Text Embedding with Multi-stage Training. https://huggingface.co/Salesforce/SFR-Embedding-2_R

[32] Gemma Team. 2024. Gemma. (2024). doi:10.34740/KAGGLE/M/3301

[33] Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2022. Text Embeddings by Weakly-Supervised Contrastive Pre-training. *arXiv preprint arXiv:2212.03533* (2022).

[34] Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. 2023. Improving Text Embeddings with Large Language Models. *arXiv preprint arXiv:2401.00368* (2023).

[35] Masaharu Yoshioka, Youta Suzuki, and Yasuhiro Aoki. 2022. Hukb at the coliee 2022 statute law task. In *JSAI International Symposium on Artificial Intelligence*. Springer, 109–124.

KIS: COLIEE 2025 Task 4 Solver Using Japanese LLM

Takaaki Onaga
Shizuoka University
Hamamatsu, Shizuoka, Japan
tonaga@kanolab.net

Yoshinobu Kano
Shizuoka University
Hamamatsu, Shizuoka, Japan
kano@inf.shizuoka.ac.jp

ABSTRACT

With the advancement of large language models (LLMs), natural language processing (NLP) technologies are being rapidly applied to the legal domain. In particular, LLMs have shown promising performance in complex tasks such as legal reasoning and case law analysis. However, systematic evaluations of LLMs on legal entailment tasks in Japanese remain limited.

In this study, we apply prompt engineering techniques using LLMs to COLIEE 2025 Task 4, a legal entailment recognition task based on Japanese bar exam questions. Specifically, we compare different prompt designs—Zero-shot, Few-shot, and label-balanced Few-shot—and evaluate how label distribution and the number of examples affect inference performance. We also propose a structured prompt incorporating Chain-of-Thought (CoT) reasoning, and examine its effectiveness through comparison with existing methods.

Experimental results show that the Few-shot (Balanced) setting achieved the highest average accuracy, with stable performance at $n = 6$. In addition, the proposed CoT-based prompt demonstrated significant accuracy improvements, particularly for case-type questions involving the application of legal provisions to specific factual scenarios.

CCS CONCEPTS

• **Computing methodologies** → **Natural language processing**; Few-shot learning; • **Social and professional topics** → *Legal aspects of computing*.

KEYWORDS

COLIEE, Question Answering, Legal Bar Exam, Legal Information, Large Language Models, LLM, Chain-of-Thought

ACM Reference Format:
Takaaki Onaga and Yoshinobu Kano. 2025. KIS: COLIEE 2025 Task 4 Solver Using Japanese LLM. In *Proceedings of COLIEE 2025 workshop, June 20, 2025, Chicago, USA*. ACM, New York, NY, USA, 10 pages.

1 INTRODUCTION

Natural language processing (NLP) technology has been rapidly adopted in the legal domain, gaining increasing attention for its potential utility. In particular, large language models (LLMs) have demonstrated strong performance not only in general NLP tasks but also in specialized, knowledge-intensive tasks such as legal reasoning and case law analysis. Notably, OpenAI's GPT-4 reportedly achieved a score exceeding the passing threshold for the Uniform Bar Exam in the United States [3].

One of the major NLP tasks in the legal domain is the Competition on Legal Information Extraction and Entailment (COLIEE) [1] [10] [9] [9] [7] [8] [15] [16] [14] [11] [5] [6], which has been held annually. Among its tasks, Task 4 targets binary questions from the Japanese bar examination (civil law) and aims to determine, whether a given question statement is legally valid, based on the associated legal statutes. This so-called legal entailment task requires the model to make judgments such as whether a particular contractual act is valid under civil law.

This task cannot be solved through simple surface-level matching or lexical processing; it demands deeper understanding of statutory structure, applicability conditions, exception clauses, and logical consistency between the statute and the facts presented in the question. In COLIEE 2024 Task 4, a variety of approaches utilizing LLMs have been proposed, many of which report improved accuracy and interpretability by incorporating few-shot prompting and Chain-of-Thought (CoT) prompting techniques [4].

However, most existing studies are based on English-language LLMs, and examples using models specifically designed for Japanese remain limited. Given that Task 4 is conducted entirely in Japanese, it is essential to systematically evaluate the capabilities of Japanese LLMs and to investigate the design of effective prompts tailored to them.

In this study, we explore the applicability of Japanese LLMs to legal reasoning tasks through a prompt engineering–based approach. First, we compare three baseline prompting methods—Zero-shot, Few-shot, and Balanced Few-shot (which considers label distribution)—and quantitatively evaluate how different prompt designs impact task performance.

Second, to address problems involving the application of legal statutes to complex factual scenarios, we propose a structured CoT prompting framework. This approach encourages step-by-step reasoning, consisting of "fact analysis, statute selection, reasoning, and conclusion,"with the aim of improving both inference stability and accuracy.

However, due to the constraints on allowable models in the COLIEE 2024 formal run, the LLM we used did not respond effectively to CoT-style prompts. As a result, our submitted system adopted a simpler Few-shot prompt design.

Nevertheless, our proposed methods achieved high accuracy in COLIEE 2024 Task 4, demonstrating the effectiveness of prompt engineering in leveraging Japanese LLMs for legal reasoning tasks.

COLIEE 2025, June 20, 2025, Chicago, USA

2 RELATED WORK

2.1 COLIEE

The Competition on Legal Information Extraction and Entailment (COLIEE) is an international workshop held annually with the aim of promoting research on natural language processing (NLP) technologies in the legal domain. The competition evaluates the performance of automated question-answering systems, particularly in the context of Japan's bar examination.

COLIEE consists of four tasks: Task 1 and Task 2 focus on Canadian case law, while Task 3 and Task 4 are based on Japanese bar examination questions.

Task 3 involves retrieving relevant statutes for a given question, whereas Task 4 is a legal entailment recognition task that determines, based on the given statutes, whether the question statement is legally correct or incorrect.

This study focuses on Task 4.

2.1.1 COLIEE Task 4. In COLIEE Task 4, a set of relevant statutes and a question based on the Japanese Civil Code are provided, and the model is required to determine whether the statement is legally correct. A concrete example is shown in Figure 1. This task demands not only the accurate interpretation of legal terminology but also advanced reasoning capabilities to assess the applicability conditions and exceptions of statutes, as well as the logical consistency between the legal text and the factual scenario described in the question.

 # 関連条文 / Relevant Statute
第六百九十八条
管理者は、本人の身体、名誉又は財産に対する急迫の危害を免れさせるために事務管理をしたときは、悪意又は重大な過失があるのでなければ、これによって生じた損害を賠償する責任を負わない。
(Article 698: If the manager undertakes management of affairs to avert imminent danger to the body, honor, or property of the principal, they are not liable for any resulting damage unless acting in bad faith or with gross negligence.)

 # 問題文 / Question Statement
車にひかれそうになった人を突き飛ばして助けたが、その人の高価な着物が汚損した場合、着物について損害賠償をする必要はない。
(A person who was about to be hit by a car was saved by being pushed out of the way, but their expensive kimono was soiled in the process. In such a case, there is no need to compensate for the damage to the kimono.)

Figure 1: Example question (ID: H18-2-2, Gold label: Yes; the original text is in Japanese)

Existing Approaches in COLIEE 2024 Task 4. In COLIEE 2024 Task 4, many studies adopted approaches based on large language models (LLMs). Fujita et al. [4] proposed a prompt design that explicitly guides the model through the reasoning process, demonstrating that such design can improve both performance and interpretability. They also emphasized that transparency in the legal reasoning process contributes to enhancing the trustworthiness of the model's answers.

Furthermore, Nguyen et al.[13] utilized Zero-Shot Chain-of-Thought (CoT)[12] prompting, using simple instructions such as "Let's think step by step."They proposed a method that reuses the output of such prompts as examples in Few-Shot prompting, as well as a fine-tuning approach using training data, both of which achieved high performance.

2.2 Prompt Engineering

2.2.1 Zero-Shot / Few-Shot Prompting. Zero-shot prompting is a method in which the model is given only instructions and input, without any examples. In contrast, few-shot prompting provides the model with a small number of illustrative examples, allowing it to learn the structure of the task. This approach has been shown to improve the consistency and accuracy of the model's outputs [2]. A comparison of the two prompting strategies is shown in Figure 2.

Zero-Shot Prompting
Instruction: Please answer whether the following sentence is positive or negative.
Sentence: I am angry

Few-Shot Prompting
Instruction: Please answer whether the following sentence is positive or negative.
Sentence: He is happy
Output: Positive
Sentence: She fell down
Output: Negative
Sentence: I am angry

Figure 2: Examples of Zero-Shot and Few-Shot Prompting

2.2.2 Chain-of-Thought (CoT). Chain-of-Thought (CoT)[17] is a prompting method that instructs the model to explicitly output the intermediate reasoning steps, rather than only the final answer. This approach has been shown to improve accuracy, particularly in tasks that require step-by-step reasoning. There are two common methods: providing actual reasoning examples in a few-shot setting, or using simple instructions such as "Let's think step by step."in a zero-shot setting[12]. An example is shown in Figure 3.

3 METHOD

In this study, we investigate a prompt engineering–based approach using large language models (LLMs) for COLIEE 2024 Task 4. All methods adopt a unified format in which a relevant legal statute and a question statement are given as input, and the model is required to determine whether the statement is correct or incorrect.

As one of our proposed methods, we develop a step-by-step reasoning framework that incorporates Chain-of-Thought (CoT) prompting.

Question:
Roger has 5 tennis balls. He then buys two cans, each containing 3 tennis balls.
How many tennis balls does Roger have in total now?

Answer:
Roger initially had 5 balls. The two cans contain 3 balls each, for a total of 6 balls.
5 + 6 = 11. Therefore, the answer is 11.

Figure 3: An example of Chain-of-Thought (CoT) prompting

Due to the COLIEE 2025 rule prohibiting the use of LLMs trained on data collected after July 9, 2024 (JST)—the day before the Japanese bar exam—we selected a compliant model (Swallow-70B) for our formal run. Meanwhile, we explored Chain-of-Thought (CoT) prompting in preliminary experiments using other models.

However, Swallow-70B did not reliably follow CoT-style prompts. Consequently, we submitted systems for the formal run that did not use CoT, but instead employed simpler Zero-shot or Few-shot prompting strategies.

Consequently, we submitted a system for the formal run that employed a simpler Few-shot prompting approach, without using CoT.

In the following prompt templates, placeholders enclosed in curly braces (e.g., ...) indicate where the corresponding input text was inserted during actual usage.

The list of methods implemented and evaluated in this study is as follows:

- **KIS: Formal Run Submissions**
  - **KIS1**: Zero-shot prompt
  - **KIS2**: Few-shot prompt
  - **KIS3**: Few-shot prompt with label balancing
- **CoT: Chain-of-Thought Methods**
  - **CoT-base**: Baseline CoT prompt based on existing format
  - **CoT-ours**: Structured CoT prompt using a fixed format

3.1 KIS: Formal Run Submissions

In this section, we describe the formal run submissions made by our team, KIS, for COLIEE 2024.

3.1.1 KIS1: Zero-shot Prompt. In KIS1, we adopted a zero-shot prompting approach in which the pre-trained LLM is provided only with task instructions, without any concrete examples. The prompt template is shown in Figure 4.

3.1.2 KIS2: Few-shot Prompt. In KIS2, we employed few-shot prompting by including multiple examples (shots) within the prompt, aiming to improve the model's task understanding and reasoning accuracy. The prompt template is shown in Figure 5.

In the few-shot setting, n example instances (shots) were selected based on the above template and inserted into the prompt. To select these shots, we first generated sentence embeddings for each training sample by concatenating the relevant statute and the

Zero-shot Prompt Template

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。
(Based on the relevant statute, please determine whether the following statement is correct or incorrect.)

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。
(If the statement is correct, answer with "解答: 正しい"; if it is incorrect, answer with "解答: 誤り".)

関連条文: {premise}
Relevant Statute: {premise}

問題文: {hypothesis}
Question Statement: {hypothesis}

Figure 4: Example of a zero-shot prompt template (original text is in Japanese)

Few-shot Prompt Template

指示 # *(Instruction)*
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。
(Based on the relevant statute, please determine whether the statement is correct or incorrect.)

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。
(If the statement is correct, answer with "解答: 正しい"; if incorrect, answer with "解答: 誤り".)

入力 # *(Input)*
関連条文: {premise}
Relevant Statute: {premise}

問題文: {hypothesis}
Question Statement: {hypothesis}

解答形式 # *(Answer format)*
解答: {label}
Answer: {label}

Figure 5: Example of a few-shot prompt template (original text is in Japanese)

question statement. Then, we calculated the cosine similarity between each training sample and the test case, and selected the top n examples with the highest similarity scores.

3.1.3 KIS3: Few-shot Prompt with Label Balancing. KIS3 is an extension of the KIS2 method, in which label balance (i.e., correct/incorrect) is explicitly considered during shot selection. Specifically, the n selected shots are adjusted such that half of them have the label "correct"and the other half "incorrect."

This design aims to mitigate output bias caused by imbalanced label distributions within the prompt, thereby improving the stability of the model's reasoning. Apart from the label balancing strategy, the overall prompt structure remains identical to that of KIS2.

3.2 Chain-of-Thought

3.2.1 CoT-ours. In this study, we propose a step-by-step reasoning framework based on Chain-of-Thought (CoT), aiming to improve both the transparency and accuracy of legal reasoning.

A previous method by Fujita et al. [4] employed a two-step prompting structure consisting of (1) summarizing key points and (2) referencing relevant statutes. In contrast, our framework reorganizes the reasoning process into the following four steps to enable more advanced inference:

(1) **Fact Analysis:** Clearly identify the relevant facts of the case.
(2) **Statute Analysis:** Organize the legal provisions necessary for judgment in a structured manner.
(3) **Reasoning:** Logically evaluate the relationship between the facts and statutes to derive a conclusion.
(4) **Conclusion:** State the final judgment. At the end, output the answer in the specified format.

We expect that this structured CoT prompting approach will enhance the transparency of the reasoning process and improve the accuracy of model outputs compared to previous methods.

Figure 6 shows the prompt templates for both the baseline Zero-shot CoT and the proposed structured CoT.

CoT-base Prompt Template

関連条文に基づいて、問題文が正しいか誤っているかをステップバイステップで解答してください。
(Based on the relevant statute, please answer step-by-step whether the question statement is correct or incorrect.)

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。
(If the statement is correct, answer with "解答: 正しい"; if incorrect, answer with "解答: 誤り".)

関連条文: {premise}
Relevant Statute: {premise}
問題文: {hypothesis}
Question Statement: {hypothesis}

CoT-ours Prompt Template

関連条文に基づいて、問題文が正しいか誤っているかを以下の推論ステップに従って解答してください。
(Based on the relevant statute, please determine whether the question statement is correct or incorrect by following the reasoning steps below.)

推論ステップ: *(Reasoning Steps)*
1. 問題文の事例（事実関係や人物関係）と論点を整理する。
Clarify the facts and relationships in the question.
2. 問題解決に必要な条文を整理する。
Identify and organize the relevant statutes for solving the problem.
3. 問題文の内容について、条文の要件に該当するか検討し、導かれる効果を説明する。
Evaluate whether the facts meet the statutory requirements and explain the resulting legal implications.
4. 問題文の正誤を結論づける。そして、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。
State the final conclusion. If the statement is correct, answer with "解答: 正しい"; if incorrect, answer with "解答: 誤り".

関連条文: {premise}
Relevant Statute: {premise}
問題文: {hypothesis}
Question Statement: {hypothesis}

Figure 6: Prompt templates for CoT-base and CoT-ours (original text is in Japanese)

4 EXPERIMENTS

In this chapter, we describe the experimental settings and results conducted to evaluate the proposed methods. The experiments were performed individually for the five methods introduced in the Method section: Zero-shot, Few-shot, Few-shot (Balanced), CoT-base, and CoT-ours.

The target task is COLIEE Task 4, which is a binary classification task. The model is required to determine whether a given statement is legally correct or incorrect, with reference to the relevant statutes.

4.1 Dataset

For the experiments, we used the training and test datasets provided in COLIEE 2025 Task 4. Although no model fine-tuning was performed in this study, the training data was utilized to select examples for few-shot prompting.

To analyze year-by-year performance variation, we used evaluation data from seven years, spanning from 2018 to 2024 —that is, 2018, 2019, 2020, 2021, 2022, 2023, and 2024. This corresponds to

the same setting as previous COLIEE formal runs for the past seven years, with 2024 representing the formal run setting for COLIEE 2025. In the dataset, case IDs follow the Japanese era naming convention, where for example, H30 corresponds to 2018 (Heisei 30), R01 to 2019 (Reiwa 1), and so on up to R06 for 2024.

Each sample in the training and test datasets consists of the following information:

- Relevant statute
- Question statement
- Label (Correct / Incorrect)

Evaluation Metric

We used accuracy as the evaluation metric to assess the performance of each method. The model outputs were analyzed to extract either "解答: 正しい"or "解答: 誤り,"and accuracy was computed based on exact matches with the ground truth labels.

4.2 KIS: Formal Run Submissions

4.2.1 LLM. For the formal run, we used the following large language model:

tokyotech-llm/Llama-3.1-Swallow-70B-Instruct-v0.3 [1]

This model is a 70B-parameter variant of Llama 3.1 that has been further pre-trained on Japanese data, enabling high-accuracy responses in Japanese.

The generation settings were configured as follows:

- `seed`: 42
- `max_model_len`: 8192
- `max_tokens`: 1024
- `temperature`: 0.6
- `top_p`: 0.9

4.2.2 Few-shot Example Selection Method. In the few-shot prompting setting, examples (shots) were selected only from the data of years prior to the target test year. Specifically, for a test year N, shots were extracted from data up to and including year $N-1$.

The selection procedure was as follows:

(1) For each sample, we concatenated the relevant statute and question statement, and converted the resulting text into a sentence embedding using Sentence Transformers, intfloat/multilingual-e5-large [2].
(2) We computed the cosine similarity between the test case and each training sample.
(3) We selected the top n examples with the highest similarity scores as few-shot examples.

In the Few-shot (Balanced) setting, the selected n examples were adjusted so that the number of examples labeled as "correct"and "incorrect"was evenly split, i.e., $n/2$ examples for each label. This was done to mitigate potential output bias caused by label imbalance within the prompt and to improve the stability of the model's reasoning. The number of shots n was varied from 1 to 6 for comparative evaluation.

[1] https://huggingface.co/tokyotech-llm/Llama-3.1-Swallow-70B-Instruct-v0.3
[2] https://huggingface.co/intfloat/multilingual-e5-large

4.3 CoT: Chain-of-Thought Methods

4.3.1 Comparison Methods. The CoT-based methods (CoT-base and CoT-ours) were applied to the same COLIEE 2025 Task 4 question set used in the formal run. Although a different LLM was used for these post-hoc experiments, the data and evaluation procedure were kept consistent with the other methods in this study.

We compared the following two types of step-by-step reasoning (Chain-of-Thought, CoT) approaches as baseline and proposed methods:

- **Zero-Shot Chain-of-Thought (CoT-base):** A zero-shot prompting method that encourages step-by-step reasoning using simple instructions such as "Let's think step by step." [12]
- **Proposed Chain-of-Thought (CoT-ours):** Our proposed method adopts a structured reasoning framework consisting of three steps: *fact analysis*, *statute analysis*, and *reasoning*. While conventional approaches often follow a two-step structure—summarizing key points and referencing relevant statutes—our method is designed to enable more precise evaluation of the relationship between legal provisions and the facts presented.

In this comparative experiment, we evaluated all methods under the zero-shot setting, without using few-shot examples. While few-shot prompting can be effective when the selected examples are highly similar to the test case, it also poses the risk of leading the model toward incorrect reasoning when the similarity is low. Therefore, to isolate and examine the pure effect of step-by-step reasoning, we adopted the zero-shot setting for all CoT-based evaluations in this study.

4.3.2 LLM. The following large language model (LLM) was used for both the proposed and baseline methods:

cyberagent/Llama-3.1-70B-Japanese-Instruct-2407 [3]

This model is based on Meta's Llama 3.1 70B and has been further trained on Japanese data by CyberAgent. It was selected for its strong instruction-following performance in Japanese.

The generation settings were configured as follows:

- `max_new_tokens`: 2048
- `do_sample`: False

5 RESULTS

In this chapter, we report the experimental results of both the proposed and baseline methods. Accuracy was used as the evaluation metric, and performance was calculated for each individual year.

5.1 KIS: Formal Run Submissions

Table 1 shows the results of the KIS team's formal run submissions for COLIEE 2025 Task 4. The team submitted outputs based on three different prompt designs: Zero-shot (KIS1), Few-shot (KIS2), and Few-shot with label balancing (KIS3).

Among them, KIS3 achieved the highest performance, correctly answering 67 out of 74 questions, with an accuracy of 90.54

In this section, we present a comparison of accuracy across the Zero-shot, Few-shot, and Few-shot (Balanced) methods.

[3] https://huggingface.co/cyberagent/Llama-3.1-70B-Japanese-Instruct-2407

Table 1: COLIEE 2025 Task 4 formal run results (number of correct answers and accuracy)

run	number of correct	Accuracy	run	number of correct	Accuracy
KIS3	**67**	**0.9054**	KLAP.H1	48	0.6486
KIS1	65	0.8784	OVGU3	47	0.6351
LUONG01	64	0.8649	RUG_V3	46	0.6216
UIRunCot	63	0.8514	AIIRLLaMA	45	0.6081
KIS2	63	0.8514	OVGU2	45	0.6081
CAPTAIN2	60	0.8108	RUG_V2	45	0.6081
JNLP002	60	0.8108	AIIRMistral	42	0.5676
JNLP003	59	0.7973	BaseLine	38	0.5135
CAPTAIN1	58	0.7838			
CAPTAIN3	58	0.7838			
UA2	58	0.7838			
UA3	58	0.7838			
JNLP001	57	0.7703			
KLAP.H2	57	0.7703			
UA1	56	0.7568			
NOWJ.run1	55	0.7432			
NOWJ.run2	55	0.7432			
NOWJ.run3	55	0.7432			
OVGU1	55	0.7432			
RUG_V1	49	0.6622			

Table 2: Accuracy (%) by year for the Zero-shot method. Values in parentheses indicate the number of questions.

2018 (70)	2019 (111)	2020 (81)	2021 (109)	2022 (101)	2023 (109)	2024 (74)	Avg.
87.14	77.48	83.95	83.49	78.22	79.82	87.84	82.99

Results of Zero-shot Prompting. The zero-shot method achieved stable performance overall. In particular, it recorded the highest accuracy of 87.14 in year 2018. On the other hand, the accuracy dropped slightly in 2019 and 2022, indicating potential limitations in the model's generalization ability across different test years.

Table 3: Accuracy (%) by number of shots (n) in the Few-shot setting

n	2018	2019	2020	2021	2022	2023	2024	Average
1	82.86	73.87	83.95	83.49	78.22	84.40	85.14	81.70
2	85.71	74.77	83.95	84.40	79.21	86.24	83.78	82.58
3	87.14	79.28	87.65	84.40	81.19	85.32	85.14	84.30
4	87.14	81.08	85.19	83.49	80.12	83.49	86.49	83.86
5	88.57	81.08	85.19	82.57	81.19	86.24	85.14	84.28
6	84.29	81.08	81.48	82.57	79.21	84.40	83.78	82.40

5.1.1 Results of Few-shot Prompting. Compared to the zero-shot setting, the few-shot prompting approach demonstrated overall improvements in accuracy. In particular, high accuracy was observed for $n = 3$ and $n = 5$, with $n = 3$ achieving the highest score of 87.65 in 2020. On the other hand, the average accuracy slightly decreased at $n = 6$, suggesting that an excessive number of shots may negatively affect reasoning performance.

Table 4: Accuracy (%) by number of shots (n) in the Few-shot (Balanced) setting

n	2018	2019	2020	2021	2022	2023	2024	Average
2	87.14	81.08	85.19	84.40	79.21	85.32	87.83	84.31
3	85.71	78.37	85.19	83.49	82.18	84.40	85.14	83.50
4	85.71	80.18	87.65	86.24	81.19	85.32	86.49	84.68
5	87.14	83.78	86.42	84.40	79.21	85.32	89.19	85.07
6	84.29	82.88	87.65	83.49	81.19	86.24	90.54	85.18
7	85.71	81.08	86.42	83.49	81.19	84.40	87.84	84.30

5.1.2 Results of Few-shot (Balanced) Prompting. Few-shot (Balanced) achieved more stable accuracy by balancing the label distribution and demonstrated strong overall performance. In particular, the average accuracy reached **85.18** at $n = 6$, marking the highest value among all settings. Even at $n = 6$, the model exhibited outstanding performance, with scores of **87.65** for 2020 and **86.24** for 2023.

These results indicate that few-shot prompting provides a clear improvement over zero-shot prompting. Furthermore, incorporating label balancing in example selection helps stabilize model outputs. The consistently high performance observed in the range of $n = 4$ to $n = 6$ suggests that both the number of shots and the label composition are critical factors in prompt design.

5.2 CoT: Chain-of-Thought

Next, we compare the performance of the proposed method incorporating step-by-step reasoning (CoT-ours) with the baseline method, Zero-Shot Chain-of-Thought (CoT-base).

Table 5: Comparison of CoT-based methods in terms of accuracy (%)

Method	2019	2020	2021	2022	2023	2024	Average
CoT-base	74.77	85.19	81.65	78.22	**85.32**	79.73	80.81
CoT-ours	**82.88**	**91.36**	**89.91**	**81.19**	84.40	81.08	**85.14**

CoT-ours outperformed CoT-base in terms of overall average accuracy. Notable improvements were observed in 2019 through 2022. On the other hand, the result for 2023 was slightly lower than that of CoT-base, which may be attributed to differences in question characteristics or reasoning complexity specific to that year.

Overall, the results suggest that the proposed method, with its structured step-by-step reasoning framework, is an effective approach for enhancing the legal reasoning capabilities of LLMs.

6 DISCUSSION

In this chapter, we provide a detailed analysis of the performance differences and trends observed in the previous chapter. In particular, we examine how the number of shots in the Few-shot and Few-shot (Balanced) settings affects model performance. Additionally, we discuss the qualitative impact of prompt design on the reasoning process.

6.1 KIS: Formal Run Submissions

6.1.1 Evaluation Perspectives. To comprehensively compare the performance of the few-shot-based methods, we defined the following four evaluation criteria:

- **Maximum Accuracy:** The highest accuracy recorded among all test sets.
- **Minimum Accuracy:** The lowest accuracy recorded among all test sets.
- **Average Accuracy:** The average accuracy across all years.
- **2023 Accuracy:** The accuracy for 2023, which contains the largest number of test samples.

Through these criteria, we evaluated not only the accuracy but also the stability and reliability of each method in practical applications.

6.1.2 Analysis of Few-shot Prompting. In the Few-shot setting without label balancing, the highest average accuracy (**84.30**) was achieved with $n = 3$, which also recorded the highest score of **87.65** in 2020. This suggests that high performance can be attained with a relatively small number of shots.

Meanwhile, the highest score for 2023 (**86.24**) was obtained with $n = 5$, indicating that increasing the number of shots may be effective when a sufficient amount of data is available. However, the average accuracy dropped to **82.18** at $n = 6$, revealing that an excessive number of shots may degrade the model's reasoning performance.

Overall, using $n = 3$ to $n = 5$ shots appears to be effective in Few-shot prompting, with $n = 3$ offering a particularly favorable balance between accuracy and stability.

6.1.3 Analysis of Few-shot (Balanced) Prompting. In the Few-shot (Balanced) setting, controlling the label distribution led to consistent improvements in performance. In particular, the $n = 6$ configuration can be highly evaluated based on the following points:

- **Maximum Accuracy:** 87.65 in 2020 (tied with $n = 4$)
- **Minimum Accuracy:** 84.29 in 2018 (the highest minimum value among all configurations)
- **Average Accuracy:** 84.29 (comparable to $n = 4$ and $n = 5$)
- **2023 Accuracy:** 86.24 (highest among all n values)

The configuration with $n = 6$ demonstrates strong performance in terms of both accuracy and stability, making it a promising option, particularly in scenarios that assume real-world deployment.

6.1.4 Overall Comparison of Methods. The overall evaluation of the three prompting methods is as follows:

Zero-shot offers the simplest configuration while achieving a reasonable level of accuracy, making it suitable as a baseline. However, its performance showed large fluctuations on more difficult questions, indicating a lack of stability.

Few-shot demonstrated improved accuracy when the number of shots was appropriately configured. In particular, the $n = 3$ setting offered a good balance between performance and implementation cost, making it an effective option.

Few-shot (Balanced) further enhanced stability by adjusting label distribution. The $n = 6$ configuration achieved consistently high performance across all evaluation criteria, making it the most suitable choice for practical applications.

In summary, for tasks requiring legal reasoning, few-shot prompting with label balancing provides the best trade-off between accuracy and stability. Among the configurations, $n = 6$ was found to be especially effective.

6.2 Qualitative Analysis of Few-shot Prompting

To better understand the influence of few-shot prompting on the reasoning process of large language models (LLMs), we conducted a qualitative analysis on cases where the model gave an incorrect answer in the zero-shot setting but produced the correct answer when using few-shot prompting.

The example shown in Figure 7 (Question ID: R05-06-A) concerns Article 132 of the Japanese Civil Code (Illegal Conditions). In the zero-shot setting, the model failed to correctly interpret the provision that "a legal act conditional upon refraining from an illegal act is also void,"and produced an incorrect judgment.

In contrast, under the few-shot setting, the prompt included a similar example involving the same statute. This allowed the model to better understand the legal intent of the provision and arrive at the correct answer. This result suggests that the auxiliary information provided by few-shot examples enhanced the model's ability to interpret the statute and apply it appropriately.

Through few-shot prompting, the contextual meaning and interpretative nuance of the statute were more effectively conveyed to the model, enabling more accurate legal reasoning.

Example of Correct Answer via Few-shot Prompting (Question ID: R05-06-A)

問題 ID: R05-06-A, 正解ラベル: 正しい
Question ID: R05-06-A, Gold Label: Correct

関連条文:
第 132 条（不法条件）:「不法な条件を付した法律行為は、無効とする。不法な行為をしないことを条件とするものも、同様とする。」
Article 132 (Illegal Conditions): "A legal act with an illegal condition shall be void. The same applies to a legal act that is conditional upon refraining from an illegal act."

問題文:
ＡがＢとの間で、Ｂが一定期間窃盗をしなかったら１０万円をＢに与える旨の贈与契約を締結した場合において、その期間窃盗をしなかったＢがＡに１０万円の支払を請求したときは、Ａは、これを拒むことができる。
A and B entered into a gift agreement stipulating that B would receive 100,000 yen if B refrained from theft for a certain period. After fulfilling this condition, B demanded payment, but A refused.

