Overview of Benchmark Datasets and Methods for the Legal Information Extraction/Entailment Competition (COLIEE) 2024

Randy Goebel[1], Yoshinobu Kano[2], Mi-Young Kim[3](✉), Juliano Rabelo[1], Ken Satoh[4], and Masaharu Yoshioka[5]

[1] Department of Computing Science and Alberta Machine Intelligence Institute, University of Alberta, Edmonton, AB, Canada
{rgoebel,rabelo}@ualberta.ca

[2] Faculty of Informatics, Shizuoka University, Hamamatsu, Shizuoka, Japan
kano@inf.shizuoka.ac.jp

[3] Department of Science, Augustana Faculty, University of Alberta, Camrose, AB, Canada
miyoung2@ualberta.ca

[4] National Institute of Informatics, Chiyoda-ku, Tokyo, Japan
ksatoh@nii.ac.jp

[5] Faculty of Information Science and Technology, Hokkaido University, Sapporo-shi, Hokkaido, Japan
yoshioka@ist.hokudai.ac.jp

Abstract. We summarize the 11th Competition on Legal Information Extraction and Entailment (COLIEE 2024). In this eleventh edition, the competition included four tasks on case law and statute law. The case law component includes an information retrieval task (Task 1), and the confirmation of an entailment relation between an existing case and a selected unseen case (Task 2). The statute law component includes an information retrieval task (Task 3), and an entailment/question-answering task based on retrieved civil code statutes (Task 4). Participation was open to any group based on any approach. Ten different teams participated in the case law competition tasks, most of them in more than one task. We received results from 10 teams for Task 1 (26 runs) and 6 teams for Task 2 (18 runs). On the statute law task, there were 12 different teams participating, most in more than one task. 8 teams submitted a total of 20 runs for Task 3, and 8 teams submitted a total of 23 runs for Task 4. We describe the variety of approaches, our official evaluation, and analysis of our data and submission results.

Keywords: COLIEE2024 · legal information retrieval · legal information entailment

1 Introduction

The objective of the Competition on Legal Information Extraction/Entailment (COLIEE) is to encourage the development of state of the art for information

T. Suzumura and M. Bono (Eds.): JSAI-isAI 2024, LNAI 14741, pp. 109–124, 2024.
https://doi.org/10.1007/978-981-97-3076-6_8

retrieval and entailment methods using legal texts. It is usually co-located with JURISIN, the Juris-Informatics workshop series, which was created to promote community discussion on both fundamental and practical issues on legal information processing, with the intention to embrace many disciplines: these include law, social sciences, information processing, logic and philosophy, and the existing conventional “AI and law” area. In alternate years, COLIEE is organized as a workshop of the International Conference on AI and Law (ICAIL), which was the case in 2017, 2019, 2021, and 2023. Until 2017, COLIEE consisted of two tasks: information retrieval (IR) and entailment using Japanese Statute Law (civil law). From COLIEE 2018, we introduced a new and challenging case law IR and entailment tasks based on Canadian case law.

Task 1 is a legal case retrieval task, and it involves reading a query case and extracting supporting cases from the provided case law corpus, hypothesized to be relevant to the query case. Task 2 is the legal case entailment task, which involves the identification of relevant paragraphs or paragraphs from existing cases, which can be confirmed to entail a given fragment of a new case. Tasks 3 and 4 are statute law tasks that use questions from the Japanese Bar exam to judge whether the given statement is true or not. Task 3 is an information retrieval task that identifies relevant articles for the legal entailment (Task 4). Finally, Task 4 is a legal entailment task that judges whether the given statement is true or not. In contrast to COLIEE 2023, COLIEE 2024 introduced 400 new query cases for Task 1 and 100 for Task 2. Furthermore, for the test data of Task 3 and Task 4 in COLIEE 2024, 109 new questions sourced from the 2023 bar exam were used.

The rest of our paper is organized as follows: Sects. 2, 3, 4, and 5 describe each task, presenting their definitions, datasets, list of approaches submitted by the participants, and results attained. Section 6 presents some final remarks.