Zero-shot 出力: 解答: 誤り（False）← 誤答
Zero-shot Output: Answer: Incorrect —Incorrect Response

 # Few-shot サンプル #
問題 ID: H24-6-I, 正解ラベル: 誤り
条文: 第 132 条（不法条件）:「不法な条件を付した法律行為は、無効とする。不法な行為をしないことを条件とするものも、同様とする。」
Article 132 (Illegal Conditions): "A legal act with an illegal condition shall be void. The same applies to a legal act that is conditional upon refraining from an illegal act."
問題: 不法な条件を付した法律行為は無効であるが、不法な行為をしないことを条件とする法律行為は有効である。
A legal act with an illegal condition is void, but a legal act conditioned on not committing an illegal act is valid.

Few-shot 出力: 解答: 正しい（True）← 正答
Few-shot Output: Answer: Correct —Correct Response

Figure 7: Qualitative analysis of reasoning improvement via Few-shot Prompting (Question ID: R05-06-A; original text is in Japanese)

6.3 Error Analysis

This section presents an example of a problem on which the model made an incorrect prediction and analyzes the nature of the error.

Example of Model Error (Question ID: R01-2-I)

問題 ID: R01-2-I, 正解ラベル: 正しい
Question ID: R01-2-I, Gold Label: Correct

関連条文:
第二十八条：管理人は、第百三条に規定する権限を超える行為を必要とするときは、家庭裁判所の許可を得て、その行為をすることができる。
第百三条：権限の定めのない代理人は、次に掲げる行為のみをする権限を有する。
一保存行為
二代理の目的である物又は権利の性質を変えない範囲内において、その利用又は改良を目的とする行為
Article 28: The administrator must obtain permission from the family court when performing acts exceeding the authority under Article 103.
Article 103: An agent without expressly granted authority may only perform: (1) preservative acts; (2) acts that do not change the nature of the object or right but use or improve it.

問題文:
A がその財産の管理人を置かないで行方不明となったことから、家庭裁判所は、B を不在者 A の財産の管理人として選任した。A が所有する現金が発見された場合、B が A を代理してその現金を D 銀行の A 名義普通預金口座に預け入れるためには、家庭裁判所の許可を得る必要はない。
Since A went missing without appointing a property administrator, the family court appointed B as the administrator of A's property. When cash owned by A is found, B deposits the cash into A's bank account at D bank. The question is whether B must obtain family court permission for this act.

モデル出力: 解答: 誤り（False）← 誤答
Model Output: Answer: Incorrect —Incorrect Response

Figure 8: Qualitative analysis of a model error on Question ID: R01-2-I (original text is in Japanese)

As shown in Figure 8, the selected case (ID: R01-2-I) concerns an interpretation of Articles 28 and 103 of the Japanese Civil Code. The key issue is whether a court-appointed administrator of an absentee's property must obtain permission from the family court to deposit the absentee's cash into a bank account.

In this case, the correct answer is True. Depositing cash into a bank account under the absentee's name is considered a preservative act under Article 103. Therefore, the administrator is not required to obtain permission from the family court.

However, the model incorrectly judged this act as one that changes the nature of the property and predicted the answer as False. This indicates that the model lacked the ability to correctly understand and apply the subtle legal distinction of whether the act constitutes a preservative act. Instead, it appears the model relied on superficial lexical matching rather than performing deeper normative legal reasoning.

The model might lack the ability to understand and apply subtle legal distinctions and tends to rely on superficial lexical matches. Focusing on prompt designs that facilitate legal reasoning and training on datasets that include reasoning steps is our future work.

6.4 CoT: Chain-of-Thought

6.4.1 Effectiveness Analysis by Question Type. To gain a more fine-grained understanding of the effectiveness of the proposed method, we manually categorized the evaluation questions into the following two types:

- **Statute-type questions:** Questions that directly ask about the content of a legal provision.
- **Case-type questions:** Questions that require applying a legal provision to a specific factual scenario.

The questions shown in Figure 1 and Figure 7 both fall under the category of *case-type questions*. As shown in Table 6, the proposed method achieved high accuracy on both question types, with particularly notable improvements observed in case-type questions.

Table 6: Accuracy by question type (number of questions shown in parentheses)

Prompt	Statute (308)	Case (194)	Total (511)
CoT-base	86.04	74.74	81.41
CoT-ours	**87.66**	**85.05**	**85.71**

The structured prompt design of the proposed method—consisting of "fact analysis, statute analysis, reasoning, and conclusion" —appears to be particularly effective for questions involving complex factual scenarios. This suggests that the method helps LLMs more accurately apply legal provisions to facts in a consistent manner, a task that has been challenging for previous approaches.

Furthermore, the CoT-based method not only demonstrated higher accuracy in case-type questions but also achieved favorable overall accuracy in certain test years. For example, in 2020 and 2021, the structured CoT approach outperformed all other methods, including Balanced Few-shot. This suggests that CoT prompts are particularly well-suited to scenarios requiring layered reasoning and precise legal interpretation.

7 CONCLUSION

In this study, we investigated the effectiveness of prompt design using large language models (LLMs) for COLIEE 2024 Task 4, a legal entailment recognition task based on questions from the Japanese bar examination. In particular, we examined the comparative performance of Zero-shot, Few-shot, and label-balanced Few-shot (Balanced) prompting, and demonstrated that introducing step-by-step reasoning via Chain-of-Thought (CoT) prompts can enhance the accuracy and stability of LLMs in legal reasoning tasks.

Experimental results showed that the Few-shot (Balanced) approach outperformed both Zero-shot and standard Few-shot prompting in terms of overall accuracy and stability, with the $n = 6$ configuration achieving the best performance. Additionally, the proposed CoT-based method exceeded the performance of the baseline Zero-Shot CoT (ZS-CoT), especially in case-type questions that require applying legal provisions to specific factual scenarios.

Our qualitative analysis further revealed that Few-shot prompting helped deepen the model's understanding of legal statutes, allowing it to avoid incorrect answers in certain cases. These findings underscore the significant impact of prompt design on the model' s ability to apply legal knowledge, reinforcing the importance of prompt engineering.

Overall, the results demonstrate that the structure of the prompt has a direct influence on LLM performance in legal reasoning tasks. In particular, techniques such as label balancing and structured step-by-step reasoning represent effective approaches for enabling LLMs to solve bar exam questions more accurately.

ACKNOWLEDGMENTS

This research was partially supported by Kakenhi, MEXT Japan (JP22H00804, JP23K22076), JST PRESTO (JPMJPR2461), JST AIP Acceleration Research (JPMJCR22U4), and SECOM Science and Technology Foundation.

REFERENCES

[1] 2014. Competition on Legal Information Extraction/Entailment (COLIEE-14) Workshop on Juris-informatics (JURISIN) 2014. http://webdocs.cs.ualberta.ca/~miyoung2/jurisin_task/index.html.

[2] Tom 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. 2020. Language Models are Few-Shot Learners. arXiv:2005.14165 [cs.CL] https://arxiv.org/abs/2005.14165.

[3] OpenAI et al. 2024. GPT-4 Technical Report. arXiv:2303.08774 [cs.CL] https://arxiv.org/abs/2303.08774.

[4] Masaki Fujita, Takaaki Onaga, and Yoshinobu Kano. 2024. LLM Tuning and Interpretable CoT: KIS Team in COLIEE 2024. In *New Frontiers in Artificial Intelligence*. 140–155.

[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*. 472–480.

[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 Competition (COLIEE) 2024. In *New Frontiers in Artificial Intelligence*. 109–124.

[7] Yoshinobu Kano, Mi-Young Kim, Randy Goebel, and Ken Satoh. 2017. Overview of COLIEE 2017. In *COLIEE 2017. 4th Competition on Legal Information Extraction and Entailment (EPiC Series in Computing, Vol. 47)*, Ken Satoh, Mi-Young Kim, Yoshinobu Kano, Randy Goebel, and Tiago Oliveira (Eds.). 1–8. https://doi.org/10.29007/fm8f

[8] Yoshinobu Kano, Mi-Young Kim, Masaharu Yoshioka, Yao Lu, Juliano Rabelo, Naoki Kiyota, Randy Goebel, and Ken Satoh. 2019. COLIEE-2018: Evaluation of the Competition on Legal Information Extraction and Entailment. In *New Frontiers in Artificial Intelligence*, Kazuhiro Kojima, Maki Sakamoto, Koji Mineshima, and Ken Satoh (Eds.). Springer International Publishing, Cham, 177–192.

[9] Mi-Young Kim, Randy Goebel, Yoshinobu Kano, and Ken Satoh. 2016. COLIEE-2016: evaluation of the competition on legal information extraction and entailment. In *International Workshop on Juris-informatics (JURISIN 2016)*.

[10] Mi-Young Kim, Randy Goebel, and Satoh Ken. 2015. COLIEE-2015: evaluation of legal question answering. In *Ninth International Workshop on Juris-informatics (JURISIN 2015)*.

[11] Mi-Young Kim, Juliano Rabelo, Randy Goebel, Masaharu Yoshioka, Yoshinobu Kano, and Ken Satoh. 2023. COLIEE 2022 Summary: Methods For Legal Document Retrieval And Entailment. 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* (Kyoto, Japan). Springer-Verlag, Berlin, Heidelberg, 51–67. https://doi.org/10.1007/978-3-031-29168-5_4

[12] 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. 22199–22213.

[13] 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*. 125–139.

[14] 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. https://doi.org/10.1007/s12626-022-00105-z

[15] Juliano Rabelo, Mi-Young Kim, Randy Goebel, Masaharu Yoshioka, Yoshinobu Kano, and Ken Satoh. 2020. A Summary of the COLIEE 2019 Competition. In *New Frontiers in Artificial Intelligence*, Maki Sakamoto, Naoaki Okazaki, Koji Mineshima, and Ken Satoh (Eds.). Springer International Publishing, Cham, 34–49.

[16] Juliano Rabelo, Mi-Young Kim, Randy Goebel, Masaharu Yoshioka, Yoshinobu Kano, and Ken Satoh. 2020. COLIEE 2020: Methods for Legal Document Retrieval and Entailment. In *New Frontiers in Artificial Intelligence: JSAI-IsAI 2020 Workshops, JURISIN, LENLS 2020 Workshops, Virtual Event, November 15–17, 2020, Revised Selected Papers*. Springer-Verlag, Berlin, Heidelberg, 196–210. https://doi.org/10.1007/978-3-030-79942-7_13

[17] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. arXiv:2201.11903 [cs.CL] https://arxiv.org/abs/2201.11903.

Investigating Expert-Based Prompt Engineering for Legal Entailment 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

Tadeusz Zbiegień
tadeusz.zbiegien@doctoral.uj.edu.pl
Department of Legal Theory, Jagiellonian University
Poland

Dries Wedda
Bernoulli Institute of Mathematics, Computer Science and Artificial Intelligence, University of Groningen
The Netherlands

Junjun Liu
Bernoulli Institute of Mathematics, Computer Science and Artificial Intelligence, University of Groningen
The Netherlands

Abstract

Legal reasoning is complex and multi-faceted, requiring a broad set of skills. By employing domain knowledge from legal experts, we design five elements that can be included in prompts for large language models that could aid in legal reasoning tasks. We use additional legal guidelines, 1-shot prompting, dictionary definitions, knowledge representations of legal articles, and IRAC-style prompting. We investigate the effect of each prompt element on the model's performance on a legal entailment task. Certain prompt elements can improve performance, depending on the context and the model. For the smaller model, increasing the number of prompt elements improves performance on average. For any particular combination of model and sub-task, only using a subset of the prompt elements seems to work best. For the most advanced reasoning model we evaluate, using a selection of prompt elements increases average performance across all sub-tasks we evaluate. Results indicate that the problem space of the legal entailment task may be too large for a single model and prompt. In future research, we therefore aim to investigate the capabilities of an ensemble of specialized models.

CCS Concepts

• **Computing methodologies** → **Natural language generation**; **Natural language processing**; **Knowledge representation and reasoning**; **Machine learning**; • **Applied computing** → **Law**.

Keywords

Large language models, legal reasoning, prompt engineering

ACM Reference Format:
Cor Steging, Ludi van Leeuwen, Tadeusz Zbiegień, Dries Wedda, and Junjun Liu. 2025. Investigating Expert-Based Prompt Engineering for Legal Entailment Tasks. In *Proceedings of COLIEE 2025 workshop, June 20, 2025, Chicago, USA*. ACM, New York, NY, USA, 10 pages.

COLIEE 2025, Chicago, USA

1 Introduction

Passing the bar exam is difficult. It tests for a wide range of skills that lawyers should possess to be admitted to the bar of their jurisdiction. One part of such a bar exam may be a set of written questions, where lawyers are challenged not only on their knowledge of the law, but also on their legal reasoning capabilities: a complex and multi-faceted process, requiring different types of skills [9]. Recent advances in AI have produced large language models (LLMs) that appear to perform well at a large variety of reasoning tasks, and the GPT-4o model was even shown to pass the US bar exam [17], though questions have been raised about the results [20]. Currently, work is being done to systematically evaluate and improve the legal reasoning capabilities in these LLMs, for example, by creating benchmarks [13]. While LLMs can appear to possess emerging legal reasoning capabilities, their performance can be adjusted using prompt engineering techniques [16]. In this study, we investigate several methods to integrate expert knowledge in prompts to improve the legal reasoning capabilities of LLMs. We focus on the task of solving legal entailment questions from the Japanese bar exam as part of the COLIEE competition.

2 Background

To investigate the effects of expert-based prompt engineering in the legal domain, we focus on the task of predicting legal entailment. Specifically, we design and employ prompt elements in the context of the COLIEE: a competition that aims to progress the state of the art of legal information retrieval and entailment.

2.1 Legal entailment task

We focus on task 4 of the COLIEE, dealing with translated Japanese bar exam questions where one needs to determine whether there is a legal entailment between a set of legal articles S and a statement Q. We show two example questions in Figure 1. The systems designed to solve these questions should perform the required legal reasoning to determine the label: whether there is an entailment between S and Q. The type of legal reasoning that is needed, however, differs per question.

Article(s) S:
Article 620
If a lease is canceled, the cancellation becomes effective solely toward the future. In such a case, the cancellation does not preclude a claim for compensation for loss or damage.
Article 684
The provisions of Article 620
apply mutatis mutandis to partnership contracts.

Statement Q:
The cancellation of a partnership contract shall solely become effective toward the future.

Label: Yes

(a) Example question A.

Article(s) S:
(Capacity for Liability)
Article 712 If a minor has inflicted damage on another person but did not have sufficient intellectual capacity to appreciate their own liability for that act, that minor is not liable to compensate for that act.

Statement Q:
Mental capacity means the capacity to appreciate one's own liability for one's acts.
Label: No

(b) Example question B.

Figure 1: Two examples of legal entailment question.

In example question *A* (Figure 1a), there is a clear entailment between *S* and *Q*, as Article 620 also applies to partnership contracts according to Article 684. This does, however, require an understanding of the term 'mutatis mutandis', which is information that is not present in either the article or statement. In other words, this particular question requires additional expert domain knowledge. In example question *B* (Figure 1b), there is no legal entailment, because there is a mismatch in the terms used: intellectual capacity and mental capacity are two distinct terms with different meanings. Recognizing when such a 'specificity issue' occurs is a matter that lawyers are trained to look out for. These two examples are not the only type of questions that occur in the legal entailment task, and solving every question thus requires a wide range of legal reasoning skills.

2.2 Legal reasoning

From a lawyer's perspective, the space of legal reasoning is broad and thus hard to define. Nevertheless, we try to point out some typical themes raised in the legal literature. Legal reasoning, among others, encompasses deductive, inductive, abductive, analogical, and teleological reasoning, each of which has a different role depending on the situation [36].

Deductive reasoning has long been seen as a component of 'mechanical jurisprudence', where a legal syllogism yields the outcome [36]. However, jurists note that a strictly deductive model rarely suffices in practice, since most legal arguments are *defeasible* [26]. In fact, many cases cannot be resolved using formal logic alone, meaning courts must consider other ways of reasoning.

Inductive reasoning is also fundamental, especially for the development of common law, as it is based on generalising principles from particular cases. In his classic work, Edward H. Levi described legal reasoning as 'reasoning by example' - an inductive process in which a rule emerges from one case and is applied to the next [19]. Within legal scholarship legal reasoning is often examined through the two aforementioned primary lenses of inductive and deductive reasoning, which (depending on position taken) roughly correspond to: the 'top-down' and 'bottom-up' approaches [2, 23].

Analogical reasoning likewise has been emphasised by scholars as crucial in legal thought. Rather than strict deduction, judges often draw analogies between fact patterns: they compare a new dispute to earlier precedents and reason that like cases should be treated alike [33, 3].

Abductive reasoning, as inference to the best explanation, has been highlighted by scholars in legal thought. A recent article by Bjarte Askeland in *Ratio Juris* [4] argues that abductive inference has *"great potential for categorising new phenomena under norms"* – for instance, when courts must fit unprecedented situations into existing legal categories.

Teleological reasoning is another dimension in which legal decision-makers consider the object or spirit of a provision [18]. This type of reasoning complements more formal reasoning by ensuring that the law is not applied in a vacuum but rather in light of its purpose.

Legal decision-makers shift among modes as a case requires, and no single inferential method dominates across all legal questions. For example, a court may begin with a deductive application of a clear rule; if the rule's terms are ambiguous, it might then resort to analogies from precedent; if precedent is lacking, it may invoke the broader purpose of the law or infer a new principle that best explains the existing legal landscape.

Within the domain of legal scholarship. AI & Law has significantly contributed to our understanding of legal reasoning. These advancements allow for systematic explorations of legal arguments, which can be automatically analyzed and structured for consistency, coherence, and interpretative rigor [35, 27, 12]. This approach is reflected in the COLIEE competition tasks, where legal entailment problems drawn from bar examinations offer a setting for investigating capabilities of computational models in performing complex legal reasoning tasks.

2.3 State of the art

The legal entailment task has been featured in the COLIEE competition in previous years. The increased capability of generative AI, specifically LLMs, has raised questions about the importance of 'reasoning' in this task. This is because LLM-based models perform as well or better than models that use explicit, logical reasoning. The best performing system in the previous iteration of the competition used n-shot prompting with a google-flan-xxl model, where an example was provided in the prompt to make use of the in-context learning ability of LLMs [22]. Other approaches in last year's competition used ensemble methods with majority voting and various methods of prompt engineering such as CoT prompting, fine-tuning, or data augmentation [10]. The shift from rule-based systems combined with NLP techniques like BERT [25] to newer LLM-based approaches [11, 10] has brought about new challenges. These involve both methodological challenges, such as discovering how one should investigate the performance of an LLM, and ethical challenges: LLMs that come from a closed source are controversial

because we cannot be sure that they are not contaminated with the test data. In the broader legal domain, prompt engineering techniques have been explored for the legal syllogism task [16], a sub-task of the legal entailment task that we investigate in this study. This prompting technique uses explicit legal syllogism instructions, and it was shown to achieve a better performance than baseline and conventional chain of thought prompting.

3 Prompt elements

In our approach, we make use of the in-context learning abilities of generative language models by adding additional information to the prompt to aid the model in the legal entailment task. We explore five different elements that can be added to the prompts and investigate the extent to which these influence the performance of the models. These prompt elements consist of additional expert legal knowledge in the form of guidelines, example cases from the past, dictionary definitions of difficult legal terms, results from reasoning using knowledge-based representations of legal articles, and rules to encourage IRAC-style reasoning. In this section, we describe each of these five elements, the reasoning behind why they might be useful, and how they are implemented.

3.1 Legal reasoning guidelines

Based on an extensive manual examination of COLIEE entailment task questions and an error analysis of state-of-the-art models, we developed a set of explicit legal reasoning guidelines designed to reduce reasoning errors made by language models. These guidelines correspond closely to established jurisprudential reasoning concepts, without following any specific jurisdiction, but rather employing certain basic and key-notions that would correspond to formalistic legal reasoning. By examining the identified errors and taking a legal perspective, we were able to understand why questions were answered incorrectly. As indicated above we created the guidelines based on the detected errors. Consequently, the guidelines do not cover all possible legal reasoning guidelines, as this would not be feasible due to the broad nature of legal reasoning. They were intended as an intervention on the particular datasets and tasks studied and not as a fully formed legal reasoning framework.

The resulting guidelines can be found in Figure 2. Guideline (1) embodies classical legal formalism and syllogistic reasoning, requiring logical entailment [29]. Guideline (2) aligns with textualist interpretative strategies, restricting the inference strictly to information provided explicitly within the statutory text (see e.g. [28]). Guideline (3) focuses on rigorous rule-based reasoning with clear distinctions between conjunctive and disjunctive statutory conditions (see e.g. [30]). Guideline (4) focuses on defeasible reasoning principles, demanding explicit checks for conditional and exception clauses (see e.g. [24]). Guidelines (5) and (6) emphasize systematic, step-by-step verification of each condition and clear articulation of inferential reasoning (see also Section 2.2).

Collectively, these guidelines follow some of the traditional legal guidelines pertaining to statutory interpretation. However, legal theorists have long noted that this picture is too simplistic for all situations. H.L.A. Hart famously observed that while many cases lie within a *'core of settled meaning'* where rules apply, there is also a *"penumbra of debatable cases"* where it's not obvious how the rule applies [14].

Use all of the following guidelines to help you determine entailment:

(1) Q must be inevitable under S. If S could be true and Q false, then answer "No." Do not confuse Q merely fitting or being consistent with S for entailment – it must logically follow.
(2) Only use information from S. Do not assume any facts, duties, or conditions not stated in S. Interpret the language of S exactly and do not expand its scope.
(3) Treat all "and" conditions in S as jointly required (all must be present). Treat "or" conditions as alternatives (any one suffices). Apply these connectors strictly as written.
(4) If S has an "if" or condition clause, ensure those conditions hold true for Q. If S has an exception (e.g., "unless..."), check if Q falls into that exception. An exception that applies means S does not entail Q.
(5) Consider each condition of S and verify it against Q step by step. Use a logical order: check conditions, then exceptions, then draw the conclusion. Think of each condition/exception as a checkpoint in your reasoning.
(6) Present your reasoning in clear steps (Step 1, Step 2, Step 3, ...). For each step, state what you are checking or inferring. Only after laying out the analysis, give the final line as "Answer: Yes" or "Answer: No." Make sure the conclusion follows inevitably from the prior steps.

Figure 2: Our guidelines that can be added to the prompt.

3.2 1-shot prompting

One popular way of leveraging in-context learning is by providing example cases from the past to indicate how the model should behave. This method is called n-shot prompting, where n denotes the number of examples provided in the prompt [7]. In the preceding iteration of the COLIEE competition, the best performance on task 4 was achieved with the help of n-shot prompting as well [22]. In our system, we use 1-shot prompting, meaning that we provide a single example in the prompt. This example is a bar exam question from the past that is deemed to be most similar to the current question that the system needs to solve. We investigate two methods for determining similarity in questions: using Jaccard similarity and BM25.

In the first method, we determine the similarity between the current question and past questions by computing their Jaccard similarity [15]. Given two bar exam questions, A and B, the Jaccard similarity is defined as the ratio of their intersection to their union: $J(A, B) = \frac{|A \cap B|}{|A \cup B|}$.

In our second method, we select the most similar question using BM25 ranking, similar to the best-performing model in the COLIEE competition of last year [22]. BM25 (Best Matching 25) is a ranking function used in information retrieval to estimate the relevance of a document to a query [1]. It is based on the probabilistic retrieval framework and extends TF-IDF weighting with a saturation function and document length normalization.

Using either Jaccard similarity or BM25 ranking, we select the most similar question from the training data. This selected question is then added to the prompt.

3.3 Dictionary definitions

We augment the prompt by directly adding definitions of legally relevant terms to the prompt, in order to distinguish these terms from their more colloquial use. The aim is to find all legally relevant

Use these definitions:
- Juridical: Relating to administration of justice, or office of a judge.
- Court: In legislation. A legislative assembly.
- Apply: To make a formal request or petition, usually in writing, to a court, officer, board, or company, for the granting of some favor, or of some rule or order, which is within his or their power or discretion.
- Capacity: Legal capacity is the attribute of a person who can acquire new rights, or transfer rights, or assume duties, according to the mere dictates of his own will, as manifested in juristic acts, without any restraint or hindrance arising from his status or legal condition.

Figure 3: An example of our definitions that can be added to the prompt.

terms contained in the relevant (article, question) pair, and for each of these terms present a relevant legal definition. We do this by first performing TF-IDF on the entire corpus of the training set, then selecting the relevant scores using a threshold cutoff. We take this as the set of legally relevant terms. For each of these terms, we then look up their definitions in an open source legal dictionary (Black's Law Dictionary, 2nd Edition [6]). We then add the definition of the five most relevant terms to the prompt. We manually exclude specific terms where the legal definition is similar to the common connotation, such as "people". Furthermore, we remove extensive references to case law in the definitions. An example the type of dictionary definitions that we generate can be seen in Figure 3.

3.4 ANGELIC Domain Models

In previous research relating to the task of legal entailment, knowledge representations of legal articles were created and used to reason and determine legal entailment [32]. While promising, the approach was limited to questions about logical syllogisms and questions pertaining to single articles. In this study, we expand upon the approach by using knowledge representations of legal articles to reason about the questions and adding this information to the prompt.

Following previous research, we apply the ANGELIC II methodology [5] to create a knowledge representation for each legal article of the Japanese Civil Code. These representations of the legal articles are referred to as ADMs: ANGELIC Domain Models. In an ADM, a legal article S is represented as a type of hierarchical flowchart that can be used to reason about cases and come to a conclusion or verdict based on the factors of a case: the legally relevant fact patterns of the question that we are examining. To reason with an ADM, one thus needs to gather the required information from a particular statement Q such that we can ascribe the values of the ADM's factors. We ascribe the factors of the ADM by prompting a generative language model with questions about statement Q and parsing its output, following the methods of previous research [32]. The ADM then yields a verdict, which here is a necessary outcome based on the articles S and the facts of Q. Additionally, we also evaluate what claim is made in statement Q using the same language model.

In previous research, ADMs were crafted manually, as well as generated artificially using state-of-the-art language models. These artificially generated ADMs were not up to par with manually crafted ones [32]. In this research, we use the latest state-of-the-art models to generate our ADMs. We use OpenAI's o3-mini model (o3-mini-2025-01-31)[1], which was specifically designed with reasoning and coding in mind. In the provided COLIEE training data, there are 574 legal articles in total, of which o3-mini was able to generate 482 ADMs. Note that these ADMs are only checked based on whether they are executable, not whether the representation accurately matches the legal article. We use these 482 ADMs in our system.

Once an ADM has been selected and the factors have been ascribed, we can provide additional information about the bar exam question to the prompt. In this study, we add one of three observations to the prompt:

- If none of the factors apply to statement Q: '*Article S seems to be unrelated to statement Q*.'
- If the verdict of the ADM does not match the claim made in Q: '*By applying Article S, we cannot arrive at the claim made in statement Q*.'
- If the verdict of the ADM does match the claim made in Q: '*By applying Article S, we arrive at the claim made in statement Q*.'

For each legal article in the question, we fetch its associated ADM and use it to reason about statement Q. Each of the resulting observations is then added to the prompt. To ensure that the model does not over-rely on these observations provided by the ADM, we add the following preamble: 'Here is what experts say about this question:'.

3.5 IRAC

The last prompt element aims to direct the LLM towards a reasoning system that is also used by lawyers, namely by following the IRAC framework. IRAC stands for "Issue, Rule, Application, Conclusion". It is a frequently used framework for legal writing. Writers first identify the legal issue, describe the applicable legal rule, apply it to the relevant facts, and finally, conclude based on the performed reasoning. Metzler [21] indicates that IRAC mirrors syllogistic reasoning, providing a template that assists in addressing legal problems in a structured and easy-to-follow manner. Burton [8] and Turner [34] likewise discuss IRAC as a helpful tool in acquiring skills necessary to 'think like a lawyer'. Within the domain of AI & Law, Yu et al. [37] experimentally demonstrate that explicitly prompting LLMs with IRAC-based instructions may improve their ability to perform complex legal reasoning tasks, such as entailment assessments based on bar examination scenarios. We extend upon this study, by adding the following instructions seen in Figure 4 to our prompt. Systematic structures, like IRAC, may potentially provide scaffolding for computational modeling of legal reasoning, facilitating not only its accuracy but also the interpretability.

[1] https://platform.openai.com/docs/models/o3-mini

Use the IRAC structure to determine entailment:
(1) Major Premise (Article S)
  (a) Extract the core rule(s) from Article S. If there are conditions, exceptions, or logical connectors ("AND"/"OR"), break them down explicitly.
  (b) If multiple articles apply, determine their relation.
  (c) multiple articles apply at the same time, determine which one is most specific to the situation.
  (d) Determine the purpose of the rule, meaning the policy objective the rule aims to achieve.
(2) Minor Premise (Facts in Q)
  (a) Identify the key factual claim(s) in Statement Q.
  (b) Compare each fact against the conditions in S. Does Q satisfy all the required conditions for the rule to apply?
  (c) Be precise: do not assume extra facts beyond what is in Q.
(3) Contradictions
  (a) If Q contradicts S, entailment fails (Answer: No).
  (b) If Q introduces an exception or condition not present in S, then entailment fails (Answer: No).
  (c) If Q defines a legal concept, exact wording must match S to be entailed.
(4) Does S Necessarily Lead to Q?
  (a) Apply the legal rule(s) (major premise) to the facts (minor premise).
  (b) Ensure all conditions of S are met in Q.
  (c) If S states "If A, then B", then check whether A exists in Q. If A is missing or altered, entailment fails.
  (d) If S has an exception ("unless C"), check whether Q invokes that exception. If so, entailment fails.
(5) Conclusion
  (a) If Q necessarily follows from S: Answer: Yes.
  (b) If Q does not necessarily follow (even if it is consistent): Answer: No.
  (c) Check whether the sense of the conclusion is in line with the purpose of the rule. Reject entailment if Q contradicts or is against the law's objective.

Figure 4: The IRAC prompt element that can be added to the prompt.

3.6 Final prompt

In our system, we can select any combination of the five prompt elements to be included in the prompt that we provide to our language model. As an example, we show a prompt that uses all settings in Figure 5. In this example, we use guidelines 3, 4 and 5, 1-shot prompting, dictionary definitions, ADMs, and the IRAC structure.