2 Task 1 - Case Law Retrieval

2.1 Task Definition

This task consists of finding which cases, amongst a set of provided candidate cases, should be “noticed” with respect to a given query case[1]. More formally, given a query case q and a set of candidate cases $C = \{c_1, c_2, ..., c_n\}$, the task is to find the supporting cases $S = \{s_1, s_2, ..., s_n \mid s_i \in C \wedge \mathit{noticed}(s_i, q)\}$ where $\mathit{noticed}(s_i, q)$ denotes a relationship which is true when $s_i \in S$ is a noticed case with respect to q.

2.2 Case Law Dataset

The dataset consists of a total of 7,350 case law files. That dataset contains a labelled training set of 5,616 cases, of which 1,278 are query cases. On average,

[1] “Notice” is a legal technical term that denotes a legal case description that is considered to be relevant to a query case.

the training data includes approximately 4.16 noticed cases per query case, which are to be identified among the 5,616 cases. To prevent competitors from merely using existing embedded conventional legal citations in historical cases to identify cited cases, citations are suppressed from all candidate cases and replaced by a "FRAGMENT_SUPPRESSED" tag indicating that a fragment containing a citation was removed from the case contents.

The test set consists of a total of 1,734 cases, with 400 query cases and a total of 1,562 true noticed cases (an average of 3.90 noticed cases per query case). Initially, the golden labels for that test set are not provided to competitors.

2.3 Approaches

We received 26 submissions from 10 different teams for Task 1. In this section, we present an overview of the approaches taken by the teams which submitted papers describing their methods.

- **UMNLP** [2] **(3 runs)** developed a pairwise similarity ranking framework. The authors train a feed-forward neural network to perform a binary classification task based on several features from each query-candidate case pair. Those features include the extraction and similarity matching for a novel feature which the authors call a "proposition" (a short summary of the basis upon which a noticed case has been cited), as well as the name of the judge deciding the case, verbatim quotations from the text, and several other novel features.
- **JNLP** [5] **(3 runs)** proposes a three-phase approach: the first stage performs retrieval after splitting the query document into paragraphs and using a BM25 model with top-k cutout to retrieve candidate documents. Phase two is a re-ranking stage. The last stage is where prediction actually happens: after the re-ranking stage, for each query document, the authors select the top-k candidate documents from the re-ranked list as prediction with k selected, using grid-search on the validation set. They also developed an ensemble strategy by concatenating the prediction results of the re-rankers before selecting the top-k to boost the recall metric of the system.
- **BM24 (1 run)** the authors organize each case into segments summarized by gpt-3.5. Among them, one segment is selected to represent the case. An embedding of that segment is stored in FAISS. A segment of the query case is used to query the FAISS vector store to retrieve similar cases. AnglE is used as the sentence embedding model, trained from SeanLee97/angle-llama-7b-nli-20231027 (from the HuggingFace repository) on the Task 1 training set pre-processed in the same way.
- **CAPTAIN** [6] **(3 runs)** performs some heuristic pre-processing steps, then uses TF-IDF and BM25 to extract keywords and retrieve relevant documents. The team then applies LLMs to summarize the decisions and perform fine-tuning of a retrieval model based on such summaries.
- **NOWJ** [7] **(3 runs)** developed an approach based on a combination of BM25 and a pre-trained Longformer. After an initial pre-processing step, BM25 is

used to calculate the similarity between each pair of query case and candidate case. The result is used as a pre-ranking input to the LongFormer model. Scores from BM25 and LongFormer are then combined, with parameters being defined after a grid search is conducted.

- **MIG (1 run)** chose to offer an informative baseline for Task 1 that does not apply any LLMs. The authors vectorized the cases with a tool on BERT-base and BERT-large. After vectorizing the cases, they compute the cosine similarity between the candidate cases and the given new case using FAISS. Then, for a new case, the authors ranked the candidate cases by their cosine similarity with the new case, and chose 20 candidates that were most similar to the new case. Then the difference between the cosine similarity between the i-th most similar case and the $(i+1)$-th most similar case (d_i) is calculated, and the first i cases are recommended if $d_i > 2d_1$.
- **UBCS (3 runs)** applied TF-IDF to rank cases varying how the model is used. Their first approach is a baseline, with vanilla TF-IDF weighting model being used to retrieve and rank noticed cases for each given query case. The second approach applies summarization only on the query cases before using TF-IDF for retrieval. The third approach applies summarization for both the query and candidate cases.
- **TQM** [4] **(3 runs)** used lexical matching and dense vector retrieval to generate features (plus some simple features such as case length) that were submitted to a learning to rank method. The authors also applied pre and post processing to avoid irrelevant information. Their method not only applies all of those techniques, but aims at a deeper understanding of the case trying to capture the main facts described in the case.