4 Methods

In a set of experiments, we investigate the extent to which the prompt elements described in the previous section affect the performance of different generative language models on the legal entailment task. In all experiments, we make use of the Task 4 datasets of the COLIEE competition. This dataset is divided into several subsets, each consisting of the bar exam questions of a single year. For consistency reasons, we use the subset naming conventions used by the COLIEE. In chronological order, the entire dataset contains subsets H18 to H30, and R01 to R06. Note that in each experiment, we take the effect of time into account and thus only use bar exam questions from the past for selecting example cases when using 1-shot prompting. For example, for testing on test set R03, we only include subsets H18 to R03 in the training data.

There are three models that we evaluate in our experiments: Llama-3B-instruct[2], gpt-4o-mini (gpt-4o-mini-2024-07-18)[3], o3-mini (o3-mini-2025-01-31)[4]. We use the Llama model as it is an open-source generative language model that was released before July

[2]https://huggingface.co/meta-llama/Meta-Llama-3-8B-Instruct
[3]https://platform.openai.com/docs/models/gpt-4o-mini
[4]https://platform.openai.com/docs/models/o3-mini

The following is a Japanese bar exam question, where your task is to determine whether there is a legal entailment between legal article(s) S and a statement Q.
Legal article(s) S: $Article\_S$
Statement Q: $Statement\_Q$

Use all of the following guidelines to help you determine entailment:
$Guidelines$

Use the IRAC structure to determine entailment:
$IRAC\_Prompt$

Use these definitions:
$Dictionary\_Definitions$

Here is an example of a bar exam question with the correct answer:
Legal article(s): $Example\_Articles$
Statement: $Example\_Statement$
Answer: $Example\_Answer$.

Here is what experts say about this question:
$ADM\_Information$

Final question: is there legal entailment between article(s) S and statement Q? Think step by step and end your answer with: "Answer: yes or no"

Figure 5: An example of a prompt that contains all five prompt elements.

2024, and is thus eligible to be used in the COLIEE competition of 2025. The gpt-4o-mini model is a commercial language model that is commonly used in a variety of real-world applications. The o3-mini model is a language model that was specifically designed for reasoning purposes, and may thus yield a higher performance on the legal entailment task. The o3-mini model is therefore expected to perform best, followed up by the gpt-4o-mini model, and then the llama model. We keep the default setting of each model and use a fixed seed across all experiments.

We explore the prompt elements in a set of three experiments. In the first experiment, we evaluate the performance of each of our three models using the guidelines that we describe in Section 3.1. we explore every combination of the guidelines and evaluate the performance of each model on test set H29 of the dataset.

In the second experiment, the models are evaluated using all possible combinations of the five prompt elements. For this experiment, we only use the best guidelines (Section 3.1) based on the results of the previous experiment. For the 1-shot example, we evaluate the models using both Jaccard similarity and BM25 to investigate which of the two performs better. We evaluate the three models on all combinations using test set R02 of the dataset.

In the third experiment, we select the best performing prompt element combinations based on the previous experiment, and use these combinations to evaluate all three models on four different test sets: R03, R04, R05, and R06. In addition to the best performing settings, we also evaluate the performance of the models using a 'barebones' prompt with none of the five elements, and a prompt that uses all of the five elements.

We evaluate the models performance using accuracy, the default metric used in task 4 of the COLIEE competition. Additionally, we report F1-score, Matthew's Correlation Coefficient (MCC), precision, and recall, where appropriate. All metrics are scaled from 0 to 100, except for MCC, which is scaled from -100 to 100.

Guidelines	Accuracy llama	gpt-4o-mini	o3-mini
None	60.34	72.41	**86.21**
1	56.9	72.41	82.76
2	65.52	79.31	77.59
3	63.79	70.69	79.31
4	63.79	79.31	84.48
5	55.17	65.52	84.48
6	65.52	75.86	84.48
(3, 4, 5)	**74.14**	68.97	82.76
(4, 5, 6)	53.45	70.69	84.48
(1, 2, 3, 5)	58.62	**81.03**	74.14
(1, 2, 3, 4, 5, 6)	62.07	77.59	75.86

Table 1: A summary of the accuracy of each model on test set H29 for different combinations of guidelines. Best performances are shown in bold.

5 Results

5.1 Effects of guidelines

In our first experiment, we evaluated the effect of the different combinations of guidelines described in Section 3.1. Including the prompt with no guidelines, there are a total of 64 combinations that we evaluated. In Table 1, we show a summary of the results of this experiment, with the accuracy for each model on a selection of the different combinations of guidelines. The highest accuracies are shown in bold.

In the summary of results in Table 1, we show the effects of including every guideline on its own, as well as all of the guidelines at once. Additionally, we include the best-performing combination of guidelines for each model. For llama, this is the combination with guidelines 3, 4, and 5. For gpt-4o-mini, the best result is achieved with guidelines 1, 2, 3, and 5. For o3-mini, where the best result was gained with a prompt without any guidelines, we also include in Table 1 the combination of guidelines that yielded the second-best accuracy for that model, which is guidelines 4, 5, and 6.

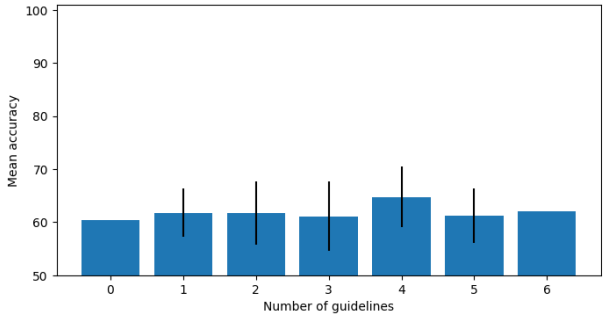
Additionally, we show the mean accuracy of each model versus the total number of guidelines in the prompt in Figure 6.

5.2 Effects of prompt elements

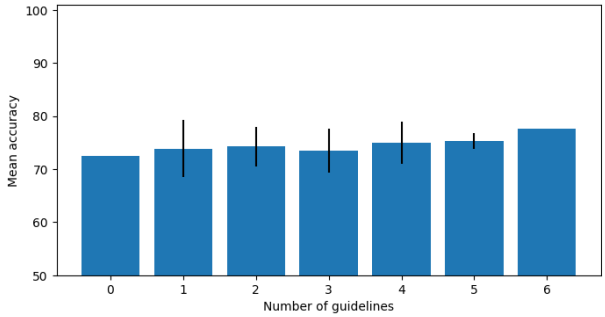
In our second experiment, we explore the effects of the five prompt elements discussed in Section 3. We only evaluate the best performing combination of guidelines for each model as shown in bold in Table 1, and the second-best performing guidelines for o3-mini.

A summary of the performance of each of the three models can be seen in Table 2, where we show the top 5 best performing prompt settings for each model, and the worst-performing setting for each model. Additionally, we show the performance of the 'barebones' prompt without any of the five elements, and of the prompt that includes all five elements. Note that we only show 7 rows instead of 8 rows for the llama and gpt-4o-mini model, as the prompt with all five elements is the fourth best performing prompt overall for both models. For each model, the settings and their performance in Table 2 are sorted by their Matthew's Correlation Coefficient.

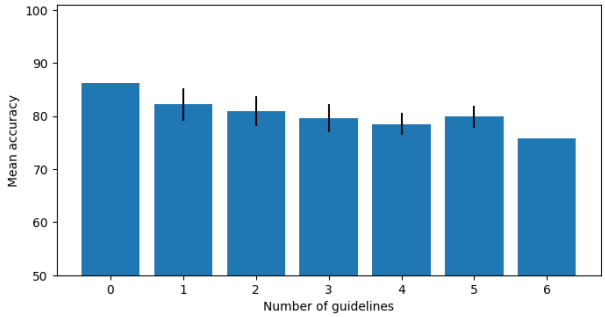
To evaluate the effect of the individual prompt elements, we show the accuracy of each model when using a single prompt element in Table 3. In Table 4, we show the mean accuracy across all combinations of settings where examples were used, split by whether the examples were selected using Jaccard similarity or BM25 ranking. Additionally, in Figure 7, we show the mean accuracy of each model versus the total number of prompt elements included in the prompt.



(a) llama-3B-instruct



(b) gpt-4o-mini



(c) o3-mini

Figure 6: The mean accuracy of each model on test set H29 versus the total number of guidelines used. Note that the y-axis is scaled from 50% to 100%.

5.3 Evaluating the models

In our third and last experiment, we evaluated each model across a number of different test sets, comparing the effects of a barebones prompt, a prompt with the optimal settings based on Table 2, and a prompt that contains all five prompt elements. The results of this experiment can be found in Table 5. We report the accuracy per test set, as well as the average accuracy, F1-score, and MCC across all test sets.

	Prompt settings					Performance				
Model	**Guidelines**	**Example**	**Dictionary**	**ADM**	**IRAC**	**accuracy**	**F1-score**	**MCC**	**Precision**	**Recall**
Llama	(3, 4, 5)	Jaccard	-	-	-	79.01	76.71	57.83	73.68	80.0
	-	Jaccard	-	Yes	Yes	77.78	78.57	57.03	86.84	71.74
	(3, 4, 5)	BM25	-	-	Yes	77.78	76.32	55.39	76.32	76.32
	(3, 4, 5)	BM25	Yes	Yes	Yes	76.54	75.32	53.0	76.32	74.36
	(3, 4, 5)	Jaccard	-	Yes	Yes	76.54	74.67	52.85	73.68	75.68
			...			...				
	-	-	-	-	-	62.96	68.09	30.68	84.21	57.14
			...			...				
	-	Jaccard	Yes	Yes	-	55.56	60.87	13.99	73.68	51.85
gpt-4o-mini	(1, 2, 3, 5)	-	-	Yes	Yes	91.36	91.14	82.96	94.74	87.8
	-	Jaccard	-	-	Yes	90.12	89.47	80.17	89.47	89.47
	(1, 2, 3, 5)	BM25	-	-	-	90.12	89.47	80.17	89.47	89.47
	(1, 2, 3, 5)	BM25	Yes	Yes	Yes	88.89	88.61	78.01	92.11	85.37
	-	BM25	Yes	-	Yes	88.89	88.31	77.75	89.47	87.18
			...			...				
	-	-	-	-	-	86.42	86.42	73.5	92.11	81.4
			...			...				
	-	BM25	Yes	-	-	80.25	80.00	60.88	84.21	76.19
o3-mini	(4, 5, 6)	Jaccard	-	-	-	95.06	94.74	90.09	94.74	94.74
	(4, 5, 6)	-	Yes	-	Yes	93.83	93.15	87.8	89.47	97.14
	(4, 5, 6)	Jaccard	-	-	Yes	93.83	93.15	87.8	89.47	97.14
	(4, 5, 6)	-	-	-	Yes	92.59	91.89	85.19	89.47	94.44
	-	Jaccard	Yes	-	-	92.59	92.11	85.13	92.11	92.11
			...			...				
	-	-	-	-	-	90.12	89.19	80.21	86.84	91.67
			...			...				
	(4, 5, 6)	Jaccard	Yes	Yes	Yes	85.19	82.86	70.78	76.32	90.62
			...			...				
	-	-	-	Yes	Yes	70.37	61.29	41.94	50.0	79.17

Table 2: Summary of the performance of all three models on test set R02 using different prompt settings sorted by MCC.

	Model		
	llama	**gpt-4o-mini**	**o3-mini**
Barebones	62.96	86.42	90.12
Guidelines	67.90	82.30	89.50
Example	70.37	84.47	88.30
Dictionary	64.20	85.19	90.12
ADM	62.96	82.72	70.37
IRAC	61.73	87.65	88.89

Table 3: The accuracy of each model on test set R02 using a single prompt element.

Similarity	Model		
measure	**llama**	**gpt-4o-mini**	**o3-mini**
Jaccard	70.14	85.48	86.88
BM25	70.29	84.44	88.17

Table 4: The mean accuracy of each models on test set R02 when using either Jaccard similarity or BM25 ranking across all combinations of prompt elements.

6 Discussion

In Table 1, where we display the results of the experiment investigating the effect of including all possible combinations of guidelines in the prompt, we see that o3-mini is the best performing model 'out of the box' when no guidelines are added to the prompt. The gpt-4o-mini model comes in second, and the llama model third. This is an expected result, and we generally see this difference in performance between the models across all results.

We see little consistency in the performance versus the guidelines used across the three models. For instance, including only guideline 3 in the prompt seems to improve performance for the llama model when compared to the barebones prompt, but it decreases performance for the gpt-4o-mini and o3-mini models. All combinations of guidelines seem to decrease performance for o3-mini, whereas a maximal increase in performance of 13.8 and 8.62 percentage points can be achieved by including particular guidelines for llama and gpt-4o-mini, respectively.

In Figure 6, where we plot the mean accuracy versus the number of guidelines included in the prompt, we observe different trends for the different models. For the llama model (Figure 6a), the mean accuracy fluctuates slightly for any number of guidelines. For gpt-4o-mini (Figure 6b), including guidelines seems to increase

	Prompt settings					Accuracy per test set				Overall Performance		
model	**Guidelines**	**Example**	**Dictionary**	**ADM**	**IRAC**	**R03**	**R04**	**R05**	**R06**	**Accuracy**	**F1-score**	**MCC**
llama	-	-	-	-	-	**62.39**	61.39	64.22	**66.22**	63.36	70.73	**27.88**
	(3, 4, 5)	Jaccard	-	-	-	52.29	60.40	**65.14**	60.81	59.54	62.23	18.76
	(3, 4, 5)	BM25	Yes	Yes	Yes	**62.39**	**62.38**	61.47	64.86	62.6	63.16	25.27
gpt-4o-mini	-	-	-	-	-	**81.65**	**78.22**	**79.82**	79.73	79.90	81.06	**59.70**
	(1, 2, 3, 5)	-	-	Yes	Yes	74.31	71.29	74.31	72.97	73.28	73.68	46.67
	(1, 2, 3, 5)	BM25	Yes	Yes	Yes	77.98	71.29	78.90	**81.08**	77.10	77.94	54.14
o3-mini	-	-	-	-	-	85.32	82.18	**87.16**	**83.78**	84.73	85.58	69.40
	(4, 5, 6)	Jaccard	-	-	-	**89.91**	**85.15**	**87.16**	**83.78**	86.77	87.50	**73.49**
	(4, 5, 6)	Jaccard	Yes	Yes	Yes	86.24	77.23	78.90	79.73	80.66	80.61	61.66

Table 5: The performance of all three models across different test sets using a barebones prompt, an optimal prompt (based on Table 2), and a prompt that includes all five prompt elements. Best results are shown in bold.

performance, but there seems to be little difference in mean accuracy between the amount of guidelines used. Lastly, for o3-mini (Figure 6c), including any guidelines appears to lower accuracy, and generally, more guidelines yield worse results.

Taking the best guidelines from Table 1, we investigated all prompt element combinations and showed a summary of their results in Table 2. In this table, we can see that for every model, the best-performing prompts always include the guidelines. For both the llama and o3-mini model, the best performing prompt settings are a combination of the guidelines and an example selected using Jaccard similarity. This seems to indicate that these two prompt elements are beneficial to the performance of the model. The best performing settings for gpt-4o-mini includes the guidelines, ADM, and IRAC instructions.

The prompt that includes all five prompt elements performs well for both llama and gpt-4o-mini: they are the fourth-best-performing prompts for both models. For the o3-mini model this is not the case, and the prompt that includes all five elements performs worse than the barebones prompt.

When evaluating the effect of each individual prompt element in Table 3, we see that each of the five prompt elements has a different effect depending on what model is used. For the llama model, most elements have a positive or neutral effect on the performance, except for the IRAC element, which decreases accuracy slightly. For the gpt-4o-mini model on the other hand, the IRAC element is the only element that increases performance when compared to a barebones prompt without prompt elements. For the o3-mini model, every element by itself seems to decrease performance slightly, and performance remains the same for the dictionary prompt element. comparing this to the results of Table 1, we can say that a combination of prompt elements seems to work best for this particular task.

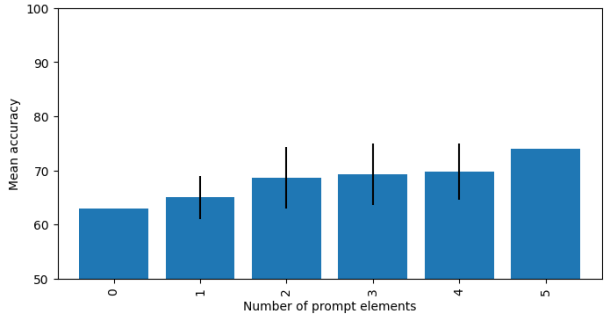
In Figure 7, we see that the effect of the number of prompt elements added to the prompt on the performance differs per model as well. For the llama model, the performance on average increases for every element added to the prompt. For the gpt-4o-mini model, the effect is less pronounced: the performance first decreases, but then increases again until it reaches the same mean accuracy as the barebones prompt. For o3-mini, the average performance decreases with 1 or two elements, but remains roughly the same for 3, 4, 5, or 6 elements.

In terms of similarity measures for selecting the most relevant examples, we can see in Table 2 that the best-performing prompts tend to use Jaccard similarity. However, on average, the Jaccard similarity and BM25 ranking seem to yield similar results across all combinations, as evident by Table 4.

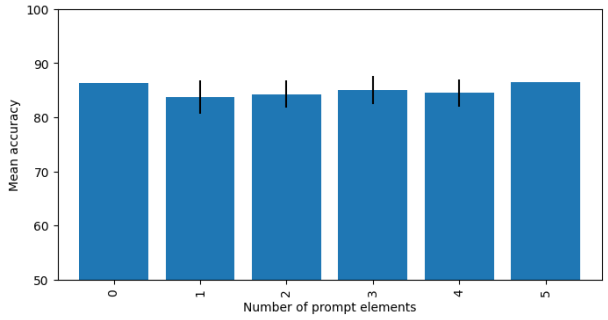
After selecting the optimal prompt elements in our second experiment, we evaluated the performance of each model using these optimal settings in Table 5. On average, only the o3-mini model performs better using the 'optimal' settings (73.49 MCC) when compared to the barebones prompt (69.40 MCC) across all test sets. For the other two models, the barebones prompt is, on average, the best-performing prompt. When we examine the performance per test set, however, we see that this is not always the case: what the best-performing prompt is differs depending on the test sets. The performance across test sets varies quite widely for each prompt setting as well.

The variance in performance across test sets can be attributed to the wide problem space of the legal entailment task. Legal reasoning is a difficult and multi-faceted task, as described in Section 2. Because of this, a prompt elements might work well for a certain question but not for another. For example question *B* (Figure 1b), guideline 2 from Figure 2 is essential, as it prescribes the model to interpret the language in the article exactly, and that it should not expand its scope. In other words, it should not assume that intellectual capacity and mental capacity are the same terms. For example question *A* (Figure 1a), however, guideline 2 might have a negative effect, as it could prevent the model from using other information than the information present in the article, such as the definition of the term 'mutatis mutandis'. While a lawyer understands when to apply what guideline, large language models in our experiment were prone to hyper-fixate on instructions. This hyper-fixation can limit the scope of the model's reasoning capabilities, thus yielding worse results in tasks with a wide problem space.

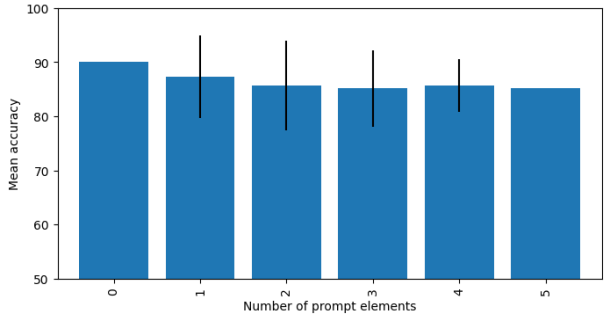
The better models (gpt-4o-mini and o3-mini) seem to perform rather well with just the barebones prompt. They seem to have an almost inherent ability to perform legal reasoning to a certain degree, if we narrowly define reasoning capabilities as the ability to perform reasoning tasks well. By including all of the additional prompt elements, the model may fixate on the elements themselves, consequently limiting the model's inherent ability to perform the legal entailment task, thus leading to a worse performance.



(a) llama-3B-instruct



(b) gpt-4o-mini



(c) o3-mini

Figure 7: The mean accuracy of each model on test set R02 versus the total number of elements used in the prompt. Note that the y-axis is scaled from 50% to 100%.

Using a single language model to solve all types of questions may thus not be the best way forward. Instead, in future research we could opt to use an ensemble of expert models, where each model in the ensemble is specialized in answering a single type of question based on a specific type of legal reasoning. Such a system should first classify the question into a particular question type and then use the appropriate model to solve the question.

Some further extensions of this research involves the use of other or different prompt elements. For instance, we only explored the use of 1-shot prompting in this study, and did not investigate the use of more examples. We should also note that we evaluated our models purely using performance, based on their final answer for every question in the task. We did not evaluate the explanations as to why the models' came to their decisions. Furthermore, even if the model gives the correct final answer and plausible explanations, there is no guarantee that this explanation matches the actual internal reasoning process that occurs, as the models are black boxes [31]. A more thorough evaluation of the reasoning process is therefore required to make actual claims about the reasoning of large language models.

7 Conclusion

In this study, we design five types of prompt elements for large language models based on legal expert knowledge and investigate the effect of these prompt elements in a legal entailment task. We relate each prompt element to the types of legal reasoning that is commonly used by lawyers, and systematically evaluate each element in a set of experiments. We show that the elements can improve performance on the COLIEE legal entailment task, but due to the large problem space of the task, elements that improve performance in cases can decrease performance in other cases. In future research, we aim to create an ensemble of expert models that classifies each entailment task into sub-tasks, and uses different specialized models for each sub-task. By working together with legal experts, and incorporating more domain knowledge into AI systems, we hope to increase their capabilities for legal reasoning.

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.

References

[1] L. Liu and M. T. Özsu, (Eds.) 2009. *Bm25. Encyclopedia of Database Systems.* Springer US, Boston, MA, 257–260.

[2] M. Araszkiewicz and J. Šavelka. 2011. Two methods for representing judicial reasoning in the framework of coherence as constraint satisfaction. In *Legal Knowledge and Information Systems. JURIX: The Twenty-Fourth Annual Conference* (Frontiers in Artificial Intelligence and Applications). K. Atkinson, (Ed.) Vol. 235. IOS Press, Amsterdam, Netherlands, 165–166.

[3] K.D. Ashley. 2011. Precedent and legal analogy. In *Handbook in Legal Reasoning and Argumentation.* G. Bongiovanni, D. Postema, A. Rotolo, G. Sartor, C. Valentini, and D. Walton, (Eds.) Springer, Dordrecht, Netherlands, 673–710.

[4] B. Askeland. 2020. The potential of abductive legal reasoning. *Ratio Juris*, 33, 1, 66–81.

[5] K. Atkinson and T. Bench-Capon. 2023. ANGELIC II: an improved methodology for representing legal domain knowledge. In *ICAIL '23: Proceedings of the Nineteenth International Conference for Artificial Intelligence and Law.* ACM, Braga, Portugal, (June 2023).

[6] H.C. Black. 1910. *Black's Law Dictionary (2nd Edition).* Extracted in Json. West Publishing Company. https://gist.github.com/medelman17/55bf480caafbfcc6e9f9d22c273cf2c4.

[7] T. Brown et al. 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.

[8] K. Burton. 2017. 'Think like a lawyer' using a legal reasoning grid and criterion-referenced assessment rubric on irac (issue, rule, application, conclusion). *Journal of Learning Design*, 10, 2, 57–68.

[9] P.C. Ellsworth. 2005. *Legal reasoning.* K. Holyoak and R. Morrison, (Eds.) Cambridge University Press, New York, (Jan. 2005), 685–704.

[10] R. Goebel, Y. Kano, M.Y. Kim, J. Rabelo, K. Satoh, and M. 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.

[11] R. Goebel, Y. Kano, M.Y. Kim, J. Rabelo, K. Satoh, and M. Yoshioka. 2023. Summary of the Competition on Legal Information, Extraction/Entailment (COLIEE) 2023. en. In *Proceedings of the Nineteenth International Conference on Artificial*

Intelligence and Law. ACM, Braga Portugal, (June 2023), 472–480. Retrieved Jan. 29, 2025 from.

[12] G. Governatori, T. Bench-Capon, B. Verheij, et al. 2022. Thirty years of artificial intelligence and law: the first decade. *Artificial Intelligence and Law*, 30, 481–519.

[13] 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.

[14] H. L. A. Hart. 1958. Positivism and the separation of law and morals. *Harvard Law Review*, 71, 4, 593–629.

[15] P. Jaccard. 1901. Etude de la distribution florale dans une portion des alpes et du jura. *Bulletin de la Societe Vaudoise des Sciences Naturelles*, 37, (Jan. 1901), 547–579.

[16] C. Jiang and X. Yang. 2023. Legal syllogism prompting: teaching large language models for legal judgment prediction. In *Proceedings of the Nineteenth International Conference on Artificial Intelligence and Law* (ICAIL '23). Association for Computing Machinery, Braga, Portugal, 417–421.

[17] D.M. Katz, M.J. Bommarito, S. Gao, and P. Arredondo. 2024. Gpt-4 passes the bar exam. *Philosophical Transactions of the Royal Society A*, 382, 2270, 20230254.

[18] L.A. Kornhauser. 2011. Choosing ends and choosing means: teleological reasoning in law. In *Handbook in Legal Reasoning and Argumentation*. G. Bongiovanni, D. Postema, A. Rotolo, G. Sartor, C. Valentini, and D. Walton, (Eds.) Springer, Dordrecht, Netherlands, 387–412.

[19] E.H. Levi. 1948. An introduction to legal reasoning. *University of Chicago Law Review*, 15, 501–574.

[20] E. Martínez. 2024. Re-evaluating gpt-4's bar exam performance. *Artificial Intelligence and Law*, (Mar. 2024).

[21] J. Metzler. 2003. The importance of irac and legal writing. *University of Detroit Mercy Law Review*, 80, 501–503.

[22] P. Nguyen, C. Nguyen, H. Nguyen, M. Nguyen, A. Trieu, D. Nguyen, and L. Nguyen. 2024. Captain at coliee 2024: large language model for legal text retrieval and entailment. In *New Frontiers in Artificial Intelligence*. T. Suzumura and M. Bono, (Eds.) Springer Nature Singapore, Singapore, 125–139.

[23] R. A. Posner. 1992. Legal reasoning from the top down and from the bottom up: the question of unenumerated constitutional rights. *University of Chicago Law Review*, 59, 433–450.

[24] H. Prakken and G. Sartor. 2004. The three faces of defeasibility in the law. *Ratio Juris*, 17, 1, 118–139.

[25] 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, (Apr. 2022), 111–133.

[26] G. Sartor. 2011. Defeasibility in law. In *Handbook in Legal Reasoning and Argumentation*. G. Bongiovanni, D. Postema, A. Rotolo, G. Sartor, C. Valentini, and D. Walton, (Eds.) Springer, Dordrecht, Netherlands, 315–364.

[27] G. Sartor, M. Araszkiewicz, K. Atkinson, et al. 2022. Thirty years of artificial intelligence and law: the second decade. *Artificial Intelligence and Law*, 30, 521–557.

[28] A. Scalia and B.A. Garner. 2012. *Reading Law: The Interpretation of Legal Texts*. Thomson/West, St. Paul, MN.

[29] F. Schauer. 1988. Formalism. *Yale Law Journal*, 97, 4, 509–548.

[30] M. J. Sergot, F. Sadri, R. A. Kowalski, F. Kriwaczek, P. Hammond, and H. T. Cory. 1986. The british nationality act as a logic program. *Communications of the ACM*, 29, 5, 370–386.

[31] C. Steging, S. Renooij, and B. Verheij. 2021. Discovering the rationale of decisions: Towards a method for aligning learning and reasoning. In *ICAIL '21: Proceedings of the Eighteenth International Conference for Artificial Intelligence and Law*. J. Maranhão and A. Wyner, (Eds.) ACM, São Paolo, Brazil, (June 2021), 235–239.

[32] 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).

[33] C.R. Sunstein. 1992. On analogical reasoning commentary. *Harvard Law Review*, 106, 741–791.

[34] T. Turner. 2012. Finding consensus in legal writing discourse regarding organizational structure: a review and proposal. *Legal Writing: Journal of the Legal Writing Institute*, 18, 165–206.

[35] S. Villata, M. Araszkiewicz, K.D. Ashley, T. Bench-Capon, L.K. Branting, J. G. Conrad, and A. Wyner. 2022. Thirty years of artificial intelligence and law: the third decade. *Artificial Intelligence and Law*, 30, 4, (Dec. 2022), 561–591.

[36] D. Walton. 2011. Introduction. In *Handbook in Legal Reasoning and Argumentation*. G. Bongiovanni, D. Postema, A. Rotolo, G. Sartor, C. Valentini, and D. Walton, (Eds.) Springer, Dordrecht, Netherlands, ix–xxiii.

[37] F. Yu, L. Quartey, and F. Schilder. 2022. Legal prompting: teaching a language model to think like a lawyer. (2022). arXiv: 2212.01326.

UQLegalAI@COLIEE2025: Advancing Legal Case Retrieval with Large Language Models and Graph Neural Networks

Yanran Tang
yanran.tang@uq.edu.au
The University of Queensland
Brisbane, Australia

Ruihong Qiu
r.qiu@uq.edu.au
The University of Queensland
Brisbane, Australia

Zi Huang
helen.huang@uq.edu.au
The University of Queensland
Brisbane, Australia

Abstract

Legal case retrieval plays a pivotal role in the legal domain by facilitating the efficient identification of relevant cases, supporting legal professionals and researchers to propose legal arguments and make informed decision-making. To improve retrieval accuracy, the Competition on Legal Information Extraction and Entailment (COLIEE) is held annually, offering updated benchmark datasets for evaluation. This paper presents a detailed description of CaseLink, the method employed by UQLegalAI, the second highest team in Task 1 of COLIEE 2025. The CaseLink model utilises inductive graph learning and Global Case Graphs to capture the intrinsic case connectivity to improve the accuracy of legal case retrieval. Specifically, a large language model specialized in text embedding is employed to transform legal texts into embeddings, which serve as the feature representations of the nodes in the constructed case graph. A new contrastive objective, incorporating a regularization on the degree of case nodes, is proposed to leverage the information within the case reference relationship for model optimization. The main codebase used in our method is based on an open-sourced repo of CaseLink [18]: https://github.com/yanran-tang/CaseLink.

CCS Concepts

• **Information systems → Specialized information retrieval**.

Keywords

Information Retrieval, Legal Case Retrieval, Graph Neural Networks

ACM Reference Format:
Yanran Tang, Ruihong Qiu, and Zi Huang. 2025. UQLegalAI@COLIEE2025: Advancing Legal Case Retrieval with Large Language Models and Graph Neural Networks. In *Proceedings of COLIEE 2025 workshop, June 20, 2025, Chicago, USA*. ACM, New York, NY, USA, 5 pages.

1 Introduction

In legal domain, a precedent refers to a judicial decision that serve as an example or authority when giving judgment to future similar cases, which ensures fairness, consistency and predictability of judgments in legal system. However, identifying relevant precedents within large legal databases is a time-consuming task, significantly reducing the work efficiency of legal practitioners. Therefore, the Competition on Legal Information Extraction and Entailment (COLIEE) [10] is held annually to encourage the competition participants to develop highly accurate legal case retrieval models. There are five task in COLIEE 2025, where Task 1 and Task 2 are case law tasks, Task 3 and Task 4 are statute law tasks and Task 5 is a pilot task that focuses on judgment of civil cases.

In COLIEE 2025, Task 1 is a legal case retrieval task of case law system, aimed at returning 'noticed cases' from a large case collection for a given query case. Specifically, a case called 'noticed' to a query case means the case is referenced by the query case. The provided cases of Task 1 are all from the Federal Court of Canada. As legal case retrieval is a basic and essential task in COLIEE, previous teams have proposed high-accuracy retrieval models. In COLIEE 2024, TQM [5] team achieves the first place in Task 1 by exploring various lexical and semantic retrieval models. THUIR [6] team develops a structure-aware pre-trained language model called SAILER to improve the model understanding ability of legal cases, which ranks the first in COLIEE 2023. While UA [11] team leverages a transformer-based model for generating paragraph embeddings and a gradient boosting classifier to decide a case is noticed or not, which ranks the first in COLIEE 2022.

In this paper, our novel CaseLink [18] model utilised in COLIEE 2025 Task 1 is proposed to further enhance the retrieval accuracy by leveraging case connectivity relations and graph-based model. Firstly, the training set and test set are transferred into two Global Case Graphs (GCG) by exploiting the Case-Case and Case-Charge and Charge-Charge relationships of cases and charges. With the constructed graph, a large language model specialized in text embedding is utilized to convert legal texts into embeddings as the node features of GCG. To leverage the connected relationships in GCG, a graph neural network module is used to generate the case representation. A contrastive loss and a degree regularisation are designed to train the CaseLink model. Our team, UQLegalAI, ranked as the second highest team in Task 1 with a F1 score of 0.2962.

2 Related Work

Legal case retrieval aims to retrieve a set of relevant cases for a query case within a large legal case dataset. The recent legal case retrieval can be roughly divided into lexical models, semantic models and graph-based models. The traditional lexical models like TF-IDF [2], BM25 [12] and LMIR [9] utilise term frequency to calculate the case similarity score. While the semantic models such as BERT-PLI [13] and PromptCase [15] are both using language model to generate case embedding.

Unlike lexical and semantic models, graph-based models leverage graph structures and Graph Neural Networks (GNNs) to enhance the performance of legal case retrieval. For example, CaseGNN[17]

COLIEE 2025, Chicago, USA

Table 1: Statistics of Task 1 dataset.

COLIEE 2025 Task 1	train	test
Language	English	
# Query	1678	400
# Candidates	7350	2159
# Avg. relevant cases	4.1007	4.3975
Avg. length (# token)	28865	31250
Largest length (# token)	650534	681027

and CaseGNN++[16] exploit the relations of legal elements of a case to construct a fact graph and a legal issue graph for each case as well as use GNN to generate case graph representations. SLR [7] and CFGL-LCR [22] integrate external knowledge graphs with GNNs to improve retrieval performance. In contrast to these graph-based models, the CaseLink [18] model utilised in this paper fully exploits the connectivity relationships among cases of a large legal dataset to enhance case retrieval effectiveness.

3 Preliminary

3.1 Task Definition

Task 1 of COLIEE 2025 is a legal case retrieval task that focuses on case law. Given a query case $q \in \mathcal{D}$, and a set of n cases $\mathcal{D} = \{d_1, d_2, ..., d_n\}$, the task of legal case retrieval is to extract a set of relevant cases $\mathcal{D}^* = \{d_i^* | d_i^* \in \mathcal{D} \wedge relevant(d_i^*, q)\}$. The $relevant(d_i^*, q)$ indicates that d_i^* is a relevant case to the query case q. In case law, the above relevant cases refer to the precedents, which are the prior cases referenced by the query case.

3.2 Dataset

The cases in the Task 1 dataset are sourced entirely from the Federal Court of Canada. The statistics for the Task 1 dataset are presented in Table 1. These statistics reveal that the number of queries and candidates in the training set are approximately three times greater than that in the test set. Additionally, the average number of relevant cases per query in both the training and test sets are close to four, suggesting that the level of difficulty for both sets is comparable. Furthermore, the average token count per case for both the training and test sets is approximately 30,000. Notably, the longest case contains up to 680,000 tokens, highlighting the challenges associated with processing and comprehending long cases.

3.3 Evaluation metric

In COLIEE 2025 Task 1, the micro-average of precision, recall, and F-measure are utilised as the evaluation metic as follows:

$$\text{Precision} = \frac{\text{the number of correctly retrieved cases for all queries}}{\text{the number of retrieved cases for all queries}}, \tag{1}$$

$$\text{Recall} = \frac{\text{the number of correctly retrieved cases for all queries}}{\text{the number of relevant cases for all queries}}, \tag{2}$$

$$\text{F-measure} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}. \tag{3}$$

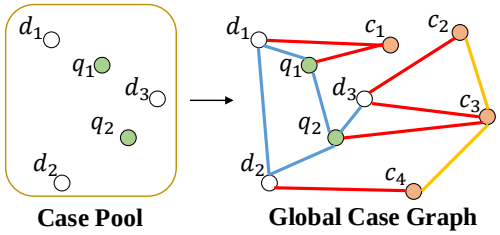


Figure 1: An example of a Global Case Graph is shown, where green nodes represent the two query cases q_1 and q_2, white nodes denote candidate cases $d_1 \sim d_3$ and orange nodes correspond to legal charges $c_1 \sim c_4$. The solid lines indicate the edges: Case-Case edges are shown in blue, Case-Charge edges in red, and Charge-Charge edges in yellow.

4 Method

4.1 Global Case Graph

In this paper, the construction method of the Global Case Graph (GCG) is adopted from our previous work CaseLink [18]. Specifically, GCG is represented as $G = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ and $\mathcal{E}$ denote the set of nodes and the set of edges, respectively. The node set $\mathcal{V}$ comprises both case nodes d, q as well as charge nodes c. The edge set $\mathcal{E}$ encompasses three types of edges: Case-Case edges, Case-Charge edges, and Charge-Charge edges. An example of GCG is shown in Figure 1.

4.1.1 Nodes. In GCG, the case nodes refer to the cases in the $\mathcal{D}$, which includes both the query cases q and candidate cases d. The charge nodes are derived from a list specified in the Federal Courts Act and Rules of Canada[1], presented as $C = \{c_1, c_2, ..., c_m\}$.

4.1.2 Edges. To effectively utilise intrinsic case connectivity relations, three edge connection strategies are exploited in GCG:

• Case-Case Edge. The construction of GCG aims to establish edges between cases that employ intrinsic connectivity. Therefore, the high similarity cases measured by BM25 [12] will be linked as neighbour nodes. The adjacency matrix of Case-Case edges $\mathbf{A}_d \in \mathbb{R}^{n \times n}$ is denoted as:

$$\mathbf{A}_{d_{ij}} = \begin{cases} 1 & \text{forTopK}(\text{BM25}(d_i, d_j | d_i, d_j \in \mathcal{D})), \\ 0 & \text{forOthers}, \end{cases} \tag{4}$$

where d_i and d_j are two cases in $\mathcal{D}$. TopK retrieves the top K most similar cases from a given list of BM25 case similarity scores.

• Charge-Charge Edge. In legal system, natural relationships exist among different legal charges. Therefore, connecting similar charges can effectively enhance case representation learning. The Charge-Charge edges symmetric adjacency matrix $\mathbf{A}_c \in \mathbb{R}^{m \times m}$, comprising m charges is defined as:

$$\mathbf{A}_{c_{ij}} = \begin{cases} 1 & \text{for} \quad \text{Sim}(\mathbf{x}_{c_i}, \mathbf{x}_{c_j} | c_i, c_j \in \mathcal{V}) > \delta, \\ 0 & \text{for} \quad \text{Others}, \end{cases} \tag{5}$$

where Sim is the similairty calculation fuction such as cosine similarity, $c_i \in \mathcal{V}$, $c_j \in \mathcal{V}$ are two charge nodes with the node features

[1] https://www.fct-cf.gc.ca/en/pages/law-and-practice/acts-and-rules/federal-court/

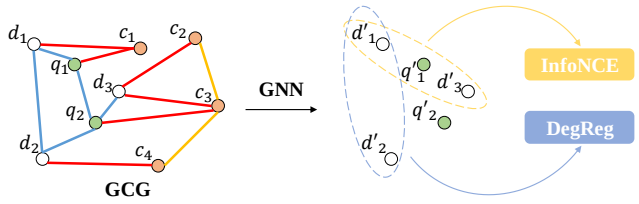


Figure 2: The overall framework of CaseLink [18].

$\mathbf{x}_{c_i} \in \mathbb{R}^d$, $\mathbf{x}_{c_j} \in \mathbb{R}^d$. The number of Charge-Charge edges is regulated by a similarity score threshold δ.

• Case-Charge Edge. A Case-Charge edge is established when a charge name appears in the case, which shows the high correlation between the charge and case. The adjacency matrix of Case-Charge edges $\mathbf{A}_b \in \mathbb{R}^{m \times n}$ is designed as:

$$\mathbf{A}_{b_{ij}} = \begin{cases} 1 & \text{for} \quad t_{c_i} \text{ appears in } t_{d_j}, \\ 0 & \text{for} \quad \text{Others}, \end{cases} \tag{6}$$

where t_{c_i} is the text of charge i, t_{d_j} is the text of case j.

• Overall Adjacency Matrix. To directly combine the Case-Case edges, Case-Charge edges, and Charge-Charge edges, the GCG overall adjacency matrix $\mathbf{A} \in \mathbb{R}^{(n+m)\times(n+m)}$ is undirected and unweighted, which is denoted as:

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_d & \mathbf{A}_b^\intercal \\ \mathbf{A}_b & \mathbf{A}_c \end{bmatrix}, \tag{7}$$

where $\mathbf{A}_b^\intercal$ denotes the transpose of the adjacency matrix.

4.1.3 Embedding Initialisation with Large Language Models. Given the high quality of text embedding encoded by large language models in recent text embedding benchmark such as Multilingual Text Embedding Benchmark (MTEB) [8], LLM is employed to encode the nodes into embedding features for GCG in this paper. The encoding process is denoted as:

$$\mathbf{x} = \text{LLM}(t), \tag{8}$$

where t is the text of a case or a charge, $\mathbf{x} \in \mathbb{R}^d$ is the generated text embedding as the node feature in GCG. LLM can be any LLM that encodes the texts into embeddings. In this paper, the top-ranked open-source model for legal retrieval task in MTEB, e5-mistral-7b-instruct [21], is chosen to be the LLM text encoder.

4.2 CaseLink

The CaseLink module is adopted from our previous work [18], which demonstrated strong performance on the legal case retrieval task. During training, the training queries, candidate cases, and legal charges are integrated into a GCG. A graph neural network (GNN) module is then applied to update the node features within the GCG. The updated features of the query and candidate nodes are subsequently used in two training objectives: contrastive learning via the InfoNCE loss and the degree regularization (DegReg) objective, to optimize the CaseLink model. The overall framework of CaseLink is demonstrated in Figure 2.

4.2.1 Graph Neural Network. With the constructed GCG and the encoded initial graph features, a GNN model is leveraged to generate the case representation as:

$$\mathbf{H} = \text{GNN}_\theta(\mathbf{X}, \mathbf{A}), \tag{9}$$

where $\mathbf{X}$ is the feature matrix consists of the node features $\mathbf{x}$, $\mathbf{H} \in \mathbb{R}^{n \times d}$ are the representations of cases in GCG, $\mathbf{h}_i \in \mathbb{R}^d$ is the case representation of case i and θ is the model parameter of GNN. GNN_θ can be any graph neural network models, such as GCN [4], GAT [20] or GraphSAGE [1].

4.2.2 InfoNCE Objective. A widely adopted approach for training the GNN model in legal case retrieval is to utilize contrastive learning based on the InfoNCE objective [19]:

$$\ell_{\text{InfoNCE}} = -\log \frac{e^{\text{Sim}(\mathbf{h}_q, \mathbf{h}_{d^+})/\tau}}{e^{\text{Sim}(\mathbf{h}_q, \mathbf{h}_{d^+})/\tau} + \sum_{i=1}^{p} e^{\text{Sim}(\mathbf{h}_q, \mathbf{h}_{d_i^-})/\tau}}, \tag{10}$$

where a relevant case d^+ and p irrelevant cases d^- are sampled for a given query case q with τ denoting the temperature parameter. Cosine similarity is chosen as the Sim function here. The positive samples correspond to the ground-truth labels, and the easy negative samples are randomly sampled from training set. Specifically, the hard negative samples are randomly selected from the negative cases of Top-K BM25 [12] ranking list.

4.2.3 Degree Regularisation. To implement degree regularisation for the candidate nodes, the pseudo-adjacency matrix is defined based on the updated node features after GNN computation as:

$$\hat{\mathbf{A}}_{ij} = \cos(\mathbf{h}_i, \mathbf{h}_j), \tag{11}$$

where $\mathbf{h}_i$ and $\mathbf{h}_j$ are the updated features of case node i and j in the case pool $\mathcal{D}$. The matrix $\hat{\mathbf{A}} \in \mathbb{R}^{n \times n}$ indicates a fully connected situation. And the degree regularisation is conducted on this $\hat{\mathbf{A}}$ only for candidate cases:

$$\ell_{\text{DegReg}} = \sum_{i=1}^{o} \sum_{j=1}^{n} (\hat{\mathbf{A}}_{ij}), \tag{12}$$

where o is candidate number.

4.2.4 Overall Objective. The overall objective is designed as:

$$\ell = \ell_{\text{InfoNCE}} + \lambda \cdot \ell_{\text{DegReg}}, \tag{13}$$

where λ is the coefficient of degree regularisation.

4.3 Inference

During testing on $\mathcal{D}_{\text{test}}$, the similarity score $s_{(q,d)}$ is calculated as:

$$s_{(q,d)} = \text{Sim}(\mathbf{h}_q, \mathbf{h}_d), \tag{14}$$

where $\mathbf{h}_q$ and $\mathbf{h}_d$ are the representations of query q and candidate d generated by CaseLink. Final top 5 ranking candidates are retrieved.

4.4 Post-processing

After the calculation of similarity score, the following post-processing strategies are conducted for improving retrieval accuracy.

Table 2: Top 10 runs of Task 1.

Team	Submission	Precision	Recall	F1
JNLP	jnlpr&fe2.txt	0.3042	0.3735	0.3353
JNLP	jnlpr&fe1.txt	0.2945	0.3667	0.3267
UQLegalAI	uqlegalair3.txt	0.2908	0.3019	0.2962
UQLegalAI	uqlegalair2.txt	0.2903	0.3013	0.2957
UQLegalAI	uqlegalair1.txt	0.2886	0.2996	0.2940
AIIR Lab	task1.aiirmpmist5.txt	0.2040	0.2319	0.2171
NOWJ	prerank_dense_bge-rerank_ bge_ft_llm2vec_major_vote.txt	0.1670	0.2445	0.1984
AIIR Lab	task1.aiircombmnz.txt	0.2317	0.1580	0.1879
AIIR Lab	task1.aiirmpmist3.txt	0.2308	0.1575	0.1872
NOWJ	prerank_dense_bge-rerank_bge_ft.txt	0.1605	0.1825	0.1708

4.4.1 Two-stage Ranking. To harness the strengths of statistical methods, the candidate lists are initially reduced to ten cases using the BM25 retrieval algorithm. Considering that the average number of relevant cases in the training set is 4.1, the number of final retrieved cases per query is fixed at five during the testing phase, based on the calculated CaseLink similarity scores.

4.4.2 Year Filtering. It is known that precedents is the prior judicial decision that serves as an example for future cases, which means that the cited precedents should happen before the given query case. Therefore, given a query, only cases with earlier dates than the query case are considered as candidates, while those with later dates are excluded in this paper. Specifically, the latest date appearing in a case is taken as its representative trial date.

5 Experiments and Results

5.1 Implementation

The training batch size is selected from {256, 512, 1024, 1678}. The default GNN model is GAT [20] with number of layers chosen from {1,2,3}. The dropout [14] rate is selected from {0.1, 0.2, 0.5}. The default optimiser is Adam [3] with learning rate chosen from {1e-2, 1e-3, 1e-4} and the weight decay values from {1e-3, 1e-4, 1e-5}. In contrastive training, each query is associated with one positive sample and one easy negative sample, while the number of hard negative samples is selected from {1, 5, 10}. In-batch samples of other queries are also treated as easy negative samples. For degree regularisation, the coefficient λ is chosen from {0,5e-4,1e-3,5e-3}. The number of TopK case neighbour node K in Equation 4 is selected from {3, 5, 10}. The threshold δ in Equation 5 is chosen from {0.85, 0.9, 0.95}. Due to the 4096 token limit of e5-mistral-7b-instruct model, any case exceeding this length is truncated to 4096 tokens.

5.2 Result

The final top 10 runs of COLIEE 2025 Task 1 is shown in Table 2. The three runs of Team UQLegalAI are all run by CaseLink model with different hyperparameters. The F1 score reaches 0.2962, with a gap of less than 0.04 compared to the first-place team.

In addition to the overall performance, our method exhibits a **stable** situation. The result is stable that the variance of Precision, Recall and F1 is small. The gap is less than ±0.003 for our method. While for other methods, such as JNLP's submission, the variance is around ±0.01.

Compared with the highest performance, our method has a lower Recall score. This can be due to the selection of more candidates in ranking.

6 Conclusion

This paper presents the approach of Team UQLegalAI for Task 1 of the COLIEE 2025 competition. To leverage the intrinsic connectivity relationships between legal cases, a method called CaseLink is leveraged. Within the CaseLink framework, a Global Case Graph construction module is introduced to build a case graph comprising case-case edges, case-charge edges, and charge-charge edges for each case. Node features within the GCG are encoded by a high-quality text embedding large language model. A graph neural network module is then employed to generate informative case representations. The CaseLink model is trained with an InfoNCE contrastive objective combined with a novel degree regularization term. The final ranking results for Task 1 demonstrate the effectiveness and strong performance of the proposed CaseLink approach. In future work, developing more powerful models is still needed for enhancing the accuracy of legal case retrieval.

References

[1] William L. Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive Representation Learning on Large Graphs. In *NeurIPS*.
[2] Karen Spärck Jones. 1972. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation* 28, 1 (1972), 11–21.
[3] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *ICLR*.
[4] Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *ICLR*.
[5] 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*, Vol. 14741. 212–227.
[6] 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. *CoRR* abs/2305.06812 (2023).
[7] Yixiao Ma, Yueyue Wu, Qingyao Ai, Yiqun Liu, Yunqiu Shao, Min Zhang, and Shaoping Ma. 2023. Incorporating Structural Information into Legal Case Retrieval. *ACM Trans. Inf. Syst.* (2023).
[8] Niklas Muennighoff, Nouamane Tazi, Loïc Magne, and Nils Reimers. 2022. MTEB: Massive Text Embedding Benchmark. *CoRR* abs/2210.07316 (2022).
[9] Jay M. Ponte and W. Bruce Croft. 2017. A Language Modeling Approach to Information Retrieval. In *SIGIR*.
[10] 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. *Rev. Socionetwork Strateg.* 16, 1 (2022), 111–133.
[11] Juliano Rabelo, Mi-Young Kim, and Randy Goebel. 2022. Semantic-Based Classification of Relevant Case Law. In *JURISIN*.
[12] Stephen E. Robertson and Steve Walker. 1994. Some Simple Effective Approximations to the 2-Poisson Model for Probabilistic Weighted Retrieval. In *SIGIR*.
[13] Yunqiu Shao, Jiaxin Mao, Yiqun Liu, Weizhi Ma, Ken Satoh, Min Zhang, and Shaoping Ma. 2020. BERT-PLI: Modeling Paragraph-Level Interactions for Legal Case Retrieval. In *IJCAI*.
[14] Nitish Srivastava, Geoffrey E. Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* (2014).
[15] Yanran Tang, Ruihong Qiu, and Xue Li. 2023. Prompt-based Effective Input Reformulation for Legal Case Retrieval. *CoRR* abs/2309.02962 (2023).
[16] Yanran Tang, Ruihong Qiu, Yilun Liu, Xue Li, and Zi Huang. 2024. CaseGNN++: Graph Contrastive Learning for Legal Case Retrieval with Graph Augmentation. *CoRR* abs/2405.11791 (2024).
[17] 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 *ECIR*.
[18] Yanran Tang, Ruihong Qiu, Hongzhi Yin, Xue Li, and Zi Huang. 2024. CaseLink: Inductive Graph Learning for Legal Case Retrieval. In *SIGIR*.

[19] Aäron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation Learning with Contrastive Predictive Coding. *CoRR* abs/1807.03748 (2018).
[20] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph Attention Networks. In *ICLR*.
[21] Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. 2024. Improving Text Embeddings with Large Language Models. In *ACL*.
[22] Kun Zhang, Chong Chen, Yuanzhuo Wang, Qi Tian, and Long Bai. 2023. CFGL-LCR: A Counterfactual Graph Learning Framework for Legal Case Retrieval. In *SIGKDD*.

IRNLPUI at COLIEE 2025: Utilization of LLMs for Statute Law Retrieval and Legal Entailment Task

Bryan Tjandra, Made Swastika Nata Negara*
Alfan Farizki Wicaksono
Information Retrieval & NLP Lab. (IRNLP), Center for Legal Informatics (LEXIN)
Faculty of Computer Science, Universitas Indonesia (UI)
Depok, Indonesia
bryan.tjandra,made.swastika,alfan@ui.ac.id

Abstract

This paper describes our participation in the Statute Law Retrieval (Task 3) and Legal Textual Entailment (Task 4) tasks of the Competition on Legal Information Extraction/Entailment (COLIEE). For Task 3, we explored three distinct information retrieval (IR) approaches: a BM25-based system using score thresholding (**UIthr**), a re-ranking strategy that linearly combines BM25 and LLM-generated scores (**UIwa**), and a meta-classifier designed to learn relevance based on features including scores from BM25 and LLM prompts (**UImeta**). For Task 4, we explored three methods, each building upon approaches from previous COLIEE participants, notably JNLP. The first method adapted JNLP's approach by employing prompt selection and voting classification, but utilized a lightweight LLM and QLoRA fine-tuning (**UIRunFTune**). The second method is a prompt-voting predictor that reasons directly in Japanese, based on the premise that using the statute's original language could preserve linguistic particles and culturally specific nuances often lost in English translation (**UIRunLang**). The third method applied a voting strategy across the results generated by various advanced prompting techniques, including Chain-of-Thought (**UIRunCoT**). Finally, we discuss the system's characteristics based on the evaluation results of our COLIEE 2023 submissions.

CCS Concepts

• **Applied computing** → **Law**; • **Information systems** → **Retrieval models and ranking**.

Keywords

COLIEE, LLM, Legal Textual Entailment, Statute Law Retrieval

ACM Reference Format:
Bryan Tjandra, Made Swastika Nata Negara and Alfan Farizki Wicaksono. 2025. IRNLPUI at COLIEE 2025: Utilization of LLMs for Statute Law Retrieval and Legal Entailment Task. In *Proceedings of COLIEE 2025 workshop, June 20, 2025, Chicago, USA.* ACM, New York, NY, USA, 10 pages.

*First and second authors contributed equally to this research.

COLIEE 2025, Chicago, USA

1 Introduction

The increasing volume of legal documents generated by legislators and regulators has rendered traditional manual legal information processing methods unsustainable. Consequently, there is a growing need for automated and computing-based systems to assist legal professionals in managing and navigating this information overload. Such systems are crucial for efficient legal research, case preparation, and regulatory compliance. For instance, legal information retrieval and automatic textual entailment can accelerate the process of identifying relevant precedents, analyzing complex legal texts, and ensuring adherence to evolving regulations. This automation not only enhances productivity but also minimizes the potential for human error in critical legal decision-making.

The Competition on Legal Information Extraction/Entailment (COLIEE) serves as a forum for discussing issues related to legal information retrieval (IR) and textual entailment [5]. COLIEE features two categories of tasks: those using case law (Tasks 1 and 2) and those utilizing Japanese statute law based on Japanese bar exam questions (Tasks 3 and 4). This year, **IRNLPUI** participated in the statute law tasks, specifically Task 3: Statute Law Retrieval, and Task 4: Legal Textual Entailment. The statute law tasks used Japanese bar exam questions related to the Japanese Civil Code. We employed a portion of the Civil Code with an official English translation (768 articles). Training data (1,532 question-article pairs) was derived from previous COLIEE datasets. Test data consisted of 73 new questions from the 2025 bar exam.

Task 3 requires retrieving an appropriate subset $\{p_1, p_2, \ldots, p_n\}$ of Japanese Civil Code Articles from the collection S to determine the entailment of a legal bar exam question statement Q. Task 4 focuses on determining whether a given premise $P = \{p_1, p_2, \ldots, p_n\}$ (a set of retrieved legal articles) entails or contradicts a hypothesis Q (a legal bar exam question). This task requires evaluating the logical relationship between P and Q, which can be classified into two categories: entailment ($P \Rightarrow Q$) or contradiction ($P \Rightarrow \neg Q$).

This paper details our methods for Tasks 3 and 4 and discusses the system's characteristics based on the evaluation results of our submitted runs. For Task 3, we employed a combination of scoring regimes. This involved using the well-established BM25 algorithm, alongside an LLM-based scoring method, allowing us to capture deeper semantic relationships and contextual relevance beyond simple keyword matching. For Task 4, we utilized three distinct classification strategies: fine-tuning large language models, employing higher-level reasoning prompting techniques, and prompt-voting predictor that reasons in Japanese.

2 Related Works

2.1 Statute Law Retrieval

In COLIEE 2023, the CAPTAIN system [11] achieved first place by implementing ranking models based on large language models (LLMs). It utilized Tohoku BERT for Japanese and monoT5 for English, with the optimal results obtained by combining both outputs. The HUKB system [9] made use of an ensemble technique, merging keyword-based information retrieval (IR) with various configurations and LLM-based ranking using Tohoku BERT. Similarly, JNLP [1] used a hybrid approach, applying BM25 for Japanese retrieval and monoT5 for English ranking. The NOWJ system [18] followed a two-stage retrieval process, in which BM25 initially retrieved candidates, and a multilingual LLM-based ranking model (bert-base-multilingual-uncased) refined the results for both Japanese and English. Lastly, UA adopted a more traditional IR strategy, relying on BM25 and TF-IDF for document retrieval [17].

In COLIEE 2024, the winning team was JNLP [12], which used BERT-base-Japanese with checkpoint ensembling. Three systems employed large language models (LLMs), specifically Mistral, RankLLaMA, and Qwen, for ranking, scoring, and refining retrieved results. NOWJ [14] trained BERT for Sequence Classification with multitask learning (Tasks 3 and 4) and combined the results with BM25 scores. AMHR [15] used BM25 to retrieve the top-50 results, re-ranked them with MonoT5 fine-tuned for COLIEE, and applied LLMs for final selection.

Based on previous years' approaches, BM25 has demonstrated sufficient capability in retrieving relevant articles, particularly when the legal questions are simple and contain explicit keyword matches. However, LLMs have shown strength in understanding the semantic relationships between articles and questions, allowing them to capture relevance beyond lexical similarity. Given these findings, combining BM25 with LLM-based ranking may provide an effective balance, leveraging BM25's efficient keyword-based retrieval while incorporating LLMs' deeper semantic understanding. This hybrid strategy could lead to improved retrieval accuracy while maintaining a reasonable computational cost.

2.2 Legal Textual Entailment

JNLP@COLIEE-2023 [1] achieved first-rank performance by leveraging data augmentation and large language models (LLMs) for legal case entailment. Their approach involved fine-tuning LLMs with domain-specific datasets and applying ensemble strategies to enhance robustness. Similarly, AMHR Lab 2023 [15] demonstrated the potential of integrating generative large language models (LLMs) and ensemble strategies for legal textual entailment. Their approach utilized GPT-4 and Flan-T5 models, achieving state-of-the-art results on validation splits. This highlights the effectiveness of prompting strategies with generative LLMs. Furthermore, their ensemble methods showcased the importance of model combination strategies. These findings align with insights from "Performance of Individual Models vs. Agreement-Based Ensembles for Case Entailment" [3], which empirically validated the superiority of agreement-based ensembles over single-model predictions.

To ground these insights in concrete evidence, we reproduced two of the best-known 2023 pipelines on the COLIEE-2025 training set: (1) the encoder-centric KIS workflow based on LUKE and extensive legal pre-processing, and (2) JNLP's lightweight prompt-selection strategy in which we fine-tuned a 7-billion-parameter Qwen2 model. Based on our experiment the fine-tuned light-weight LLM achieved an F1 score of 0.676, outperforming the best encoder variant that we experiment (ModernBERT, F1 = 0.625) even after domain-specific pre-processing and data augmentation. This empirical gap motivated us to adopt the more robust Qwen2-72B as the backbone of our subsequent runs (UIRunLang, and UIRunCoT).

CAPTAIN [11] demonstrated a significant contribution by integrating keyword-based matching with neural embeddings for legal textual entailment at COLIEE 2023. They fine-tuned transformer models, specifically Tohoku BERT and monoT5, highlighting the strong performance achievable with encoder-based models through a balanced approach to lexical precision and semantic understanding. Similarly, KIS [16] employed LUKE, an encoder-based model, complemented by preprocessing steps that expanded the dataset and filtered relevant clauses to focus on critical legal content at COLIEE 2024. Their methodology also incorporated rule-based ensembling and the aggregation of predictions from multiple LUKE runs, demonstrating the efficacy of combining rule-based reasoning with neural models. This approach secured the second-rank position, underscoring the effectiveness of ensemble techniques and encoder models. Furthermore, NOWJ [14] explored multi-task learning and ensemble strategies, emphasizing the importance of prediction aggregation for robust performance. Their use of BERT-base-multilingual-uncased for both English and Japanese texts showcased the model's noteworthy cross-lingual capabilities.

Higher-level reasoning techniques have also been a focal point in recent research. The AHMR Lab team [15] and CAPTAIN (2024) [13] also utilized frameworks that leverage chain-of-thought (CoT) reasoning to address complex legal cases. These methods informed our adoption of advanced reasoning paradigms such as Graph-of-Thought (GoT) and Tree-of-Thought (ToT), which extends the capabilities of traditional CoT by modeling further reasoning between legal entities and clauses [8].

3 Methods

3.1 Statute Law Retrieval Task

The objective of Task 3 in the COLIEE 2025 competition is to retrieve a subset of Japanese Civil Code Articles S that are relevant to a given legal bar exam question, denoted as Q. Participants are required to identify a set of articles, formally represented as $P = \{p_1, p_2, \ldots, p_n\}$, that provides legal justification for answering Q. This task serves as a precursor to Task 4, which involves reasoning over the selected articles to determine the legality of Q. Our system follows a two-stage retrieval approach: (1) a hierarchical indexing strategy to structure and index the legal articles effectively, and (2) a multi-method retrieval process that incorporates lexical matching, large language models (LLMs), and a machine learning-based classification approach for final selection. To thoroughly investigate and optimize retrieval performance, we have designed and implemented three distinct retrieval pipelines, named **UIthr**, **UIwa**, and **UImeta**. This allows us to explore various trade-offs and to tailor the retrieval process to different types of legal queries, all with the goal of improving overall retrieval effectiveness of the returned legal articles.

Legal Text Structuring and Indexing. The Japanese Civil Code follows a hierarchical structure comprising parts, chapters, sections, and articles. To enhance retrieval effectiveness, we preprocessed the statute law documents by extracting and appending hierarchical metadata prior to indexing. Subsequently, each article is structured as shown in Table 1. The "Full Text" field will be used for document indexing. This enriched representation ensures that each legal article is indexed with its complete legal context. For indexing and retrieving documents from the structured corpus, we utilized BM25, a probabilistic retrieval ranking function based on term frequency-inverse document frequency (TF-IDF).

Type	Value
Part Number	I
Part Text	General Provisions
Chapter Number	II
Chapter Text	Persons
Section Number	3
Section Text	Capacity to Act
Article Title	Permission for Minors to Conduct Business
Article Number	6
Article Text	(1) A minor who is permitted to conduct one or multiple types of business has the same capacity to act as an adult as far as that business is concerned. (2) In a case as referred to in the preceding paragraph, if there are grounds that make the minor unable to sustain that business, the legal representative may revoke or limit the permission therefor in accordance with the provisions of Part IV (Relatives).
Full Text	General Provisions. Persons. Capacity to Act. Permission for Minors to Conduct Business. (1) A minor who is permitted to conduct one or multiple types of business has the same capacity to act as an adult as far as that business is concerned. (2) In a case as referred to in the preceding paragraph, if there are grounds that make the minor unable to sustain that business, the legal representative may revoke or limit the permission therefor in accordance with the provisions of Part IV (Relatives).

Table 1: Article sample with extracted hierarchical metadata

BM25-Based Initial Retrieval. BM25 is a widely adopted information retrieval model known for its ability to rank documents based on the overlap of terms between the query and the document, while also accounting for term saturation and document length normalization. We used BM25 to generate an initial candidate set comprising the top-20 articles per query. Subsequently, we explored various strategies to refine the selection process, including re-ranking, from these retrieved documents, resulting in three distinct retrieval pipelines: **UIthr**, **UIwa**, and **UImeta**.

UIthr – BM25 Score Thresholding. In this approach, we apply a fixed threshold based on the raw BM25 score. Specifically, we experimented with varying BM25 score thresholds (i.e., filtering documents whose scores exceed 30, 40, 50, or 60) to determine which threshold yields the best trade-off between precision and recall. We empirically found that a threshold of 50 achieved the optimal F-score. Thus, we selected articles with BM25 scores greater or equal than 50 as the final set of relevant candidates. Algorithm 1 provides the detailed implementation.

Algorithm 1: BM25-Based Document Retrieval with Score Thresholding.

input : Question Set Q, Article Collection D, BM25 Scoring Function $BM25(q, p)$, Score Threshold T, Top-k
output : Retrieved Set P

foreach *question* $q \in Q$ **do**
  Initialize candidate set $C \leftarrow \emptyset$;
  foreach *article* $p \in D$ **do**
    Compute $score \leftarrow BM25(q, p)$;
    Add $(p, score)$ to C;
  Sort C by score in descending order;
  Select top k documents from C as C_k;
  Initialize retrieval set $P_q \leftarrow \emptyset$;
  foreach $(p, score) \in C_k$ **do**
    if $score \geq T$ **then**
      Add p to P_q;
  if $P_q = \emptyset$ **then**
    Select document p_{max} with the highest score from C_k;
    Add p_{max} to P;
  Add P_q to P;
return P;

UIwa – BM25 and LLM-Based Scores. To enhance the retrieval pipeline with deeper semantic understanding, we integrated large language models (LLMs) to refine the initial search results. While traditional lexical-based retrieval methods, such as BM25, are effective at identifying relevant legal articles by matching keyword occurrences, they often struggle to capture the underlying semantic meaning of queries, particularly in complex legal contexts. To address this limitation, we employed an additional LLM-based ranking step to re-evaluate the relationship between a query and the retrieved documents, moving beyond simple keyword matching. We utilized `OpenHermes 2.5 Mistral 7B bnb 4bit`[1], selected for its compatibility with COLIEE 2025's restrictions on using models last updated before the 2024 Japanese bar exam. We utilized **Unsloth** [4], an optimization framework specifically designed for the efficient fine-tuning and inference of large models. This framework enables faster processing by using quantization techniques,

[1] https://huggingface.co/unsloth/OpenHermes-2.5-Mistral-7B-bnb-4bit

which reduce model size and computational overhead without significantly sacrificing accuracy. Subsequently, we refined the BM25 results by scoring each question-article pair using the LLM. The LLM assigned a relevance score based on the semantic relationship between the question and the legal article. We then computed a straightforward weighted average of the BM25 score and the LLM-assigned relevance score:

$$S_{\text{final}}(q,p) = \alpha \cdot S_{\text{BM25}}(q,p) + (1-\alpha) \cdot S_{\text{LLM}}(q,p)\,,$$

where $S_{\text{BM25}}(q,p)$ is the BM25 score of document p for query q; $S_{\text{LLM}}(q,p)$ is the LLM-based relevance score; and α is a tunable parameter controlling the balance between lexical and semantic relevance, with $0 \leq \alpha \leq 1$. Articles exceeding a predetermined threshold, T, were selected as the final results. The final set of retrieved documents, denoted by P', was determined as follows:

$$P' = \{p \in P \mid S_{\text{final}}(q,p) \geq T\}\,.$$

If no documents satisfy the threshold T, we select the highest-scoring document, $P' = \{p_{\max}\}$, where

$$p_{\max} = \arg\max_{p \in P} S_{\text{final}}(q,p)\,.$$

UImeta – Data-driven Meta-Classifier. Instead of relying on a simple weighted average, this approach makes use of a meta-classifier to determine the final set of relevant legal articles. We used **logistic regression**, a statistical model commonly used for binary classification, to predict the relevance of a question-article pair. The model was trained using features derived from:

- BM25 relevance scores, denoted by S_{BM25},
- LLM-assigned relevance scores, denoted by S_{LLM},
- Cosine similarity between question and article using embedding `legal-bert-base-uncased`[2], denoted by $S_{\cos}$,
- Article length, denoted by $len(d)$,
- Question length, denoted by $len(p)$.

After training, the logistic regression classifier predicts whether each article should be included in the final selection. The final set of retrieved documents is:

$$P' = \{p \in P \mid L(y=1,p) \geq T\}\,,$$

where $L(y=1,p) = \sigma(\mathbf{w}^T.f(p) + b)$; $L(y=1,p) = 1 - L(y=0,p)$; $\sigma(.)$ is a logistic function; $\mathbf{w} \in R^5$ and $b \in R$ are trainable parameters; and $f(p)$ represents a feature vector extracted from p: $[S_{\text{BM25}}, S_{\text{LLM}}, S_{\cos}, len(d), len(p)]$. If no document meets the threshold T, we select the highest confidence prediction, $P' = \{p_{\max}\}$, where

$$p_{\max} = \arg\max_{p \in P} L(y=1,p)\,.$$

3.2 Legal Textual Entailment Task

The objective of Task 4 in the COLIEE 2025 competition is to determine the relationship of legal textual entailment between a given premise P and a hypothesis H. Participants were required to classify the relationship between P and H into one of three categories: *entailment* or *contradiction*. In this task, P represents a legal text, typically an article or a set of articles from the Japanese Civil Code, while H represents a legal statement derived from a bar exam question. The goal is to evaluate whether the content in P logically supports H (*entailment*) or contradicts H (*contradiction*).

Our approach is centered on producing computationally light and economical solutions. We begin the study by replicating and streamlining the strategies of the JNLP and KIS teams, who were the top two performers in COLIEE Task 4 in 2023, using the English dataset of given data by COLIEE 2025 committee. The JNLP team's methodology involves collecting and comparing prompts to determine which ones perform well, then fine tuning a language model and performing evaluations [2]. On the other hand, the KIS team utilizes an encoder-based model with advanced preprocessing techniques to expand and filter relevant legal data, then employs an ensemble of fine-tuned LUKE-based models with rule-based methods [16].

UIRunFTune – Prompt Selection & Fine-Tuning. The JNLP team's original methodology emphasized prompt selection and language model tuning [2]. Our adaptation focuses on lightweight reproduction using open-source language model. The core elements include prompt selection, fine-tuning prompted LLMs, and voting classification. We utilized *PromptSource*[3] to extract 10 candidate prompts designed for natural language inference (NLI) tasks. PromptSource is a framework for efficiently creating and managing prompts for various natural language tasks, allowing for systematic prompt generation, customization, and evaluation [6]. These prompts were selected based on their compatibility with the format of Natural Language Inference and legal premise-hypothesis pairs in the COLIEE dataset. Each prompt was designed to fuse the premise (*T1*) and hypothesis (*T2*) into a question-answer format. These prompts are:

(1) "Does the claim `{hypothesis}` follow from the fact that `{premise}`? Please answer either yes or no."
(2) "We say that one sentence entails another sentence when the first sentence implies the second sentence. Consider the following two sentences: `{premise}` `{hypothesis}` Is the relationship from the first to the second sentence entailment or not entailment?"
(3) "Does `{premise}` imply that `{hypothesis}`? Please answer either yes or no."
(4) "`{premise}` Does this imply `{hypothesis}` Please answer A) yes or B) no."
(5) "`{premise}` Does this mean that `{hypothesis}` is true? A) yes or B) no."
(6) "Suppose `{premise}`. Can we infer that `{hypothesis}`? Yes or no?"
(7) "Given that `{premise}`, Does it follow that `{hypothesis}`? Yes or no?"
(8) "`{premise}` Question: Does this imply that `{hypothesis}`? Yes or no?"
(9) "Given that `{premise}`, Therefore, can we conclude that `{hypothesis}` is necessarily true? Yes or no?"
(10) "Take the following as truth: `{premise}`. Then the following statement: `{hypothesis}` is true or false?"

[2] https://huggingface.co/nlpaueb/legal-bert-base-uncased

[3] https://github.com/bigscience-workshop/promptsource

Due to resource limitations, a sample of 200 random data points was used for evaluation. Each prompt's performance was assessed using the Flan-T5 Base model[4] in a zero-shot setting. The selection of the model considers the resource limitation for the GPU and running time. If the model produced ambiguous or non-conforming answers (e.g., outputs lacking a clear "yes" or "no"), an additional inference step was introduced using an encoder-based model such as Facebook BART Large MNLI[5], which explicitly classified entailment relationships.

The top three performing prompts were selected for further fine-tuning task. Using these three prompts, we performed fine tuning on 300 samples via Qwen2-7B[6] in a zero-shot setting. We chose this model because of its state of the art model, and it demonstrates strong generalization capabilities and computational efficiency, making it suitable for handling diverse legal reasoning tasks [19]. To optimize the fine-tuning process, we utilized the **Unsloth** [4] with a quantized 4-bit precision, trained for 5 epochs using a **LoRA (Low-Rank Adaptation)** adapter [7]. After fine-tuning, predictions were made. If no explicit "yes" or "no" answer was present, the output was forwarded to the encoder-based model, such as Facebook BART Large for further clarification.

UIRunLang – Japanese Voting Prediction. Based on the initial performance comparison between the fine-tuned lightweight LLM model and the fine-tuned encoder-based model, the results indicate that the fine-tuned LLM model performs better. Consequently, we proceed to utilize a more robust LLM, such as QWEN-2 with 72 billion parameters, for error analysis on the hard cases.

Complex cases where the model consistently failed (less than 25% correct predictions among three prompts with three trials each) were presented to legal experts for further analysis. The objective was to gain insights into the reasoning patterns required to correctly predict entailment or contradiction in challenging scenarios by real human. Legal experts were asked to provide feedback on why certain hypotheses were incorrectly predicted and what are the reasoning steps that could aid in predicting these statements. Based on an interview with one of the lawyers, it was stated that laws in each country are distinct and can be influenced by that country's culture or behavior. Therefore, they cannot be generalized, as language can define the origin of the law and influence its interpretation and application.

Given that the case pertains to Japan, we experimented with the Japanese language to assess its impact on model accuracy. The results indicate that in complex cases where the model consistently failed using English language prompts, Japanese language prompts performed better in these scenarios. We hypothesize that this gain stems from the fact that the original language encodes culture–specific legal nuance that is partially lost in translation. Japanese statutory phrases often employ particles and domain-specific kanji compounds whose exact force cannot be rendered one-to-one in English. Operating directly on the Japanese version therefore allows the model to reason with the same conceptual primitives that human jurists use, leading to more faithful logical deductions in borderline cases.

[4]https://huggingface.co/google/flan-t5-base
[5]https://huggingface.co/facebook/bart-large-mnli
[6]https://huggingface.co/Qwen/Qwen2-7B-Instruct

To stabilise the predictions we also introduce a lightweight *voting* layer. Concretely, we reuse the three top-performing prompt templates (Prompt 4, 5, 10) identified during the UIRunFTune study. At inference time the model is queried with all three prompts and the final label is decided by majority vote.

UIRunCoT – Prompt Engineering. To address the remaining complex cases without translation issues, we employed advanced reasoning-based prompt engineering that leverage structured logical frameworks and hierarchical reasoning processes. These methods include utilizing Chain-of-Thought (CoT), Tree-of-Thought (ToT), and Graph-of-Thought (GoT) reasoning.

Chain-of-Thought (CoT) reasoning involves breaking down the reasoning process into sequential steps, enabling the model to explicitly articulate its thought process. We prompt the model to (1) extract the controlling clause, (2) align clause elements with the hypothesis, (3) identify any gaps or conflicts, and (4) assign a 1–5 confidence entailment score. This method is particularly effective for legal entailment tasks because it mimics the logical progression that human experts use when analyzing complex legal principles [10]. For instance, legal reasoning often requires identifying the core principle in a statute, comparing it with the hypothesis, and justifying whether the relationship is one of entailment, contradiction, or neutrality. Furthermore, as highlighted in [10], CoT significantly improves performance on tasks requiring multi-step reasoning, which are common in legal domains.

Tree-of-Thought (ToT) reasoning extends CoT by exploring multiple reasoning paths based on possible interpretations of the legal principle [8]. Legal texts often contain ambiguities or multiple plausible interpretations, especially when dealing with nested clauses or conditional relationships. ToT addresses this challenge by branching the reasoning process into three parallel paths, Path A, which checks direct support or contradiction; Path B, which resolves ambiguity via alternative readings; and Path C, which brings in external precedents or contextual factors. As noted in [8], ToT is particularly effective in scenarios where legal principles are open to subjective interpretation, as it enables the model to weigh competing arguments and select the most convincing path.

Graph-of-Thought (GoT) uses a dynamic reasoning graph to represent logical connections between reasoning steps [8]. This graphical representation is especially well-suited for legal reasoning tasks, which often involve intricate dependencies and cyclical relationships. For example, a legal principle may depend on prior interpretations, statutory rules, or judicial precedents, all of which can influence its application in a given case. This structure enables the model to dynamically explore multiple reasoning paths while maintaining coherence across the entire graph. As highlighted in [8], GoT excels in handling complex, multi-layered reasoning tasks, making it an ideal choice for legal entailment problems that require synthesizing information from diverse sources. Additionally, the graph-based approach facilitates the identification of strong and weak arguments.

The final prediction strategy, **UIRunCoT**, achieved the highest overall performance on our evaluation subset by integrating advanced reasoning methods – specifically, CoT, ToT, and GoT – into a voting mechanism. We believe this voting mechanism is able to

Prompt for CoT:

You are a legal expert analyzing legal entailment. Follow these steps:
1. Identify the core legal principle from the given article.
2. Compare it with the statement to determine if it is explicitly supported, implicitly supported, or contradicted.
3. Justify your reasoning with reference to legal logic.
4. Provide a final entailment decision on a **confidence scale (1-5)**.

Legal Article: {premise}
Statement: {hypothesis}

- [Answer]

Step 2: Compare the principle with the statement:

- [Does the statement match the article? If yes, how? If no, why?]

Step 3: Provide a final entailment decision using the confidence scale below:

1 - Highly confident it is contradicted
2 - Shows slight contradiction
3 - Neutral (neither entailed nor contradicted)
4 - Shows slight entailment
5 - Highly confident it is entailed

- [Final answer: Select a number from 1 to 5 based on the level of entailment]

Figure 1: Prompt template for Chain-of-Thought reasoning.

harness the individual strengths of each method to handle diverse challenges in legal reasoning.

Figure 1 shows the exact Chain-of-Thought prompt template we use. This structured multi-step format ensures the model systematically extracts, compares, and justifies each legal element before rendering a confidence-scored entailment decision, thereby markedly improving both interpretability and accuracy. For ToT, we extend this template by instructing the model to generate three parallel reasoning branches (direct support/contradiction, alternate interpretations, precedent-based) and then reconcile them. For GoT, we further augment the prompt to ask the model to output explicit reasoning "nodes" and "edges," forming a dependency graph whose strongest subgraph determines the final entailment decision.

4 Experiments & Discussions

4.1 Statute Law Retrieval Task

Dataset. Similarly to the format of the previous year, the data set for this task consists of 996 questions, a legal corpus (Civil Code) with 768 articles and 1532 pairs of questions and relevant articles (positive samples). For the development process, we choosed questions that have an ID starting R05 (109 questions) as a validation set and conduct model/settings evaluations on this subset.

Official Test Results. Table 2 shows the full test results of Task 3 provided by the COLIEE 2025 organization, including our models (**UIthr**, **UIwa**, and **UImeta**)[7]. Our models achieved moderate performance, with precision scores comparable to some of the higher-ranked systems. However, they exhibit a notable deficiency in recall, which impacts their overall retrieval effectiveness. While our models maintain a reasonable balance between precision and recall, they struggle to retrieve all relevant documents effectively. This results in a lower MAP score and diminishes performance in ranking metrics such as R@5, R@10, and R@30.

Run	F2	P	R	MAP	R@5	R@10	R@30
JNLP_RUN1	**.8365**	.8037	.8744				
CAPTAIN.H2	.8301	**.8333**	.8516	.6721	.7174	.7935	.9130
CAPTAIN.H3	.8204	.8002	.8584	.6721	.7174	.7935	.9130
CAPTAIN.H1	.8103	.8196	.8311	.6721	.7174	.7935	.9130
JNLP_RUN2	.7863	.7272	.8402				
JNLP_RUN3	.7861	.7420	.8265				
INFA	.6917	.7671	.6826	.6463	.7065	.7717	.8913
mpnetAIIRLab	.6674	.3562	**.8858**	**.8012**	**.8696**	**.9130**	**.9674**
OVGU3	.6041	.6347	.6142	.7134	.8152	.8587	.9674
mistralRerank	.5962	.3196	.7900	.6916	.8261	.9022	.9022
OVGU2	.5959	.6096	.6027	.7466	.7826	.8696	.9565
NVAIIRLab	.5836	.3014	.7854	.7468	.7609	.8261	.9348
UIwa	.5816	.5856	.5890	.6656	.6739	.7500	.7609
UImeta	.5793	.5788	.5890	.6720	.7283	.7609	.7609
UIthr	.5723	.6027	.5685	.6656	.6739	.7500	.7609
OVGU1	.4672	.4635	.4795	.7134	.8152	.8587	.9674
UA-mpnet	.2540	.0986	.4361	.3435	.3913	.4674	.6196
UA-gte	.2517	.0986	.4292	.3244	.3913	.4783	.6304
UA-bm25	.2113	.0795	.3699	.3168	.3152	.4457	.6196
NOWJ.H1	.0137	.0137	.0137	.0268	.0217	.0652	.1304
NOWJ.H2	.0137	.0137	.0137	.0268	.0217	.0652	.1304
NOWJ.H3	.0137	.0137	.0137	.0268	.0217	.0652	.1304

Table 2: Performance of our models on the official test set for Task 3 at COLIEE 2025.

Error analysis. To further examine the system's recall performance, we analyzed how it performs based on the number of relevant articles per query. Table 3 presents the results for our best-performing system, **UIwa**. The results indicate that the system achieves a reasonable recall for queries with only one relevant article. However, when the number of relevant articles increases, performance drops significantly. For queries with two relevant articles, recall falls sharply, leading to a decline in the F2 score. Moreover, for the one query with three relevant articles, the system failed to retrieve any relevant documents within the top three ranks. This pattern suggests that the model was more optimized for single-relevant-article queries and struggles with multi-relevant-item retrieval.

[7]The source code for our methods and experiments for Task 3 is available at the following link: https://github.com/swastikanata/coliee2025-task3-irnlpui

# Rel.	Count	Prec.	Recall	F2	# Rel. Retr.
1	55	69.54	72.72	71.43	72.72
2	17	26.47	17.64	18.62	35.29
3	1	0.00	0.00	0.00	0.00

Table 3: UIwa performance based on number of relevant articles of each query.

In addition, we analyzed a sample of the questions where our system struggled to retrieve relevant articles. In this sample, there were 14 questions with an R@5 of 0, 9 questions with an R@10 of 0, and 8 questions with an R@30 of 0. Among these 8 questions where R@30 was 0, some failures involve articles that apply *mutatis mutandis* references to other articles with slight modifications in meaning. This presents a challenge for retrieval models because there is often no direct keyword match between the question and the referenced articles. Traditional retrieval methods do not capture this implicit cross-referencing mechanism, making it difficult to rank such articles highly in search results. Addressing this issue may require incorporating explicit legal cross-references into the retrieval model.

Another error case highlights difficulties in understanding the semantic meaning of anonymous symbols, such as "A", "B", "G", and "Y". For example: "`Suppose A has a claim for the sale price of 10 million yen against B based on a purchase and sale contract with B. B donates movable property Y owned by B to G, who then donates Y to H, and H then donates Y to I . . .`". Since keyword-based retrieval models like BM25 rely on lexical overlap, they struggle to surface such semantically relevant articles, preventing them from being evaluated by LLMs in later processing stages.

4.2 Legal Textual Entailment

Prompt Selection for UIRunFTune. We evaluated 10 candidate prompts extracted from *PromptSource* that is compatible with the COLIEE dataset and the natural language inference (NLI) task. Figure 2 visualizes the performance trends across all 10 prompts, we plotted the F1 scores for each trial along with the average F1 score. The plot highlights the variability and consistency of the results. After conducting trials with a sample of 200 random data points, we identified Prompt 4, 5, 10 as the top-3 performing prompt. The content of prompts was as follows:

(1) Prompt 4:
```
{premise}
Does this imply
{hypothesis}
Please answer {A) yes or B) no.}
```

(2) Prompt 5:
```
{premise}
Does this mean that
{hypothesis}
is true? {A) yes or B) no.}
```

(3) Prompt 10:
```
Take the following as truth:
{premise}.
Then the following statement:
{hypothesis}
is {true} or {false}?
```

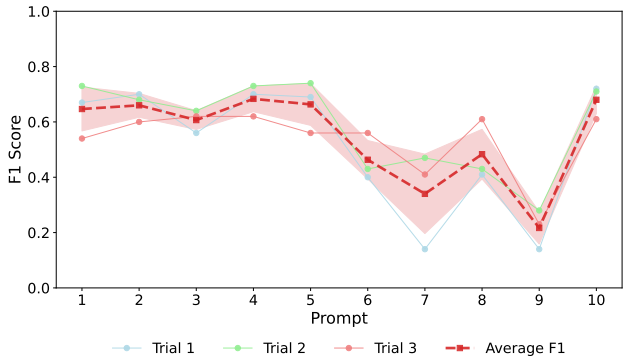


Figure 2: F1 Scores per Prompt Across Three Trials with Average and Standard Deviation.

The top three performing prompts (Prompt 4, Prompt 5, and Prompt 10) will be fine-tuned on 300 samples using Qwen2-7B in a zero-shot setting, then we evaluated their performance. Following fine-tuning, a voting mechanism was applied to combine the predictions of the three prompts.

English vs Japanese. Given that laws in each country are distinct and influenced by cultural or behavioral contexts, language plays a crucial role in defining the origin, interpretation, and application of legal texts. To investigate the impact of language on model performance, we conducted an evaluation and error analysis focusing on predictions made using both English and Japanese versions of the dataset. For this analysis, we used a sample of 250 instances from the training data, conducting nine trials (three trials each with Prompts 4, 5, and 10).

This analysis revealed 35 instances where the model can only correctly predict the result in two or fewer out of nine attempts, which are considered **hard cases**. Conversely, there are 175 instances where the model succeeded in seven or more out of nine attempts, which we classify as **easy cases**. We evaluated the performance of Qwen 2-72B on hard and easy cases for both languages. Tables 4 and 5 show the results, which indicate that Japanese predictions outperform English predictions in hard cases, while English predictions slightly outperform Japanese in easy cases.

Japanese Language. Legal systems are inherently bound to the linguistic and cultural contexts in which they develop. The Japanese Civil Code embodies conceptual frameworks shaped by Japan's unique legal tradition and societal values, with terminology that carries precise doctrinal meanings refined through decades of judicial practice. When legal texts are translated, critical nuances embedded in original lexical choices and grammatical structures often become diluted.

Model	True Pred.	False Pred.
Qwen Prompt 4 (English)	4	31
Qwen Prompt 5 (English)	4	31
Qwen Prompt 10 (English)	3	32
Qwen Prompt 4 (Japanese)	13	22
Qwen Pred 5 (Japanese)	13	22
Qwen Pred 10 (Japanese)	13	22

Table 4: Performance on Hard Cases.

Model	True Pred.	False Pred.
Qwen Prompt 4 (English)	168	7
Qwen Prompt 5 (English)	169	6
Qwen Prompt 10 (English)	171	4
Qwen Pred 4 (Japanese)	164	11
Qwen Pred 5 (Japanese)	167	12
Qwen Pred 10 (Japanese)	164	11

Table 5: Performance on Easy Cases.

Consider the determination of whether usufructuary rights apply exclusively to real property. The Japanese formulation in Figure 3 uses the term 不動産 (fudōsan), which in Japan's legal taxonomy strictly denotes land and permanent structures, a closed category excluding movable assets. This precision stems from Japan's historical land tenure system, where property rights developed around agricultural land management. The hypothesis' phrasing 成立する (seiritsu-suru, "be constituted") directly echoes Article 175's 創設 (sōsetsu, "create"), establishing a lexical chain that models can detect. In contrast, the English translation's "real estate" carries broader common-law connotations, potentially encompassing movable property in some jurisdictions. While the Japanese formulation leaves no interpretative room due to its culturally-specific definitions, the English version's ambiguity could support multiple readings. This demonstrates how original Japanese processing captures definitional rigor that translations necessarily attenuate.

UIRunCoT results. The performance of the higher-level reasoning methods is summarized in Table 6. Each method was evaluated on accuracy, precision, recall, and F1 score. Then, the voting is used to vote from the result of CoT, ToT, and GoT.

Method	Accuracy	Precision	Recall	F1 Score
CoT	0.792	0.8067	0.8329	0.8196
ToT	0.778	0.7984	0.8278	0.8128
GoT	0.764	0.7951	0.8242	0.8094
Voting	0.797	0.8099	0.8398	0.8246

Table 6: Performance metrics for LLMs that use higher-level reasoning methods

From the results, each reasoning method offers unique strengths tailored to specific types of legal reasoning tasks. Chain-of-Thought (CoT) performed well with an F1 score of 0.8196, showcasing the advantages of step-by-step reasoning in enhancing interpretability and transparency. Tree-of-Thought (ToT) outperformed other methods with an F1 score of 0.8128, as it excelled in exploring multiple reasoning paths and resolving ambiguities, making it particularly suitable for complex legal scenarios with multiple plausible interpretations. Graph-of-Thought (GoT), with an F1 score of 0.8094, demonstrated proficiency in capturing intricate dependencies but showed a slight decline in precision, indicating challenges in handling speculative or indirect reasoning. Finally, the voting mechanism (**UIRunCoT**), which combined CoT, ToT, and GoT, achieved the highest overall F1 score of 0.8246. This approach made use of

English premise

Article 175 No real right may be established other than those prescribed by laws, including this Code.

Article 265 A superficiary has the right to use another person's land in order to own structures, or trees or bamboo, on that land.

Article 270 A farming right holder has the right to pay rent and engage in cultivation or livestock farming on another person's land.

Article 280 A servitude holder has the right to use another person's land for the convenience of their own lands in accordance with purposes prescribed in the act establishing the servitude; provided, however, that this right must not violate the provisions (limited to those that relate to public policy) under Section 1 of Chapter 3 (Extent of Ownership).

Hypothesis

Usufructuary rights are only established for real estate.

Japanese premise

第百七十五条 物権は、この法律その他の法律に定めるもののほか、創設することができない。

第二百六十五条 地上権者は、他人の土地において工作物又は竹木を所有するため、その土地を使用する権利を有する。

第二百七十条 永小作人は、小作料を支払って他人の土地において耕作又は牧畜をする権利を有する。

第二百八十条 地役権者は、設定行為で定めた目的に従い、他人の土地を自己の土地の便益に供する権利を有する。ただし、第三章第一節（所有権の限界）の規定（公の秩序に関するものに限る。）に違反しないものでなければならない。

Hypothesis

用益物権は，不動産にのみ成立する。

Figure 3: Case where Japenese is correctly predicted.

the strengths of each method. The final prediction strategy, **UIRunCoT**, integrates these advanced reasoning methods to handle diverse challenges in legal reasoning. This pipeline ensures minimal computational costs while achieving state-of-the-art performance in complex legal entailment tasks.

Official Test Runs. We tested three prediction strategies that we proposed earlier on the COLIEE's official test dataset[8]. The performance of these three main strategies used in our approach was as follows:

- **UIRunCoT** achieved an accuracy of 80.09% on the case H30 to R02. This demonstrates the effectiveness of higher-level reasoning methods like Chain-of-Thought (CoT) in addressing complex legal entailment tasks;
- **UIRunLang** achieved an accuracy of 82.19% on the case R06 and 82.87% on the case H30 to R02, showcasing the value of demonstrating the effectiveness of using the Japanese language and voting pipeline;
- **UIRunFTune** achieved an accuracy of 60.27% on the test case and 64.51% on the case H30 to R02, indicating that while fine-tuning lightweight LLMs improved performance over baseline models, it was outperformed by advanced reasoning methods like CoT and language-specific strategies.

Table 7 summarizes the performance of all participating teams in Task 4 of the COLIEE 2025 competition. Each team's predictions were evaluated on 74 test cases, and the accuracy score was calculated as the percentage of correct predictions. Note that the test run result for **UIRunCoT** was not included in the official table because, at the time of the competition, we used a `Qwen` version released after July 9, 2024 (JST).

Team	Correct Predictions	Accuracy (%)
KIS3	66	90.41
KIS1	64	87.67
KIS2	62	84.93
CAPTAIN2	60	82.19
UIRunLang	60	**82.19**
JNLP002	59	80.82
JNLP003	59	80.82
CAPTAIN1	58	79.45
CAPTAIN3	58	79.45
UA2	57	78.08
UA3	57	78.08
JNLP001	56	76.71
KLAP.H2	56	76.71
UA1	55	75.34
NOWJ.run1	54	73.97
NOWJ.run2	54	73.97
NOWJ.run3	54	73.97
OVGU1	54	73.97
KLAP.H1	48	65.75
RUG_V1	48	65.75
OVGU3	46	63.01
RUG_V3	46	63.01
RUG_V2	45	61.64
AIIRLLaMA	44	60.27
UIRunFTune	44	**60.27**
OVGU2	44	60.27
AIIRMistral	41	56.16
BaseLine	37	50.68

Table 7: Performance metrics for Task 4 test of COLIEE 2025. Our team's runs (UIRunLang and UIRunFTune) are highlighted in bold. Note that UIRunCoT was excluded from the official table because, at the time of the competition, we employed a version of an LLM that was prohibited by COLIEE regulations.

5 Conclusions

In this paper, we presented our methodologies for Tasks 3 and 4 of the COLIEE 2025 competition. For Task 3, our system employed a novel two-stage retrieval framework. This involved a hierarchical indexing strategy for efficient organization of legal articles, followed by a multi-faceted retrieval process leveraging lexical matching for precise keyword identification, large language models (LLMs), and machine learning classification for refined selection. We investigated three distinct information retrieval approaches: a threshold-based BM25 system (**UIthr**), a linear re-ranking strategy combining BM25 and LLM scores (**UIwa**), and a meta-classifier learning relevance from BM25 and LLM features (**UImeta**).

For Task 4, our approach involved a systematic evaluation of various prompts to identify those with strong performance. The selected prompts then informed the fine-tuning of a language model, which was subsequently used to generate predictions (**UIRunFTune**). Since the law is created in Japanese, we utilize Japanese-based data since it can influence the model to relate to its country behavior and context (**UIRunLang**). Finally, our (**UIRunCoT**) strategy demonstrated the effectiveness of a voting ensemble combining the outputs of CoT, ToT, and GoT prompting techniques.

The development of our systems revealed several key challenges. One notable limitation was the difficulty in handling *mutatis mutandis* references, where articles apply modified meanings from other articles. Furthermore, our reliance on exact phrase matching proved insufficient for capturing paraphrased or reworded legal concepts, as exemplified by the mismatch between "`becomes effective only for the future`" in questions and "`retroactive to the time of the conclusion`" in relevant articles. To address this, future work should explore the integration of advanced semantic processing mechanisms. Next, inconsistencies such as mistranslations and contradictions between the Japanese and English versions of the dataset also presented obstacles. Finally, although fine-tuning lightweight LLMs improved the baseline models, they still revealed a performance gap compared to zero-shot methods on LLMs with a significantly larger number of parameters. Further investigation is necessary to determine the cause of this discrepancy, specifically

[8]The source code for our methods and experiments for Task 4 is available at the following link: https://github.com/bryan273/UI-COLIEE2025/tree/main

whether it arises from employing parameter-efficient tuning (rather than full parameter tuning) or simply from the inherent limitations of a lower parameter count.

Acknowledgments

We would like to express our gratitude to our legal experts – a law student, a law lecturer, and a lawyer – for their generous participation in interviews that aided our analysis of complex cases where the model exhibited persistent inaccuracies in its predictions.

References

[1] Quan Minh Bui, Dinh-Truong Do, Nguyen-Khang Le, Dieu-Hien Nguyen, Khac-Vu-Hiep Nguyen, Trang Pham Ngoc Anh, and ML Nguyen. 2023. 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)*.

[2] Quan Minh Bui, Dieu-Hien Nguyen, Dinh-Truong Do, Khac-Vu-Hiep Nguyen, and Minh Le Nguyen. 2023. JNLP @COLIEE-2023: Data Augmentation and Large Language Model for Legal Case Retrieval and Entailment. In *Proceedings of COLIEE 2023 Workshop*. ACM, Braga, Portugal, (June 2023), 10.

[3] Michel Custeau and Diana Inkpen. 2023. Performance of individual models vs. agreement-based ensembles for case entailment. *Proceedings of the 2023 Competition on Legal Information Extraction/Entailment (COLIEE)*, 58–62.

[4] [SW] Michael Han Daniel Han and Unsloth team, Unsloth 2023. URL: http://github.com/unslothai/unsloth.

[5] 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 *JSAI International Symposium on Artificial Intelligence*. Springer, 109–124.

[6] Jiawei Han, Yichao Lu, Haoyan Liu, Xinyi Wang, Fei Liu, and Yiming Yang. 2021. Promptsource: a framework for systematic prompt engineering. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. https://arxiv.org/abs/2104.05267.

[7] Edward J. Hu, Yelong Shen, et al. 2021. Lora: low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*. https://arxiv.org/abs/2106.09685.

[8] Komal Kumar et al. 2025. LLM Post-Training: A Deep Dive into Reasoning Large Language Models. (2025). https://arxiv.org/abs/2502.21321 arXiv: 2502.21321 [cs.CL].

[9] Yasuhiro Aoki Masaharu Yoshioka. 2023. Hukb at coliee 2023 statute law 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)*.

[10] Chau Nguyen and Le-Minh Nguyen. 2024. Employing Label Models on ChatGPT Answers Improves Legal Text Entailment Performance. (2024). https://arxiv.org/abs/2401.17897 arXiv: 2401.17897 [cs.CL].

[11] Chau Nguyen, Phuong Nguyen, Thanh Tran, Dat Nguyen, An Trieu, Tin Pham, Anh Dang, and Le-Minh Nguyen. 2024. Captain at coliee 2023: efficient methods for legal information retrieval and entailment tasks. *arXiv preprint arXiv:2401.03551*.

[12] 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.

[13] 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*. Toyotaro Suzumura and Mayumi Bono, (Eds.) Springer Nature Singapore, Singapore, 125–139. ISBN: 978-981-97-3076-6.

[14] 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 *JSAI International Symposium on Artificial Intelligence*. Springer, 183–199.

[15] 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 *JSAI International Symposium on Artificial Intelligence*. Springer, 200–211.

[16] Takaaki Onaga, Masaki Fujita, and Yoshinobu Kano. 2024. Contribution Analysis of Large Language Models and Data Augmentations for Person Names in Solving Legal Bar Examination at COLIEE 2023. *The Review of Socionetwork Strategies*, 18, 123–143. doi:10.1007/s12626-024-00155-5.

[17] Juliano Rabelo, Mi-Young Kim, and Randy Goebel. 2023. Transformer-based legal information extraction. In *Workshop of the tenth competition on legal information extraction/entailment (COLIEE'2023) in the 19th international conference on artificial intelligence and law (ICAIL)*.

[18] 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, 145–165.

[19] An Yang et al. 2024. Qwen2 technical report. (2024). https://arxiv.org/abs/2407.10671 arXiv: 2407.10671 [cs.CL].

Received 1 April 2025; revised 20 May 2025; accepted 5 May 2025

LLMs, Knowledge Graphs, and Hybrid Search: Task-Specific Approaches to Legal AI in COLIEE

Sabine Wehnert
Otto von Guericke University Magdeburg
Magdeburg, Germany

Bhavya Baburaj Chovatta Valappil
Otto von Guericke University Magdeburg
Magdeburg, Germany

Ernesto William De Luca
Otto von Guericke University Magdeburg
Magdeburg, Germany
Leibniz Institute for Educational Media | Georg Eckert Institute
Braunschweig, Germany

Abstract

We describe Otto von Guericke University Magdeburg's participation in all COLIEE 2025 tasks, including legal case retrieval, legal case entailment, statute law retrieval, legal textual entailment, and the pilot task on Japanese tort law. Our systems combine traditional IR methods, lightweight LLMs, and legal metadata using techniques such as proposition-based reformulation, chunked summarization, judge-aware reranking, and silver data fine-tuning. Despite limited resources, our models performed competitively, especially in legal case entailment and rationale extraction. We present our methodology, compare against top systems, and reflect on challenges in domain adaptation and hybrid modeling for legal NLP.

CCS Concepts

• **Theory of computation** → Shortest paths; • **Information systems** → *Expert systems*; *Content analysis and feature selection*; *Query reformulation*; **Language models**; *Top-k retrieval in databases*; *Question answering*; *Expert search*; Graph-based database models; • **Applied computing** → **Law**.

Keywords

Norm Retrieval, Legal Case Retrieval, Entailment Classification, Legal Judgment Prediction, BM25Plus, Large Language Models, Knowledge Graphs, Hybrid Search Algorithms

ACM Reference Format:
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, June 20, 2025, Chicago, USA*. ACM, New York, NY, USA, 10 pages.

1 Introduction

Legal judgment prediction and retrieval tasks pose significant challenges for NLP systems due to the complexity, length, and formal structure of legal texts. The COLIEE 2025 competition provides an opportunity to evaluate models across diverse legal reasoning tasks [9], including case retrieval, statute law application, textual entailment, and tort prediction.

COLIEE 2025, Chicago, USA

Our team from Otto von Guericke University Magdeburg (OVGU) participated in all COLIEE 2025 tasks. We adopted a unified strategy emphasizing efficient hybrid pipelines that combine symbolic and statistical methods. Our systems used traditional IR methods such as BM25Plus for initial candidate selection and incorporated LLMs via the Ollama framework for tasks requiring deeper semantic modeling or natural language generation. To enable efficient experimentation, we used quantized local models that allowed for large-scale inference. This paper describes the methods and results of our submissions.

2 Background

In this section, we briefly introduce two core concepts that played a significant role across our competition submissions: the use of quantized models via the Ollama framework, and the distinction between BM25 and its extension, BM25Plus.

2.1 Ollama and Model Quantization

Ollama[1] is an open-source framework designed to simplify the deployment and usage of large language models (LLMs) on local machines. Unlike platforms such as Hugging Face, which typically distribute models in their original floating-point precision (e.g., FP16 or BF16), Ollama distributes models in quantized formats (e.g., Q4, Q5, or Q8). Quantization refers to the process of reducing the numerical precision of a model's weights and activations - typically from 16- or 32-bit floating point to 4-, 5-, or 8-bit integers - resulting in smaller model sizes and reduced memory and computational requirements [4]. While quantization can introduce a minor loss in model accuracy, it enables efficient inference on consumer-grade hardware. This makes it possible to run large-scale models, such as LLaMA or Phi-series models locally without relying on cloud infrastructure. In our work, we primarily used models distributed via Ollama in quantized form, which allowed for rapid experimentation and integration into ensemble pipelines. It is important to note that these quantized versions are generally not directly compatible with Hugging Face model architectures and tooling unless explicitly converted.

2.2 BM25 and BM25Plus

BM25 is a widely used ranking function in information retrieval, particularly effective in tasks such as statute law retrieval [13], legal case retrieval [5, 10], and legal case entailment [7]. It ranks

[1] https://ollama.com

documents based on a combination of term frequency (TF), inverse document frequency (IDF), and document length normalization.

In the implementation[2] we used, the scoring function for a document D with respect to a query Q is defined as:

$$\text{BM25}(D,Q) = \sum_{t \in Q} \text{IDF}(t) \cdot \frac{f(t,D) \cdot (k_1+1)}{f(t,D) + k_1 \cdot \left(1 - b + b \cdot \frac{|D|}{\text{avgdl}}\right)}$$

where:

- $f(t,D)$ is the frequency of term t in document D,
- $|D|$ is the length of document D,
- avgdl is the average document length in the corpus,
- k_1 and b are tunable hyperparameters controlling term frequency saturation and length normalization.

The inverse document frequency (IDF) component reflects the importance of a term across the entire document collection. It assigns higher weight to rarer terms, which are assumed to be more discriminative for relevance. A common form of IDF used in BM25 is:

$$\text{IDF}(t) = \log\left(\frac{N - n(t) + 0.5}{n(t) + 0.5}\right)$$

where N is the total number of documents in the collection, and $n(t)$ is the number of documents containing term t. This formulation prevents division by zero and avoids negative IDF values.

$$\text{BM25}(D,Q) = \sum_{t \in Q} \text{IDF}(t) \cdot \left(\frac{f(t,D) \cdot (k_1+1)}{f(t,D) + k_1 \cdot \left(1 - b + b \cdot \frac{|D|}{\text{avgdl}}\right)}\right)$$

BM25Plus is an extension of BM25 designed to address two known limitations: (1) over-penalization of long documents due to the length normalization term, and (2) assigning zero scores to low-frequency terms with small IDF values. The BM25Plus scoring function introduces a tunable additive constant δ, which boosts all term contributions uniformly:

$$\text{BM25+}(D,Q) = \sum_{t \in Q} \log\left(\frac{N+1}{f(t,D)}\right) \cdot \left(\frac{f(t,D) \cdot (k_1+1)}{f(t,D) + k_1 \cdot \left(1 - b + b \cdot \frac{|D|}{\text{avgdl}}\right)} + \delta\right)$$

The additional δ term improves the ranking of underrepresented but potentially relevant documents, and mitigates the impact of structural length variations. In our experiments, we employed BM25Plus both as a standalone retriever and as a component in hybrid scoring strategies for case retrieval and paragraph selection.

3 Methods

3.1 Task 1: Legal Case Retrieval

During the COLIEE 2025 submission process for Task 1, our team submitted two runs. Due to an internal miscommunication under time constraints, the first run (OVGU1) was mistakenly generated using predictions on the training set rather than the official test data. This error resulted from a team member working with a custom data split without access to the official test queries. As this run does not reflect meaningful evaluation performance, it is listed as `ignore_task1_ovgu1.txt` on the leaderboard and should be disregarded. The valid submission considered in our analysis is **OVGU2**, which was prepared using the correct test set and methodology. All subsequent discussion and evaluation in this paper refer exclusively to OVGU2. The approach can be summarized in three main steps: initial retrieval with BM25Plus, fragment reformulation via legal propositions, and final reranking using a judge citation matrix.

3.1.1 Initial Retrieval. Each base case was processed to extract all entailed fragments marked by `FRAGMENT_SUPPRESSED`. These fragments were then treated as individual sub-queries. We applied BM25Plus retrieval across the full case corpus using preprocessed tokens (lowercased, stemmed[3], punctuation-[4] and stopword-removed). Additionally, we removed French content from bilingual Canadian legal documents using language detection[5] to focus retrieval on English content. We also used regular expressions to extract decision years from all cases and excluded any retrieved case that was dated after the base case, ensuring temporal consistency in citations.

3.1.2 Query Reformulation using Propositions. To improve semantic alignment between query fragments and target cases, we reformulated each fragment into a proposition [2] using two LLMs: `phi4`[6] and `qwen2.5:32b`[7]. The prompt is shown in Listing 1, and contains one of the examples provided by Curran and Conway [2]. Propositions generated by both models were then used as additional queries for retrieval. All top 10-ranked results from each source (original fragment, phi4, qwen2.5) were merged for further processing.

```
Convert the following fragment into a clear, concise legal
      proposition. Keep it factual and neutral. Ensure the
      proposition is understandable in isolation.

Example Fragment: On this point, the Respondent argues that the
      Officer's conclusion that there was insufficient evidence
      cannot be read in isolation and must be considered in the
      context of the findings and summary of evidence prior to such
      conclusion...

Example Proposition: The Court should not interfere with the Officer
      's decision unless it is outside the range of acceptable
      outcomes.

Now generate a proposition for this fragment: [FRAGMENT TEXT]
```

Listing 1: Prompt used to generate legal propositions

3.1.3 Case-Level Ensembling and Filtering. For each query case, we combined retrieval results from the original fragments and both LLM-generated propositions using a majority voting scheme. A case was included in the final set if it was retrieved by at least two of the three retrieval methods. To reduce false positives, we additionally applied a relative score filtering threshold of 0.2 with respect to the top retrieved case.

[2] https://pypi.org/project/rank-bm25/

[3] https://www.nltk.org/api/nltk.stem.SnowballStemmer.html
[4] https://www.nltk.org/api/nltk.tokenize.punkt.html
[5] https://pypi.org/project/langdetect/
[6] from https://ollama.com/library/phi4
[7] https://ollama.com/library/qwen2.5

3.1.4 Judge-Aware Reranking. Finally, we leveraged metadata about the presiding judges to rerank the retrieved cases, which we extracted with regular expressions. We constructed a judge citation matrix from the training data, recording how often a judge in a base case cited another judge. This matrix was used to boost retrieval scores when there was a historical citation link between judges. The reranked results were then used to generate the final prediction.

3.2 Task 2: Legal Case Entailment

The legal case entailment task requires the prediction of one or multiple paragraphs of a noticed case which are entailed by a given fragment of the base case. For this task, we first performed data profiling on the training data and learned that within this dataset at most 5 paragraphs can be entailed by a base case. Our approach therefore consists of base case summarization, top-5 candidate selection with BM25Plus, LLM-based entailment prediction, and final ensembling strategies to combine prediction by individual models. We describe each step in the following.

```
Summarize this portion of a legal base case, focusing only on its
      relevance to the following legal snippet. Note that the snippet
       is a part of this base case. Summarize the following portion
      of the legal case in 150 words or fewer. Provide only the
      summary - do not include any explanations, reflections, or
      comments.

Snippet:
[SNIPPET]

Base Case Portion:
[CHUNK]

Summary:
```

Listing 2: Prompt used to summarize a portion of a legal base case

```
Given the legal snippet and summaries of portions of a legal base
      case, create a concise summary of the base case, focusing only
      on the relevance to the following legal snippet. Note that the
      snippet is a part of this base case. Summarize the portions of
      the legal case in 250 words or fewer. Provide only the summary
      - do not include any explanations, reflections, or comments.

Snippet:
[SNIPPET]

Portion Summaries:
[SUMMARY 1]
[SUMMARY 2]
...

Final Summary:
```

Listing 3: Prompt used to generate the final base case summary from chunked summaries

3.2.1 Base Case Summarization. Given the base case fragment from which we aim to identify an entailed paragraph in the noticed case, we hypothesized that the prediction task could be facilitated by providing additional context. To this end, we decided to generate summaries of the full base cases and subsequently prompt the LLMs with both the entailed fragment and the corresponding base case summary. This additional context was intended to support more accurate entailment decisions for a given paragraph. For generating these summaries, we selected the model `phi4`[8]. In instances where the full base case exceeded a predefined character limit - due to constraints on prompt length - we divided the text into smaller chunks before summarization to ensure compatibility. Each chunk of the base case was summarized using a snippet-aware prompt that focused on the portion's relevance to the entailed fragment (see Listing 2). These partial summaries were then combined into a final base case summary using a follow-up prompt (see Listing 3).

3.2.2 Top-5 Candidate Selection with BM25Plus. To reduce the computational load for downstream LLM-based entailment prediction, we first employed a retrieval step to identify a small set of candidate paragraphs likely to be entailed by the base case fragment. Based on data profiling, we observed that each base case entailed at most five paragraphs in the training data. This motivated the use of a top-k retrieval strategy with $k = 5$.

We applied the BM25Plus retrieval algorithm to rank the paragraphs of each noticed case with respect to the given entailed fragment from the base case. For each query, both the query and candidate paragraphs were preprocessed via lowercasing, punctuation removal, tokenization, stopword removal, and stemming using the Snowball stemmer.

The BM25Plus scoring model was then used to rank the candidate paragraphs based on lexical similarity to the query. For each query instance, the top 5 highest-scoring paragraphs were selected as candidate entailed paragraphs. When evaluated on the training set, this approach yielded a precision of 0.2023, a recall of 0.8675, and an F1-score of 0.3220. While the relatively low precision reflects the presence of false positives, the high recall indicates that the majority of truly entailed paragraphs were successfully retrieved. This high recall is important for ensuring that relevant candidates are passed to the LLMs in the next phase for more accurate entailment prediction.

```
Given a snippet from a legal base case and the summary of the base
      case, determine if this paragraph from a noticed case entails
      the snippet. Do not explain your reasoning, just give the
      entailment label (Yes/No).

Paragraph of noticed case: [PARAGRAPH]

Snippet: [SNIPPET]

Base Case Summary: [SUMMARY]

Entailment (Yes/No):
```

Listing 4: Prompt used for LLM-based Entailment Prediction

3.2.3 LLM-based Entailment Prediction. With the prompt in Listing 4, we let the following LLMs predict the entailment label:

- gemma3:12b[9]
- wizardlm2:7b[10]
- phi4[11]
- gemma2[12]
- deepseek-r1:32b[13]

[8] Available at https://ollama.com/library/phi4
[9] from https://ollama.com/library/gemma3
[10] from https://ollama.com/library/wizardlm2
[11] from https://ollama.com/library/phi4
[12] from https://ollama.com/library/gemma2
[13] from https://ollama.com/library/deepseek-r1

3.2.4 Ensembling Strategies. After predicting the entailment labels with different LLMs, we employ three strategies for combining their results, which then form our three submitted runs for Task 2.

Run 1 (OVGU1) uses a *majority voting ensemble* across predictions from the LLMs, where each model previously classified a query-paragraph pair as either Yes, No, or Unknown. For each pair, the system counts how many models vote Yes and how many vote Unknown. If more than half of the models vote Yes, the paragraph is selected as relevant (entailed). If none of the models vote Yes and all vote Unknown, the system concludes that there is insufficient information and assigns the label Unknown. However, if there is no majority in favor of Yes and the votes include a mix of No and Unknown, the system defaults to No, under the assumption that the lack of strong affirmative evidence indicates non-entailment. In cases where no paragraph is selected for a query after this process, the system employs a fallback mechanism, selecting the top-ranked paragraph based on BM25Plus retrieval scores to ensure coverage for every query.

Run 2 (OVGU2) implements a *precision-optimized ensemble* strategy. In this approach, only paragraphs predicted as entailed (Yes) by a majority of LLMs are considered. Among these, up to two paragraphs per query are selected based on their BM25Plus ranking, which reflects lexical similarity between the query and the paragraph. This approach combines the semantic reasoning abilities of LLMs (via entailment predictions) with traditional keyword-based relevance scoring. As in Run 1, a fallback logic is used: if no paragraph receives a confident Yes prediction from the ensemble, the system selects the top BM25Plus-ranked paragraph to ensure that every query receives at least one prediction.

Run 3 (OVGU3) extends the *precision-optimized strategy with an added BM25Plus score difference threshold* to further refine paragraph selection. After identifying the top Yes-predicted paragraphs, the system selects the top-ranked one and includes additional paragraphs only if their BM25Plus scores differ by less than 0.635 from the preceding paragraph. This encourages high precision by limiting the selection to top-scoring candidates that are close in relevance. We chose this threshold after performing a grid search, inspired by team AMHR's strategy [7] in 2024. Equal to the previous runs, fallback logic is applied when no paragraph meets the ensemble or threshold criteria, defaulting to the highest BM25Plus-ranked paragraph.

3.3 Task 3: Statute Law Retrieval

Statute Law Retrieval (Task 3) requires retrieving pertinent articles from a fixed set of Japanese civil code articles (translated to English) in response to a given query. For task 3, we implemented the Retrieval Augmented Generation (RAG) approach, using a knowledge graph across all three runs. In our two-stage methodology, semantic indexes within Neo4j are leveraged during the initial retrieval phase to identify candidate nodes, followed by applying a Large Language Model (LLM) to formulate the final response.

3.3.1 Knowledge Graph. Our knowledge graph is composed of nodes and edges, where nodes represent articles and auxiliary data, and edges capture their semantic or structural relationships, such as citations, references, etc., or hierarchical organization. The primary node type is the civil code article, supplemented by auxiliary

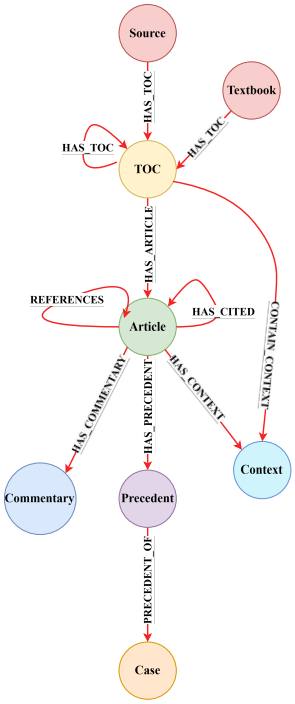


Figure 1: Schema of Knowledge graph

information drawn from external sources. To improve navigability and preserve the structure of legal documents, we extracted metadata such as part, chapter, section, and subsection of the Japanese Civil Code and encoded this information as table of contents (TOC) nodes. These TOC nodes enable structured linking between articles and their hierarchical context. The auxiliary data integrated into the graph can be grouped into two categories:

- **Online Crawled Data:** For each article, we retrieved additional content from the Japanese Wikibooks website[14], which we translated into English. This content included three types of legal knowledge:
  (1) *Precedents:* Representing past court judgments, these were linked to a corresponding "CASE" via a case identifier.
  (2) *Commentaries:* Legal explanations and interpretations discussing the article's meaning and application.
  (3) *Reference Articles:* Articles that are cited or required for the application of another article, enabling inter-article dependency modeling.
- **Textbook Knowledge:** We extracted situational applications and legal reasoning from domain-specific textbooks, following the method proposed by Wehnert et al. [11, 12]. Using a rule-based system incorporating regular expressions and part-of-speech tagging, we identified relevant legal context sentences and incorporated them as "context" nodes. To preserve their position in the textbook hierarchy, associated TOC metadata (e.g., section and subsection titles) were also added as TOC nodes, forming a structured path from high-level topics to specific legal contexts.

[14] https://ja.wikibooks.org/wiki/

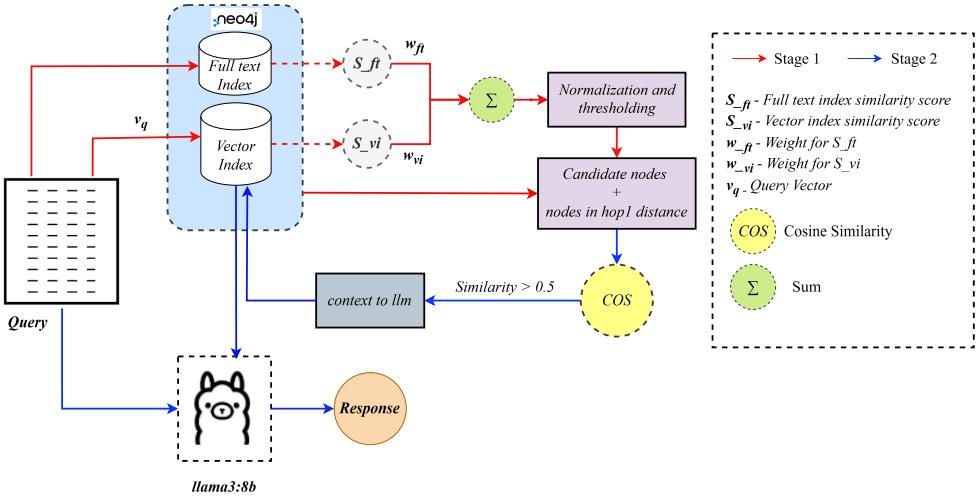


Figure 2: Overview of RAG approach for Task 3

Run	Threshold	W_ft	W_vi
OVGU1	0.65	0.0	1
OVGU2	0.9	0.1	0.9
OVGU3	0.9	0.0	1

Table 1: Score threshold for each run for Task 3 - Stage 1

In total, the knowledge graph contains eight distinct node types and nine types of semantic or structural relationships. The schema is illustrated in Figure 1. All three of our submitted runs for Task 3 were built on this knowledge graph using a two-stage Retrieval-Augmented Generation (RAG) architecture, differing only in their retrieval parameters (see Table 1). An overview of the end-to-end pipeline is provided in Figure 2.

3.3.2 Stage 1: Retrieval using semantic indexes. To retrieve relevant nodes from our knowledge graph, we employed Neo4j's built-in semantic indexing capabilities, leveraging both full-text and vector-based indexes. The graph was constructed with 'desc' properties assigned to key node types, such as articles, commentaries, precedents, and context nodes, containing the textual content used for indexing. Full-text indexes in Neo4j retain the individual words of the input and enable context-aware matching through Apache Lucene. Unlike exact or substring matching, full-text search supports flexible retrieval by evaluating the semantic proximity of query terms to indexed content. Upon execution, each query yields an approximate similarity score for every indexed node. To complement this, we also constructed vector indexes using the BGE-M3 model[15], fine-tuned on the COLIEE 2025 Task 3 training data. We fine-tuned the model following the approach outlined in the FlagEmbedding repository[16], specifically leveraging hard negative mining to improve retrieval quality. This model was used to compute dense vector embeddings for both queries and node descriptions. Vector indexing enabled retrieval based on embedding similarity. For each query, scores from both retrieval strategies - full-text and vector - were computed and linearly combined using task-specific weights (ensuring they sum to 1). The resulting composite score was then normalized using min-max scaling within the range [0, 1]. Only nodes exceeding a predefined score threshold were retained as candidates (see Table 1).

3.3.3 Stage 2: Generation using LLM. In the second stage, we focused on response generation using a Large Language Model (LLM), guided by the candidate nodes retrieved during Stage 1. We hypothesized that expanding the context around each candidate node could improve recall and provide the LLM with richer semantic signals, thereby enhancing the accuracy and completeness of its output. To this end, we augmented each candidate node by retrieving all neighboring nodes within a one-hop distance in the knowledge graph. These adjacent nodes - often containing relevant auxiliary information such as commentaries or precedents - were appended to the candidate list. For nodes whose textual content (stored in the `desc` property) exceeded a manageable input size for the LLM, we segmented the text into chunks of up to 3000 characters. This chunk length was chosen to accommodate the average size of article descriptions while preserving semantic coherence. The resulting chunks constituted the input documents for response generation. To mitigate hallucination and enhance the precision of responses, we applied a multi-stage filtering and iterative generation strategy inspired by the work of Nguyen and Satoh (2024) [6]. This process ensured that the final input to the LLM consisted of non-redundant content. For the generation step, we employed the `LLaMA3:8B` model[17], integrated via the LangChain `RetrievalQA`[18] pipeline. The query was posed using a structured prompt (see Listing 5), and the associated Neo4j-based vector store served as the underlying knowledge base for document retrieval. We set the LLM temperature to 0 for deterministic responses. Using LangChain's

[15] https://huggingface.co/BAAI/bge-m3
[16] https://github.com/FlagOpen/FlagEmbedding
[17] https://ollama.com/library/llama3
[18] https://api.python.langchain.com/en/latest/chains/langchain.chains.retrieval_qa.base.RetrievalQA.html

vector retriever, we limited retrieval to 10 documents. Higher values of k increased prompt size and reduced efficiency, while k = 10 offered the best balance between retrieval quality and performance.

```
You are a good "Legal Consultant" who is an expert in giving legal
      advice.
You are provided with information on the Japanese civil codes(
      Articles) and their descriptions (English Translation).
Additionally, the information given also contains Precedents,
      Commentary, and different contexts of civil codes/(Articles).
The question/query/statement is always a legal context/statement,
      and all the data is in English.
From the set of civil codes with description given, find the
      relevant civil codes for the following query/statement:
{question}

### Relevance Criteria:
- An article/civil code is "Relevant", if the query sentence can be
      answered **Yes/No** based on the meaning of the article/civil
      code.
- If multiple articles/civil code together (e.g., "A", "B", and "C")
       are needed to answer the query, then all of them are
      considered relevant.
- If a query can be answered independently by multiple articles (e.g
      ., "D" and "E"), then both are relevant.

### **Rules:**
1. **Strictly base your answer on the provided information/data.**
      Do not infer information outside of the provided knowledge.
2. **Only return relevant articles.** Do not include any articles
      that are not directly related.
3. **If no relevant articles are found, explicitly say:** "I do not
      know the answer."
4. **Do not generate any explanation or reasoning.** Only list the
      article/civil code, e.g. Article 1.
5. **Do not make up answers or add out-of-context information to
      answer or the query**
6. **KEEP THE ANSWER SHORT AND TO THE POINT**

### Example Answer if you know the relevant articles:
Relevant Articles:
1. Article 562
2. Article 563

### Example Answer if you do not know the relevant articles:
I do not know the answer.
```

Listing 5: Prompt used for Statue Law Retrieval

3.4 Task 4: Legal Textual Entailment

3.4.1 Silver Data Creation. To support model training for Task 4, we constructed a silver dataset by using predictions from 9 LLMs:

- "dolphin-llama3:8b"[19],
- "llama3:8b"[20],
- "wizardlm2:7b"[21],
- "qwen:14b"[22],
- "gemma:7b"[23],
- "orca2:13b"[24],
- "deepseek-v2:16b"[25],
- "superdrew100/phi3-medium-abliterated"[26], and
- "llama2:7b"[27].

[19] from https://ollama.com/library/dolphin-llama3
[20] from https://ollama.com/library/llama3
[21] from https://ollama.com/library/wizardlm2
[22] from https://ollama.com/library/qwen
[23] from https://ollama.com/library/gemma
[24] from https://ollama.com/library/orca2
[25] from https://ollama.com/library/deepseek-v2
[26] from https://ollama.com/superdrew100/phi3-medium-abliterated
[27] from https://ollama.com/library/llama2

Each model was prompted using the COLIEE 2025 training hypotheses, accompanied by problem type definitions as described by Hoshino et al. [3]. The task for the LLMs was threefold: to identify relevant legal problem types for each query, to predict the correct entailment label, and to provide a corresponding reasoning. The prompt is shown in Listing 6, and the premises $t1$ and hypotheses $t2$ are dynamically inserted.

This problem type-oriented prompting was designed to encourage the models to incorporate aspects of legal complexity and reasoning structure in their entailment predictions. In cases where a model's response was invalid (i.e., not automatically parseable), the prompt was reissued until a syntactically valid response was obtained. Once predictions were generated for all training queries, we compared the predicted entailment labels against the gold-standard labels. For each query, we selected the first valid model response that included a correct entailment label and non-empty entries for both problem type and reasoning. These entries formed our silver dataset - so named because, while the entailment labels were verified against ground truth, the associated problem types and justifications were not manually validated. This approach is inspired by the work of Pompili et al. [8] who used implicit relevance feedback as silver data for training question answering models.

3.4.2 Fine-Tuning LLMs. Using the silver dataset, we fine-tuned the Phi-3-medium-4k-instruct[28] model (resulting in fine_tuned_phi3) and gemma-1.1-7b-it[29] (which we now refer to as ft_gemma_7b). For fine-tuning, we adopted a causal language modeling (CLM) setup using Hugging Face's Transformers library. Each training instance was composed of a prompt containing the premise, hypothesis, and problem type definitions, followed by a structured JSON output (see Listing 6). The input and target output were concatenated into a single sequence and tokenized together. Tokens corresponding to the prompt portion were masked with `-100` to exclude them from the loss computation, so that only the generation of the JSON was learned. Training was carried out using a batch size of 1 and gradient accumulation over 2 steps. We trained the models for 5 epochs with early stopping (patience of 2) and cosine learning rate scheduling. The learning rate was set to 1×10^{-5}, and we used the `adamw_bnb_8bit` optimizer for memory efficiency. A custom data collator dynamically padded input sequences and labels up to a maximum length of 1024 tokens. The best-performing model checkpoint was selected based on training loss.

3.4.3 Predicting Statute Entailment with LLMs. Since we kept all queries from the training data with the prefixes `"R03"` and `"R04"` as validation data, we saw performance gains with the fine-tuned models, compared to their quantized Ollama parts. This is why the first two runs were entirely based on the predictions of these models, while the third run became an ensemble setting of the fine-tuned models and other well-performing pretrained models.

Run 1 (OVGU1) The fine_tuned_phi3 model was subsequently used for final entailment predictions. As during silver data creation, responses from the model were automatically parsed, and any invalid outputs triggered repeated prompting until a valid response was obtained.

[28] From https://huggingface.co/microsoft/Phi-3-medium-4k-instruct
[29] From https://huggingface.co/google/gemma-1.1-7b-it

```
f"""Analyze legal entailment:

Premise: {t1}
Hypothesis: {t2}

Task:
1. Identify applicable problem types.
2. Determine if the hypothesis follows from the premise.
3. Explain your reasoning.
4. Output JSON: {{"problem_types": [...], "reasoning": "...", "
      entailment": "Y" or "N"}}

**Problem Type Definitions:**
"Conditional sentence extraction": "Extract each conditional
      sentence from problem sentences and related articles.",
"Person role extraction": "Identify the role of a person in the
      problem sentence (e.g., underage, buyer, obligor).",
"Person relationship extraction": "Determine the positions and roles
       among multiple people in the sentence.",
"Morphological analysis": "Include specific case particles to make
      the sentence easier to analyze.",
"Anaphoric analysis": "Clarify what is being referenced when part of
       the sentence is omitted.",
"Ambiguity resolution": "Analyze sentences with ambiguous
      expressions.",
"Semantic role extraction": "Identify whether the noun in the
      problem sentence is the principal actor or the object of the
      action/method.",
"Verb paraphrasing": "Analyze sentences where the verb has been
      paraphrased.",
"General dictionary": "Analyze non-legal terms used in everyday life
      .",
"Predicate argument structure": "Clarify the behavior of problem
      sentences.",
"Negative interpretation": "Determine how a negative form affects
      true/false judgments.",
"Legal term dictionary": "Analyze sentences where legal terms have
      been paraphrased (e.g., 'limited ability person' -> 'minor').",
"Implication relation": "Identify hidden intent in sentences (e.g.,
      if a person requests something in a trial, they are a plaintiff
      ).",
"Dependency": "Focus on the relationship between the subject and
      predicate or parallel sentence relations.",
"Refer to article": "Identify references to other articles within
      the text.",
"Paraphrase": "Analyze sentences where terms other than verbs have
      been paraphrased.",
"Bullet": "Analyze articles that use bullet points.",
"Digitization of priorities": "Analyze articles where the priority
      order of effectiveness is indicated using bullet points."

**DO NOT include explanations or extra text before or after the JSON
      .**

**Output the JSON object NOW:**"""
```

Listing 6: Prompt used for Silver Data Generation, LLM fine-tuning, and Entailment Prediction

Run 2 (OVGU2) We used ft_gemma_7b for the final prediction of the entailment label. If invalid responses (i.e., not possible to parse automatically) were given for a query, the model was prompted again until it gave a valid response.

Run 3 (OVGU3) employs an ensemble of both pretrained and fine-tuned language models to improve entailment label prediction. The ensemble consists of six models (see their details in the Section 3.4.1 about Silver Data Creation): four pretrained models (gemma:7b, superdrew100/phi3-medium-abliterated, wizardlm2:7b, llama3) and two fine-tuned models (ft_gemma_7b and fine_tuned_phi3). The ensemble follows a straightforward majority voting strategy across model predictions. In cases of a tie (possible due to the even number of models) the final label is taken from fine_tuned_phi3, which demonstrated strong performance on the training set. The label defaults to "N" (i.e., non-entailment) if fine_tuned_phi3 fails to produce a valid prediction,

3.5 Pilot Task: Legal Judgment Prediction for Japanese Tort Cases

The pilot task consisted of two subtasks: predicting individual claim labels (i.e., whether each claim was accepted), and determining the overall court decision (i.e., whether the court ruled in favor of the plaintiff). Across all runs, we employed a two-step approach: a machine learning model for claim classification and a rule-based method for deriving the final court decision.

This approach was motivated by data profiling of the training set (see Figure 3). An analysis of claim acceptance ratios revealed a strong correlation between the proportion of accepted claims and the court's final decision. In cases where the decision was favorable to the plaintiff, the distribution of plaintiff claim acceptance ratios was heavily skewed toward 1.0, indicating that most or all claims were accepted. Conversely, the corresponding defendant ratios in these cases were generally lower, suggesting minimal success for the defense. In contrast, when the court ruled against the plaintiff, their claim acceptance ratios clustered near zero, while defendant ratios were higher, often approaching full acceptance. These opposing distributions underscore a clear inverse relationship between party success and court outcomes, highlighting the predictive potential of accepted-claim ratios in legal decision modeling.

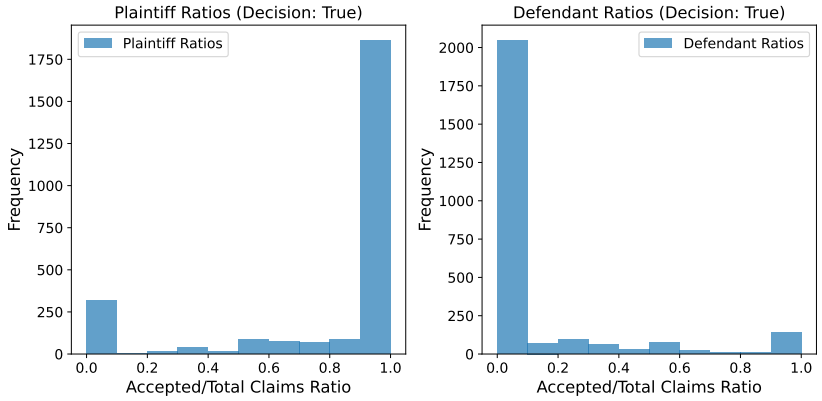
```
Summarize this portion of a legal case's undisputed facts in 300
      words or fewer. Provide only the summary - do not include
      explanations or reflections.

Portion:
[CHUNK TEXT]

Summary (in Japanese):
```

Listing 7: Prompt used to summarize a portion of undisputed fact (word count parameter for run OVGU3)

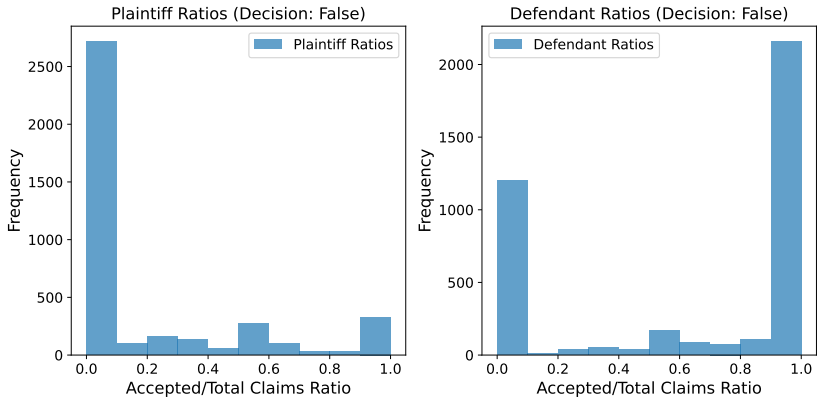
```
Given the following summaries of portions of a legal case, create a
      concise final summary in 400 words or fewer. Provide only the
      summary - do not include explanations or reflections.

Portion Summaries:
[SUMMARY 1]
[SUMMARY 2]
...

Final Summary in Japanese:
```

Listing 8: Prompt used to generate the final summary from chunked summaries (word count parameter for run OVGU3)

While the undisputed facts provided important contextual information, they consumed a substantial portion of the prompt's token budget. To address this, we implemented a chunk-based summarization strategy that respected the context limitations of the LLM. First, each chunk of the text was summarized individually (see Listing 7). Then, these partial summaries were combined into a final concise summary (see Listing 8). To facilitate efficient parameter and model selection, we restricted training to a smaller subset comprising the first 100 instances. Initially, we combined acceptance ratios with



(a) Accepted/Total Claims Ratios for cases in the training data with court decision = True



(b) Accepted/Total Claims Ratios for cases in the training data with court decision = False

Figure 3: Distribution of accepted claim ratios by party type (plaintiff vs. defendant) across court decision outcomes.

the summarized undisputed facts as input to LLMs for claim classification. However, a rule-based approach ultimately yielded better performance and was preferred due to its lower computational cost.

In **Run OVGU1**, we used the `aya-expanse:8b`[30] language model to directly predict the label for each claim. Each prompt included a task description, the claim to be classified, and the summary of the undisputed facts appended at the end (see Listing 9). This structure ensured that the model could still infer the task objective even in the event of input truncation.

```
Claim: [CLAIM TEXT]
Is the claim above accepted? Respond STRICTLY with 'true' or 'false'.

Summary of Undisputed Facts: [SUMMARY TEXT]
Respond STRICTLY with 'true' or 'false'.
```

Listing 9: Prompt used for LLM-based claim prediction

The court decision was predicted using a rule-based scoring function derived from correlation analysis on the training data. Specifically, we observed that the acceptance ratio of plaintiff claims was positively correlated with favorable outcomes ($r = 0.69$), while that of defendant claims was negatively correlated ($r = -0.55$). These absolute correlation values were used as weights in a linear decision function: score $= \alpha \cdot r_p - \beta \cdot r_d$

where r_p and r_d denote the acceptance ratios for the plaintiff and defendant, respectively, and $\alpha = |0.69|$, $\beta = |-0.55|$. If the resulting score exceeds a threshold of 0.5, the model predicts a favorable decision for the plaintiff; otherwise, it predicts an unfavorable outcome. This approach integrates signal strength from both parties' claims while maintaining interpretability and computational efficiency. Preliminary experiments showed that while `aya-expanse:8b` performed well in claim classification, it was not competitive in court decision prediction. However, its strength in processing undisputed facts made it valuable as a summarization tool for subsequent models.

In **Run OVGU2**, we used `aya-expanse:8b` to summarize the undisputed facts into a condensed version of up to 400 words. This summary was then passed to the `phi4`[31] model, which was prompted to classify each claim individually. The rule-based logic for court decision prediction remained unchanged from Run OVGU1.

[30] From https://ollama.com/library/aya-expanse

[31] From https://ollama.com/library/phi4

In **Run OVGU3**, we followed a similar pipeline but replaced `phi4` with `gemma3:12b`[32]. The undisputed facts were again summarized using `aya-expanse:8b`, but the maximum summary length was limited to 400 words because of better performance on training data, compared to 600 words. The claims were predicted individually, and the court decision was inferred using the same rule-based method described above.

The claim classification models yielded relatively low F1-scores on the training data, with Run OVGU1 achieving the best, though still uncompetitive, result of 65%. In Runs OVGU2 and OVGU3, although claim-level F1-scores were below 20%, legal case prediction accuracy exceeded 70% on the training subset.

4 Evaluation

4.1 Task 1

Our submission (OVGU2) achieved an F1-score of 0.1498, ranking 13th out of 21 total submissions, see Table 2. While this performance falls short of the top-ranked system (JNLP, F1 = 0.3353), it demonstrates moderate retrieval effectiveness given our lightweight hybrid approach combining lexical BM25Plus retrieval, LLM-generated propositions, and judge-based reranking. Future improvements may focus on optimizing semantic matching.

Team (Rank)	Run	F1	Precision	Recall
JNLP (1)	jnlpr&fe2.txt	**0.3353**	0.3042	0.3735
OVGU (13)	OVGU2	0.1498	0.1743	0.1313

Table 2: Task 1 (Legal Case Retrieval) results for the best-performing team and our submission. A total of 21 runs were submitted by 7 teams.

4.2 Task 2

Team (Rank)	Run	F1	Precision	Recall
NOWJ (1)	nowj003.txt	**0.3195**	0.3788	0.2762
OVGU (4)	OVGU2	0.2454	0.2759	0.2210
OVGU (9)	OVGU3	0.1965	0.2692	0.1547
OVGU (13)	OVGU1	0.1708	0.2400	0.1326

Table 3: Task 2 (Legal Case Entailment) results for the best-performing system and all OVGU runs. A total of 18 submissions were received from 6 teams.

Our best-performing system (OVGU2) obtained an F1-score of 0.2454, ranking 4th out of 18 total submissions (Table 3), with us being the second-best team after NOWJ. While the top-ranked system (F1 = 0.3195) showed stronger performance, our approach delivered competitive results, particularly considering its modular structure combining BM25Plus retrieval, base case summarization, and LLM-based entailment classification. The other two runs (OVGU3 and OVGU1) achieved lower scores, with F1 = 0.1965 and F1 = 0.1708, respectively, reflecting design variations in ensembling strategies.

[32]From https://ollama.com/library/gemma3

Team (Rank)	Run	F2	Precision	Recall
JNLP (1)	JNLP_RUN1	**0.7829**	0.7521	0.8184
OVGU (10)	OVGU3	0.5654	0.5940	0.5748
OVGU (11)	OVGU2	0.5577	0.5705	0.5641
OVGU (15)	OVGU1	0.4372	0.4338	0.4487

Table 4: Task 3 (Legal Case Entailment Ranking) results for the best system and all OVGU submissions. A total of 22 submissions were received from 7 teams.

4.3 Task 3

Our best system for Task 3 (OVGU3) achieved an F2-score of 0.5654, ranking 10th out of 22 submissions (see Table 4). OVGU2 followed closely with an F2 of 0.5577, while OVGU1 trailed at 0.4372. While the top-performing team (JNLP) attained an F2-score of 0.7829, our system maintained strong performance on ranking-oriented metrics, achieving Recall@30 of 0.9677. Results fell short of expectations, as validation on files prefixed with `R03`/`R04` showed ≈ 70% F2. We first suspected `BGE-M3` (trained on COLIEE 2022 [1]), but confirmed our validation split was unaffected. We now suspect Llama3, trained on undisclosed public data[33], that it may have seen COLIEE data of previous years, inflating our performance estimates. To further investigate this discrepancy in the performance, we evaluated the Precision@1 and Precision@2 and mean accurate precision (MAP) for both stage 1 and stage 2 for all the runs (Table 5). The results indicate a significant drop in performance from Stage 1 to Stage 2 in all runs. These findings corroborate our earlier concerns about the potential effects introduced by Llama3.

Run	Stage	Precision@1	Precision@2	MAP
OVGU1	Stage 1	0.6351	0.4041	0.7103
	Stage 2	0.5135	0.2174	0.4662
OVGU2	Stage 1	0.6892	0.3286	0.6585
	Stage 2	0.6338	0.2692	0.6021
OVGU3	Stage 1	0.6351	0.3333	0.6518
	Stage 2	0.6486	0.3750	0.5890

Table 5: Precision@1, Precision@2, and MAP for Task 3 on Stage 1 and Stage 2 for all the runs

4.4 Task 4

Our best run (OVGU1) scored an accuracy of 0.7432, placing 19th out of 30 submissions (Table 6). The top-ranked system (KIS3) reached a strong accuracy of 0.9054. Our other two runs, OVGU2 and OVGU3, yielded lower accuracies of 0.6081 and 0.6351, respectively. While OVGU1 showed competitive potential, further improvements are needed in handling nuanced legal entailment cases to close the gap to top-tier systems.

4.5 Pilot Task

Our results on the Pilot Task illustrate the strengths and limitations of combining LLM-based claim classification with rule-based aggregation for legal judgment prediction. For the Tort Prediction

[33]https://github.com/meta-llama/llama3/blob/main/MODEL_CARD.md

Team (Rank)	Run	Accuracy
KIS (1)	KIS3	**0.9054**
OVGU (19)	OVGU1	0.7432
OVGU (23)	OVGU3	0.6351
OVGU (26)	OVGU2	0.6081

Table 6: Task 4 (Legal Judgment Entailment Classification) results for the best-performing system and all OVGU runs. A total of 30 submissions were received from 11 teams.

Team (Rank)	Run	Accuracy
CAPTAIN (1)	JAIST-LJPJT25	**76.5%**
OVGU (9)	OVGU2	55.3%
OVGU (10)	OVGU3	53.2%
OVGU (11)	OVGU1	51.5%

Table 7: Pilot Task – Tort Prediction results for the top system and all OVGU runs. A total of 11 valid official submissions were recorded.

Team (Rank)	Run	F1-score (All)
KIS (1)	KIS5	**71.2%**
OVGU (8)	OVGU1	65.7%
OVGU (10)	OVGU2	48.6%
OVGU (11)	OVGU3	31.6%

Table 8: Pilot Task – Rationale Extraction results for the top system and all OVGU runs. A total of 11 valid official submissions were recorded.

subtask, our best-performing system (OVGU2) achieved 55.3% accuracy. The final court decision was predicted using a linear scoring function weighted by correlation values between party success and court outcomes. Surprisingly, on the training data this rule-based method yielded competitive performance even when the underlying claim-level predictions from LLMs (phi4 and gemma3) were weak (F1 < 20%). This outcome suggests that while the individual claim classifications were often incorrect, the aggregate signal across all claims - when processed through the scoring function - still aligned well with the correct court decision in many cases.

In contrast, the Rationale Extraction subtask was more sensitive to the accuracy of the claim labels. Our best result (65.7% F1) came from OVGU1, which used `aya-expanse:8b` for direct classification. OVGU2 and OVGU3, which relied on phi4 and gemma3 respectively, as well as on aye-expanse-generated summaries, performed considerably worse, indicating that these models struggled with the finer-grained reasoning required for rationale identification. This reinforces the importance of careful model selection for tasks where label precision is critical.

5 Conclusion

This paper presented our methods and findings from participating in all COLIEE 2025 tasks. We built hybrid pipelines combining BM25Plus retrieval, quantized local LLMs, fine-tuned models, rule-based logic, and legal metadata such as judge citations. While tasks such as legal case entailment and rationale extraction yielded competitive results, others exposed persistent challenges in semantic ranking and complex entailment reasoning. We observed that when LLM training data is undisclosed, there is a risk of overestimating performance - publicly available models may have been trained on data overlapping with validation sets. Our experiments suggest that modular designs enable scalable experimentation, especially under hardware or time constraints. However, reaching state-of-the-art results still demands careful integration of domain-specific reasoning, high-recall retrieval, and robust prompt engineering. In future work, we aim to refine our prompting strategies, rule-based components, and the final model selection process.

References

[1] Jianlyu Chen et al. 2024. M3-Embedding: Multi-Linguality, Multi-Functionality, Multi-Granularity Text Embeddings Through Self-Knowledge Distillation. In *Findings of the ACL 2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024*, Lun-Wei Ku et al. (Eds.). ACL, 2318–2335.

[2] Damian Curran and Mike Conway. 2024. 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 (Lecture Notes in Computer Science, Vol. 14741)*, Toyotaro Suzumura and Mayumi Bono (Eds.). Springer, 156–166.

[3] Reina Hoshino et al. 2019. Question answering system for legal bar examination using predicate argument structure. In *New Frontiers in Artificial Intelligence: JSAI-isAI 2018 Workshops, JURISIN, AI-Biz, SKL, LENLS, IDAA, Yokohama, Japan, November 12–14, 2018, Revised Selected Papers*. Springer, 207–220.

[4] Renren Jin et al. 2024. A Comprehensive Evaluation of Quantization Strategies for Large Language Models. In *Findings of the ACL 2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024*, Lun-Wei Ku et al. (Eds.). Association for Computational Linguistics, 12186–12215.

[5] Haitao Li et al. 2023. THUIR@COLIEE 2023: Incorporating Structural Knowledge into Pre-trained Language Models for Legal Case Retrieval. *CoRR* abs/2305.06812 (2023). arXiv:2305.06812

[6] Ha-Thanh Nguyen and Ken Satoh. 2024. ConsRAG: Minimize LLM Hallucinations in the Legal Domain. In *Legal Knowledge and Information Systems - JURIX 2024: The Thirty-seventh Annual Conference, Brno, Czech Republic, 11-13 December 2024 (Frontiers in Artificial Intelligence and Applications, Vol. 395)*, Jaromír Savelka, Jakub Harasta, Tereza Novotná, and Jakub Mísek (Eds.). IOS Press, 327–332.

[7] Animesh Nighojkar et al. 2024. AMHR COLIEE 2024 Entry: Legal Entailment and Retrieval. In *New Frontiers in Artificial Intelligence - JSAI International Symposium on Artificial Intelligence, JSAI-isAI 2024, Hamamatsu, Japan, May 28-29, 2024, Proceedings (Lecture Notes in Computer Science, Vol. 14741)*, Toyotaro Suzumura and Mayumi Bono (Eds.). Springer, 200–211.

[8] Filippo Pompili et al. 2019. Exploiting Search Logs to Aid in Training and Automating Infrastructure for Question Answering in Professional Domains. In *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Law, ICAIL 2019, Montreal, QC, Canada, June 17-21, 2019*. ACM, 93–102.

[9] 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. *Rev. Socionetwork Strateg.* 16, 1 (2022), 111–133. doi:10.1007/S12626-022-00105-Z

[10] Guilherme Moraes Rosa et al. 2021. Yes, BM25 is a Strong Baseline for Legal Case Retrieval. *CoRR* abs/2105.05686 (2021). arXiv:2105.05686

[11] Sabine Wehnert et al. 2019. ERST: Leveraging Topic Features for Context-Aware Legal Reference Linking. In *Legal Knowledge and Information Systems - JURIX 2019: The Thirty-second Annual Conference, Madrid, Spain, December 11-13, 2019 (Frontiers in Artificial Intelligence and Applications, Vol. 322)*, Michal Araszkiewicz and Víctor Rodríguez-Doncel (Eds.). IOS Press, 113–122.

[12] Sabine Wehnert et al. 2022. Using Textbook Knowledge for Statute Retrieval and Entailment Classification. In *New Frontiers in Artificial Intelligence - JSAI-isAI 2022 Workshop, JURISIN 2022, and JSAI 2022 International Session, Kyoto, Japan, June 12-17, 2022 (LNCS, Vol. 13859)*, Yasufumi Takama et al. (Eds.). Springer, 125–137.

[13] Masaharu Yoshioka et al. 2022. 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 (LNCS, Vol. 13859)*, Yasufumi Takama et al. (Eds.). Springer, 109–124.

AIIR Lab at COLIEE 2025: Exploring Applications of Large Language Models for Legal Text Retrieval and Entailment

Deiby Wu
University of Southern Maine
Portland, Maine, USA
deiby.wu@maine.edu

Sarah Lawrence
University of Southern Maine
Portland, Maine, USA
sarah.lawrence@maine.edu

Behrooz Mansouri
University of Southern Maine
Portland, Maine, USA
behrooz.mansouri@maine.edu

Abstract

This paper presents the approaches and results of the Artificial Intelligence and Information Retrieval (AIIR) Lab's participation in the 2025 Competition on Legal Information Extraction and Entailment (COLIEE). The AIIR Lab engaged in all four main tasks, leveraging large language models (LLMs) such as Mistral-7B and LLaMA-3. For the Legal Case Retrieval task (Task 1), the team employed LLMs for case summarization, followed by ranking using a fine-tuned bi-encoder model. In the Legal Case Entailment task (Task 2), a fine-tuned cross-encoder model was utilized to assess entailment between case paragraphs. The Statute Law Retrieval task (Task 3) involved augmenting existing training data with LLMs and then fine-tuning a bi-encoder model for search. Finally, for the Legal Textual Entailment task (Task 4), LLMs were employed with three prompting techniques, zero-shot, few-shot, and chain-of-thought (COT), with majority voting applied to determine the final answer. This paper provides the details of the proposed methodologies and experimental results for each task.

CCS Concepts

• **Information systems** → *Specialized information retrieval.*

Keywords

Legal Case Retrieval, Legal Entailment, Legal Language Processing

ACM Reference Format:
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 COLIEE 2025 workshop, June 20, 2025, Chicago, USA.* ACM, New York, NY, USA, 7 pages.

1 Introduction

The application of artificial intelligence, particularly large language models (LLMs), has advanced the field of legal document processing. These models have shown remarkable capabilities in tasks such as legal information retrieval and entailment [15], summarization [5], and question answering [12], enhancing the efficiency and accuracy of legal analyses. To support the development and evaluation of AI models in the legal domain, several specialized datasets have been introduced, including FALQU [13], LexGLUE [3], and Pile of Law [8]. These resources provide the necessary data to train, fine-tune, and evaluate models tailored for legal applications.

The Competition on Legal Information Extraction/Entailment (COLIEE) [18] is dedicated to the automated analysis of legal texts. COLIEE has two retrieval tasks, including Legal Case Retrieval (Task 1) and Statute Law Retrieval (Task 3), and two entailment tasks: Legal Case Entailment (Task 2) and Legal Textual Entailment (Task 4).

Tasks 1 and 2 use the Federal Court of Canada case laws as the corpus. In Task 1, for a given case query, the goal is to find noticed cases in the collection. The query case references a noticed case; however, the references are removed from the query case. Task 2 aims to detect the paragraphs that entail the decision for a given relevant case. Tasks 3 and 4 are based on Japanese Civil Law articles (with English translation available). The goal of Task 3 is to return relevant articles for a query. Articles relevant to a query are those that can answer the query (with Yes/No) entailed from the article. Finally, Task 4 focuses on question answering; given a legal bar question, and a Civil Law article, the task explores if the article can entail the query.

The Artificial Intelligence and Information Retrieval (AIIR) Lab from the University of Southern Maine participated for the first time in the COLIEE, proposing different systems for all four tasks. Our runs rely on two large language models (LLMs): Mistral-7B-Instruct-V0.2 [9] (hereafter referred to as Mistral) and LLaMA-3-8B-Instruct [6] (hereafter referred to as LLaMA). These LLMs are used for case summarization, ranking, and classification for entailment tasks. We also used neural information retrieval models for retrieval tasks.

Our most effective systems for each task are as follows. For Task 1, we used Mistral for case summarization, followed by a fine-tuned Sentence-BERT bi-encoder[19] for ranking. In Task 2, our fine-tuned cross-encoder model provided the best effectiveness. For Task 3, we first augmented the existing training data with Mistral and then fine-tuned a bi-encoder model for search. Finally, for Task 4, we used Mistral with three prompting techniques: zero-shot, few-shot, and chain-of-thoughts (COT), and then applied majority voting to get the final answer.

In this paper, for each task, we first review its objectives, then present our proposed models, and finally discuss the experimental results.

2 Task 1: Legal Case Retrieval

The Legal Case Retrieval task aims to evaluate the effectiveness of legal document retrieval systems in identifying relevant case laws that support a given (unseen) query case. The system must retrieve "noticed cases", which are those referenced by the query case, though explicit references are redacted to assess retrieval

COLIEE 2025, Chicago, USA

Table 1: Average number of words in Task 1 collection cases at each cleaning step.

Data	Base	THUIR	YAKE!	Mistral	LLaMA
Train	4654.07	3943.90	3547.54	347.50	196.80
Test	5125.14	4466.06	4027.06	358.77	288.47

accuracy. The corpus consists of Federal Court of Canada case laws, with training data providing query cases and their corresponding noticed cases, while test data includes only query cases without labels. The evaluation metrics for Task 1 include precision, recall, and F-measure. This task uses micro-averaging, where evaluation metrics are calculated collectively over all queries.

2.1 Proposed Models

In our proposed models, both query and candidate cases were first passed through a similar preprocessing pipeline, shown in Figure 1. We used the THUIR team approach [11] at COLIEE 2023 for the initial case cleaning. This cleaning includes steps such as removing extra spaces, handling punctuation, and removing French text. Despite the cleaning, the legal cases are longer than the maximum sequence length that neural information retrieval and large language models can support.

To overcome this issue, we relied on YAKE! keyword extractor [2], and for each legal case, we kept the paragraphs that contained the top-3 important keywords, based on YAKE! scores. We then used two large language models, Mistral and LLaMA, to summarize each case. Both LLMs were used with a temperature of 0.1 and a maximum sequence length of 2048 for generation. For Mistral, we used the prompt "Provide a concise summary of this legal case." and passed the legal case to be summarized by LLM. For LLaMA, we used a similar prompt as Mistral but also considered a system message (role) as:

> You are a legal expert that summarizes long legal cases into a precise paragraph while keeping the main content.

As shown in Table 1, the initial legal cases (Base) contain a high number of words on average, with the test cases being slightly longer than the training cases. The THUIR cleaning step reduces the word count by removing unnecessary elements, leading to a moderate reduction in length. The YAKE! filtering step further reduces the size by retaining only the most relevant paragraphs based on keyword importance, ensuring that critical content is preserved while eliminating less relevant sections. The most noticeable reductions occur after summarization with Mistral and LLaMA. Mistral produces summaries averaging around 347 words for training cases and 359 words for test cases, indicating a strong compression while maintaining consistency between datasets. LLaMA generates even shorter summaries, reducing cases to an average of 197 words for training and 288 words for test cases.

After summarizing each case, we proceeded to fine-tune a bi-encoder retrieval model using the pre-trained 'all-mpnet-base-v2' [19] model. The combination of summarization and transformer-based models has been previously explored in legal case retrieval to address sequence length limitations [1, 21]. We utilized the provided training data, which comprises 1,678 queries with an average of 4.1 associated cases. The dataset was split into a 90:10 ratio for training and validation sets. For fine-tuning, we used the Multiple Negatives Ranking Loss (MNRL) [7], which is well-suited for scenarios where only positive samples are provided. The model was fine-tuned for 10 epochs using a batch size of 16, and we selected the best-performing model based on the Mean Average Precision at rank 100 (MAP@100) on the validation set.

Based on these methods, we devised three experimental runs as follows:

(1) **AIIRmpMist5**: In this run, we use the case summaries generated by Mistral to fine-tune a bi-encoder model. For each case query, the model retrieves the top-5 most relevant results.
(2) **AIIRmpMist3**: Using the same approach as above, this run retrieves only the top-3 results, prioritizing precision. This threshold was chosen based on the average number of noticed cases in the training set.
(3) **AIIRcombMNZ**: While the previous runs relied exclusively on Mistral's summaries, we also generate summaries using LLaMA and fine-tune a separate bi-encoder model with them. Since our experiments on the training data showed that LLaMA's summaries were less effective for legal cases, we combine the outputs from both the LLaMA- and Mistral-based bi-encoder models using the CombMNZ fusion [10], and then selected the top-3 results. CombMNZ multiplies the number of ranks where the document occurs by the sum of the scores obtained with the two systems.

Table 2: Results of AIIR Lab Runs for Task 1 on the COLIEE 2025 Test Set (400 Queries).

Model	F-Measure	Precision	Recall
AIIRmpMist5	**0.2171**	0.2040	**0.2319**
AIIRmpMist3	0.1872	0.2308	0.1575
AIIRcombMNZ	0.1879	**0.2317**	0.1580

2.2 Experimental Results

Table 2 shows our results on the COLIEE 2025 test set for Task 1. 400 case queries are considered for this year's competition, with an average of 4.3975 noticed cases per query. As can be seen from this table, including LLaMA-based results led to a slight improvement in the precision, and with Mistral summaries, the recall gain at cut@5 was higher, leading to the *AIIRmpMist5* run being the most effective. For our runs at cut@3 (*AIIRmpMist3* and *AIIRcombMNZ*), the precision was significantly higher than the run with cut@5 (*AIIRmpMist5*), using the paired Student's t-test ($p < 0.05$). In contrast, with cut@5, the recall was significantly higher than the other two runs. *AIIRmpMist5* F-measure was also significantly higher than the other runs.

Looking at the results with Mistral fine-tuned data, at cut@5, the precision for 159 out of 400 query cases drops compared to cut@3. For 2 query cases, one regarding Harrington v. Microsoft (with ID 063752) and the other Canadian Union of Postal Workers v. Canada Post (with ID 025612), the precision dropped from 1 to 0.6 when the

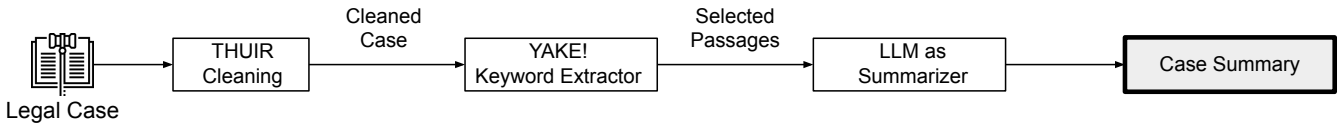


Figure 1: Proposed approach for summarizing legal cases in task 1. After processing each case with the THUIR approach, passages with top-3 keywords are passed to an LLM to summarize the case.

cut was increased from 3 to 5. For case 025612, the retrieval model included two additional cases in the top-5 results due to strong lexical matches with the query terms such as contempt, arbitration, and other procedural phrases, leading the model to identify them as potentially relevant. However, the legal issues in these cases, involving copyright injunctions and maritime contract disputes, differ from the labor and administrative arbitration focus of the query, making them irrelevant despite the initial semantic match. On the other hand, with a cut of 5, the recall increases for 117 queries. In particular, for 7 of these queries, for which there was only one relevant case, the recall increases from 0 to 1 when the depth of retrieved instances is set to 5.

Our models failed to retrieve any relevant cases for 37% of the test queries. One major issue is that the bi-encoder model discards the details of cases. For instance, for case 001083, which is related to the Refugee Protection Division, our fine-tuned bi-encoder model retrieved cases containing common lexical and semantic cues, such as references to the Refugee Appeal Division, judicial review, and subsection 110(4) of the IRPA that matched the query. However, the query concerns the applicant's refugee claim based on sexual orientation and credibility issues, and the model retrieved cases focusing on evidentiary disputes over a high school diploma. Similarly, in a query involving document production under s. 39 of the Canada Evidence Act (with ID 099982), the model retrieved cases that address constitutional challenges to ex parte representations and disputes over personal information disclosures. Although these cases share similarities in statutory references and document production terms, they differ in legal focus from the original query, which deals specifically with document certification procedures and confidentiality issues in a multi-agency litigation context.

3 Task 2: Legal Case Entailment

The Legal Case Entailment task aims to predict the decision of a new case by identifying supporting paragraphs from relevant past cases. Given a query case decision and a noticed case, the system must determine which paragraph in the noticed case entails the decision. Training data consists of triples: a query, a noticed case, and the paragraph number that supports the decision. In the test phase, only queries and noticed cases are provided, without paragraph labels. The process must be fully automated, with no human intervention or system modifications. Similar to Task 1, the evaluation metrics for Task 2 include precision, recall, and F-measure, with equal weighting for precision and recall. Precision measures the proportion of correctly retrieved paragraphs among all retrieved paragraphs, while recall calculates the proportion of correctly retrieved paragraphs among all relevant ones.

3.1 Proposed Models

For Task 2, we proposed three systems. For each query, the top-ranked paragraph according to the model's score is selected. The second-ranked paragraph is included in the result if its score exceeds a threshold. For our first two runs, this threshold is set to 0.5, and for the last run, it is set to less than 10% difference compared to the top-ranked paragraph score. Here are our runs for this task:

(1) **crossAIIRLab.** This approach uses a cross-encoder model, "ms-marco-MiniLM-L-6-v2", with the MiniLM [22] architecture trained for passage re-ranking [16] on the MS MARCO dataset. We fine-tuned this model on Task 2 training data to optimize its ability to rank legal paragraphs according to textual entailment. We used 675 queries from the original 825 queries and their associated paragraphs to fine-tune the model for 30 epochs with a batch size of 8 and a learning rate of 2e-5. The remaining 150 queries were used for testing the model's performance. The model was optimized using binary cross-entropy.

(2) **mT5AIIRLab.** Our second approach fine-tunes a MonoT5 [17] model, "monot5-base-msmarco", which adapts a pre-trained T5 encoder-decoder for passage re-ranking and is also trained on the MS MARCO dataset. For each training example, the input was given as "Query:[query text] Document: [paragraph text] Relevant: ", and the target was either *true* or *false* depending on the relevance judgment given by the model. Similar to the previous model, we fine-tuned the model using 675 queries from the 825 queries for 3 epochs with a batch size of 8 and an AdamW optimizer with loss computed using sequence-to-sequence cross-entropy. At inference, the model computes log-probabilities for both *true* and *false* outputs and uses the difference as the score.

(3) **mergeAIIRLab.** As an ensemble model, this model combines re-ranking scores from four models: BM25 [20], a pre-trained bi-encoder ("all-mpnet-base-v2") [19], and our two previous runs. For each query, scores from all models are min-max normalized and merged using a weighted average with weights set with grid-search on the train set. The weights for BM25 and bi-encoder are set to 0.1, and 0.4 for previous runs.

3.2 Experimental Results

Our results for Task 2 are presented in Table 3. The test set contains 100 queries with an average of 1.81 relevant paragraphs per query. Among all the models evaluated, the fine-tuned cross-encoder (crossAIIRLab) achieved the best performance across all the official metrics.

The *crossAIIRLab* model captured the context of legal reasoning even when different terminologies were used. For example, one

Table 3: Results of AIIR Lab Runs for Task 2 on the COLIEE 2025 Test Set (100 Queries).

Model	F-Measure	Precision	Recall
crossAIIRLab	**0.2368**	**0.2927**	**0.1989**
mergeAIIRLab	0.2229	0.2632	0.1934
mt5AIIRLab	0.1930	0.2050	0.1823

query discussing judicial impartiality and racial dynamics was correctly matched to a paragraph from *R. v. R.D.S.* that examined the reasonable person standard in the context of racial bias. Another query concerning translation errors and credibility was aligned with a paragraph addressing similar Charter protections, even sharing phrases like "differences in nuance between what is said in one language and its translation into another." These examples show the model's ability to not only focus on similarity but also to detect deeper semantic entailment in the legal field.

However, there were instances where the fine-tuned cross-encoder identified topic overlap between the query and a paragraph without achieving logical alignment. For example, a query about the likelihood of confusion in trademark law was matched with a paragraph discussing trademark ownership and first use; reflecting a legal relation but lacking an entailment element. In another case, a query comparing two different standards of judicial review was matched to a paragraph that opposed that very approach. These examples suggest that while our fine-tuned cross-encoder is capable of semantic matching, it can be misled by similar legal vocabulary when the contextual alignment is insufficient.

Similarly, the *mt5AIIRLab* model performs well when the paragraph supports the query's reasoning. For instance, a query asserting the importance of centralized residency in Canada was matched with a paragraph quoting the established legal test for residency, including the phrase "centralizes his ordinary mode of living." In another example, a query regarding the evidentiary weight of interview notes was paired with a paragraph explaining that such notes are insufficient without supporting affidavits. However, the MonoT5 model sometimes relied too heavily on surface-level similarity. In one case, a query on judicial impartiality and racial dynamics (the same query used for the cross-encoder analysis) was matched to a paragraph stating that there was no apprehension of bias, failing to address the query's underlying context and reasoning. In another instance, a query justifying the denial of deferral in light of a pending humanitarian application was paired with a paragraph that argued in favor of deferral under similar circumstances, illustrating the model's difficulty in grasping the logic of the argument.

Overall, both models show strong performance when semantic alignment is unambiguous but struggle when queries and paragraphs present opposing positions using similar terminology. As shown in Table 3, the merged model performed slightly below the cross-encoder, suggesting that weaker components such as BM25 and the bi-encoder may have negatively impacted the overall score by emphasizing lexical or superficial semantic similarity over true entailment. This finding underscores the importance of carefully controlling the contribution of each model in ensembles for entailment tasks. Future improvements include refining ensemble weighting strategies, removing models that detract from overall performance, and training on cases that emphasize differences in legal reasoning.

4 Task 3: Statute Law Retrieval

This task aims to evaluate the effectiveness and reliability of legal document retrieval systems by assessing their performance in retrieving relevant Civil Law articles based on previously unseen queries. The system operates on a static set of Japanese Civil Law articles, provided in both Japanese and English translation, and must automatically identify all relevant articles that contribute to answering a query. An article is considered relevant if its meaning entails a yes/no response to the query.

The evaluation metrics for this task include precision, recall, and F2-measure, with an emphasis on recall since the retrieval process serves as a pre-selection step for entailment. Precision measures the proportion of correctly retrieved articles among all retrieved articles, while recall calculates the proportion of correctly retrieved articles among all relevant articles. The F2-measure prioritizes recall by weighting it more heavily than precision. Additionally, Mean Average Precision (MAP) is used to analyze system performance. The final evaluation score is computed using macro-averaging, where the metric is calculated for each query and then averaged across all queries. For this task, there were 1,206 samples provided for training purposes, and the search collection contained 776 articles with an average of 71.96 words in each case. Compared to Task 1, the cases in this task are cleaner and do not need pre-processing steps.

4.1 Proposed Models

For Task 3, we developed three retrieval models, and in each run, we considered the top-3 results. To increase the number of training samples, we use Mistral for data augmentation with 10 different prompts to rewrite the original sample queries in the training set. Each prompt is designed to provide a different query and increase both the number and diversity of training data. If (Q_i, A_j) are in the training data as positive sample, where Q_i is the original query, and A_j is a relevant article, we generate $\{Q_{i0}, Q_{i1}, ..., Q_{i9}\}$ and consider positive samples as $\{(Q_i, A_j), (Q_{i0}, A_j), (Q_{i1}, A_j), ..., (Q_{i9}, A_j)\}$ for fine-tuning. Our prompts include the following sample that aims for summarization, simplification, and rephrasing:

- Reword the legal query into a **concise summary** that highlights the essential legal elements, using precise legal language.
- Transform the query into a **simplified** yet formally worded version that maintains all critical legal details.
- **Paraphrase** the following legal query using alternative legal terms and sentence structures, while preserving the original legal meaning and nuance.

Table 4 shows how a training query (ID H23-1-4) is rewritten using three augmentation prompts. In the Paraphrased variant, phrases are reworded; for instance, "cannot refused execution of his or her obligation" becomes "cannot withhold performance," while the timing and rescission conditions are retained, showing structural and lexical variation without loss of meaning. The Summarized form highlights the core legal elements: "contract formation through another party's fraud," "the defrauded party cannot refuse

Table 4: Rewriting of training query (H23-1-4) via three augmentation prompts: paraphrase, concise summary, and formal simplification for Task 3 data augmentation.

Query	In cases where the person who makes the manifestation forms a contract based on the fraud of the other person, even if the obligation of the other person is not yet executed based on the contract, the person who makes the manifestation cannot refused execution of his or her obligation for reason of fraud as long as the manifestation of intention is not rescinded.
Paraphrased	When a contract is formed through another party's deceit, prior to their contractual obligation's execution, the deceived party cannot withhold performance based on the fraud. This rule holds as long as the intention to be bound has not been revoked.
Summarized	In instances of contract formation through another party's fraud, the defrauded party cannot refuse performance of their obligations prior to contract execution, provided their manifestation of intent has not been rescinded.
Simplified	In instances where a contract is formed through the deceit of one party, the opposing party's obligation to perform under the contract cannot be refused execution based on the fraudulent inducement, unless and until the manifestation of intent to be bound has been effectively rescinded.

performance prior to contract execution," and the binding "manifestation of intent." Finally, the Simplified version preserves critical details but refines syntax and phrasing, such as rephrasing the obligation's status ("obligation... cannot be refused execution"), to improve readability and flow while retaining formality. We leave further exploration of rewriting and augmentation of legal queries for future work.

After data augmentation, we use the original training samples, along with the augmented data to fine-tune a bi-encoder model, "all-mpnet-v2" for 10 epochs. We split the data in a 90:10 ratio for the training and validation sets. The best model on the validation set is selected based on the highest Spearman correlation score by assessing the similarity of the generated embeddings by comparing them, using cosine similarity, Euclidean, and Manhattan distances, to the gold standard labels. This forms our first run, **mpnetAIIRLab**.

In our second run, **mistAIIRLab**, we re-ranked the top-10 results from the previous run (*mpnetAIIRLab*) for each query using a pairwise approach with Mistral. For this, we used the following prompt and passed the query with two candidates for re-ranking:

> Being a ranking model, your task is to decide for a given legal query in the context of Japanese civil law, which of the two articles is more relevant.

Finally, our third run, **NVAIIRLab**, uses another bi-encoder architecture, using NVIDIA model's 'NV-Embed-v2' [4]. For this approach, we used the pre-trained model as a zero-shot baseline without any further fine-tuning.

4.2 Experimental Results

Table 5 presents our results for Task 3 on the COLIEE 2025 test queries. Among our submissions, the *mpnetAIIRLab* run achieved the best performance across all evaluation metrics. This higher performance was statistically significant (Student's t-test, $p < 0.05$) over *mistAIIRLab* for the MAP metric and over *NVAIIRLab* for Precision.

Looking at the results, the *mpnetAIIRLab* model has a recall of 1 for 82.4% of queries, while it fails to retrieve any relevant documents for 8.1% of queries. However, for 78.3% of the queries, only one of the retrieved articles out of three is considered as relevant, leading to a precision of 0.33 for those instances. On average, each query in the test set has 1.26 relevant articles, with only one query having three relevant articles and the remainder having fewer. Therefore, our approach of always considering top-3 results should be further improved by learning the correct cutoff for the top-k retrieved results.

Table 5: Results of AIIR Lab Runs for Task 3 on the COLIEE 2025 Test Set (74 Queries).

Model	F2	Precision	Recall	MAP
mpnetAIIRLab	**0.6246**	**0.3333**	**0.8291**	**0.7931**
mistAIIRLab	0.5672	0.3034	0.7521	0.6867
NVAIIRLab	0.5554	0.2863	0.7479	0.7412

Exploring *mpnetAIIRLab* retrieval results, consider a topic such as R06-05-O, which is related to a debtor who causes collateral to be lost before the assigned time. Our model, which uses topic similarity, finds two relevant documents; however, it also retrieves a non-relevant article (with ID 706) that addresses an entirely different scenario of early performance of an obligation, making it irrelevant to the query regarding loss of collateral. When re-ranked with Mistral, this irrelevant article was dropped from the result. However, Mistral includes another article (with ID 135) that focuses on the timing for the performance or expiration of a juridical act, rather than addressing the consequences of the debtor's actions. This article describes the general principle of a "time of commencement" for a legal act or obligation.

While *mpnetAIIRLab* was on average more effective than our other two runs, for topics such as R06-15-I concerning ineffective pledge of a claim under a no-pledge clause, our other two runs were able to find the relevant article (with ID 466) as the top result. The bi-encoder approach retrieves statutes by measuring surface-level similarity between the query and statute text embeddings, so it often favors passages that share common keywords ("pledge," "property," "possession") even when those statutes aren't about prohibitions on pledging or their legal effect. In contrast, the language model–based retrieval captures deeper contextual and logical relationships: it recognizes that prohibiting assignment of a claim and the consequences for a third party with knowledge of that prohibition are directly parallel to prohibiting a pledge of the claim.

Overall, our experimental results show that while our fine-tuned encoder model primarily considers topical relevance, there is still room for improvement by better leveraging LLMs' reasoning capabilities. We also need a more effective mechanism for selecting relevant documents, as using a constant top-k for all queries was not effective.

5 Task 4: Legal Textual Entailment

The Legal Textual Entailment task aims to develop yes/no question-answering systems for legal queries by determining whether relevant Civil Law articles entail the query. Given a legal bar exam question, the system evaluates whether the retrieved content entails the question. The training data consists of query-article-answer triples, while the test data includes only queries and relevant articles without answer labels. The evaluation measure is accuracy, based on whether the yes/no question is correctly answered. This task includes 1,206 training and 74 test queries.

5.1 Proposed Models

For this task, we submitted two runs using LLaMA and Mistral LLMs with a similar approach. With each model, we considered three prompting techniques to decide if a legal article entails the legal question:

(1) **Zero-shot:** The LLM directly predicts Yes/No without any examples, based solely on the candidate legal article and the query.
(2) **Few-shot:** The model is provided with one positive (answer: "Yes") and one negative (answer: "No") example before answering the test input.
(3) **Zero-shot COT:** Similar to the zero-shot approach but incorporating "Let's think step-by-step" to encourage reasoning before answering.

For LLaMA, we included the following system prompt to guide the model's responses:

> You are an expert Japanese lawyer who will decide if a given legal query can be entailed by a given legal article. You will answer with Yes or No. Then, you will provide a brief explanation.

To enhance robustness, we aggregated the model outputs using majority voting, inspired by the approach proposed by Nguyen et al. [14]. Our models were named **AIIRLLaMA** (based on LLaMA) and **AIIRMistral** (based on Mistral).

5.2 Experimental Results

Table 6 shows our results on the COLIEE 2025 test set for Task 4. Our results indicate that *AIIRLLaMA* outperforms *AIIRMistral*, achieving an accuracy of 60.81% compared to 56.76%. This suggests that LLaMA's entailment reasoning capabilities were more effective in this legal domain task. For 30 out of 74 queries, almost 90% or more participating runs had the correct answers, which may indicate these queries were less challenging. Among these 30 queries, both models had only two wrong predictions on the same instances. On the other hand, for 7 queries, less than 10 participating runs were able to predict entailment correctly. While Mistral failed to predict any of these instances correctly, LLaMA predictions for two queries (with Ids R06-20-O and R06-22-A) were correct. LLaMA successfully identified the contextual cues linking the shared obligations and reimbursement rights among guarantors, whereas Mistral failed to do so.

Table 6: Accuracy of AIIR Lab Runs for Task 4 on COLIEE 2025 and previous years Test Sets.

Model	R06 (2025)	R02	R01	H30
AIIRLLaMA	**60.81**	**65.43**	36.04	51.43
AIIRMistral	56.76	64.20	**37.84**	**61.43**

While our proposed models used majority voting, the effectiveness of each prompting technique varied. With LLaMA, both zero-shot and few-shot prompting resulted in an accuracy of 58.11%, while chain-of-thought prompting increased the accuracy to 63.51%. The pattern observed for Mistral differed; with zero-shot prompting, the accuracy was 56.76%, which improved to 59.46% using chain-of-thought prompting. Notably, when few-shot prompting was used, the accuracy further increased to 68.92%. These results suggest that the chosen prompting technique influences the performance of large language models in tasks such as legal entailment. Moreover, exploring alternative ensembling techniques could potentially enhance our majority voting approach. Finally, the varying accuracies observed across different query sets from previous labs indicate that the two language models exhibit distinct strengths depending on the query type. This indicates the need for further investigation into query characteristics and more refined strategies for selecting and fine-tuning LLMs for legal entailment.

6 Conclusion

In conclusion, this paper presents the AIIR Lab's exploration of large language models (LLMs) to enhance legal information retrieval and entailment tasks in the 2025 Competition on Legal Information Extraction and Entailment (COLIEE). Our participation spanned four distinct tasks, each tackling unique challenges and opportunities within legal text processing. We found that models such as Mistral and LLaMA could be effectively leveraged for case summarization and relevance ranking. However, accurately capturing the nuanced context of legal language remains a challenge. As this marks our first participation in COLIEE, several avenues remain unexplored. Future work will focus on optimizing the integration of neural rankers and classification and ranking approaches with LLMs, exploring ensembling techniques, and refining strategies for better semantic alignment.

References

[1] Arian Askari, Suzan Verberne, O Alonso, S Marchesin, M Najork, and G Silvello. 2021. Combining Lexical and Neural Retrieval with Longformer-based Summarization for Effective Case Law Retrieval.. In *DESIRES*. 162–170.
[2] Ricardo Campos, Vítor Mangaravite, Arian Pasquali, Alípio Jorge, Célia Nunes, and Adam Jatowt. 2020. YAKE! Keyword extraction from single documents using multiple local features. *Information Sciences* 509 (2020), 257–289. doi:10.1016/j.ins.2019.09.013
[3] 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)*, Smaranda Muresan, Preslav Nakov, and Aline Villavicencio

(Eds.). Association for Computational Linguistics, Dublin, Ireland, 4310–4330. doi:10.18653/v1/2022.acl-long.297

[4] Gabriel de Souza P. Moreira, Radek Osmulski, Mengyao Xu, Ronay Ak, Benedikt Schifferer, and Even Oldridge. 2025. NV-Retriever: Improving text embedding models with effective hard-negative mining. arXiv:2407.15831 [cs.IR] https://arxiv.org/abs/2407.15831

[5] Aniket Deroy, Kripabandhu Ghosh, and Saptarshi Ghosh. 2024. Applicability of large language models and generative models for legal case judgement summarization. *Artificial Intelligence and Law* (2024), 1–44.

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

[7] Matthew Henderson, Rami Al-Rfou, Brian Strope, Yun hsuan Sung, Laszlo Lukacs, Ruiqi Guo, Sanjiv Kumar, Balint Miklos, and Ray Kurzweil. 2017. Efficient Natural Language Response Suggestion for Smart Reply. arXiv:1705.00652 [cs.CL] https://arxiv.org/abs/1705.00652

[8] Peter Henderson, Mark Krass, Lucia Zheng, Neel Guha, Christopher D Manning, Dan Jurafsky, and Daniel Ho. 2022. Pile of law: Learning responsible data filtering from the law and a 256gb open-source legal dataset. *Advances in Neural Information Processing Systems* 35 (2022), 29217–29234.

[9] Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7B. arXiv:2310.06825 [cs.CL] https://arxiv.org/abs/2310.06825

[10] Joon Ho Lee. 1997. Analyses of multiple evidence combination. In *Proceedings of the 20th annual international ACM SIGIR conference on Research and development in information retrieval*. 267–276.

[11] 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:2305.06817 [cs.CL] https://arxiv.org/abs/2305.06817

[12] 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.

[13] Behrooz Mansouri and Ricardo Campos. 2023. Falqu: Finding answers to legal questions. *arXiv preprint arXiv:2304.05611* (2023).

[14] 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 *New Frontiers in Artificial Intelligence*, Toyotaro Suzumura and Mayumi Bono (Eds.). Springer Nature Singapore, Singapore, 167–182.

[15] 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 *JSAI International Symposium on Artificial Intelligence*. Springer, 125–139.

[16] Rodrigo Nogueira and Kyunghyun Cho. 2020. Passage Re-ranking with BERT. arXiv:1901.04085 [cs.IR] https://arxiv.org/abs/1901.04085

[17] Rodrigo Nogueira, Zhiying Jiang, and Jimmy Lin. 2020. Document Ranking with a Pretrained Sequence-to-Sequence Model. arXiv:2003.06713 [cs.IR] https://arxiv.org/abs/2003.06713

[18] 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 (01 Apr 2022), 111–133. doi:10.1007/s12626-022-00105-z

[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)*, Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (Eds.). Association for Computational Linguistics, Hong Kong, China, 3982–3992. doi:10.18653/v1/D19-1410

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

[21] Julien Rossi and Evangelos Kanoulas. 2019. Legal search in case law and statute law. In *Legal Knowledge and Information Systems*. IOS Press, 83–92.

[22] Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. 2020. MiniLM: Deep Self-Attention Distillation for Task-Agnostic Compression of Pre-Trained Transformers. arXiv:2002.10957 [cs.CL] https://arxiv.org/abs/2002.10957

Author Index

Keyword Index