2.4 Results and Discussion

Table 1 shows the results of all submissions received for Task 1 for COLIEE 2024. A total of 26 submissions from 10 different teams were evaluated. Similar to what happened in recent COLIEE editions, the f1-scores are generally low, which reflects the fact that the task is now more challenging than its previous formulation[2]. However, this year we witnessed a relevant increase of almost 50% in the performance of the winning team, from an f1-score of 0.30 in 2023 to 0.44 in the current edition.

In this edition of COLIEE, we improved our sampling method to provide test data which has similar properties/data distribution to the training data, something we noticed could be improved from the next competition. We have also improved case duplication identification, although some duplicate cases were still present. We intend to further improve our method of duplicate identification in the next competition.

Most of the participating teams applied some form of traditional IR technique such as BM25, transformer based methods such as BERT or more recent

[2] For a description of the previous Task 1 formulation, please see the COLIEE 2020 https://sites.ualberta.ca/~rabelo/COLIEE2020/.

Table 1. Task 1 results

Team	F1	Precision	Recall	Team	F1	Precision	Recall
TQM	0.4432	0.5057	0.3944	TQM	0.4342	0.5082	0.3790
UMNLP	0.4134	0.4000	0.4277	UMNLP	0.4097	0.3755	0.4507
UMNLP	0.4046	0.3597	0.4622	YR	0.3605	0.3210	0.4110
TQM	0.3548	0.4196	0.3073	YR	0.3483	0.3245	0.3758
YR	0.3417	0.3184	0.3688	JNLP	0.3246	0.3110	0.3393
JNLP	0.3222	0.3347	0.3105	JNLP	0.3103	0.3017	0.3195
WJY	0.3032	0.2700	0.3457	BM24	0.1878	0.1495	0.2522
CAPTAIN	0.1688	0.1793	0.1594	CAPTAIN	0.1574	0.1586	0.1562
NOWJ	0.1313	0.0895	0.2465	NOWJ	0.1306	0.0957	0.2055
NOWJ	0.1224	0.0813	0.2478	WJY	0.1179	0.0870	0.1831
WJY	0.1174	0.0824	0.2042	MIG	0.0508	0.0516	0.0499
UBCS	0.0276	0.0140	0.7196	UBCS	0.0275	0.0140	0.7177
UBCS	0.0272	0.0139	0.7100	CAPTAIN	0.0019	0.0019	0.0019

LLMs, or a combination of both. Specific error analysis for Task 1 would require manual analysis of the whole dataset, which is not feasible due to the sheer amount of data involved in this task. When it comes to the approaches used in this task, we can see the consolidation of trends observed in recent COLIEE editions, especially the combination of traditional IR methods (usually applied at an initial stage) with LLMs used to perform a more sophisticated (but more computationally intensive) processing on a smaller subset of the data.

3 Task 2 - Case Law Entailment

3.1 Task Definition

Given a base case and a chosen specific text fragment together with a second case relevant to the base case, this task consists in determining which paragraphs of the second case entail that fragment of the base case. More formally, given a base case b and its entailed fragment f, and another case r represented by its paragraphs $P = \{p_1, p_2, ..., p_n\}$ such that $noticed(b, r)$ as defined in Sect. 2 is true. The task consists in finding the set $E = \{p_1, p_2, ..., p_m \mid p_i \in P\}$ where $entails(p_i, f)$ denotes a relationship which is true when $p_i \in P$ entails the fragment f.

3.2 Case Law Dataset

In Task 2, 725 query cases and 25,783 paragraphs were provided for training. There were 100 query cases and 3,651 paragraphs in the testing dataset. On average, there are 35.22 candidate paragraphs for each query case in the training dataset, and 35.58 candidate paragraphs for each query case in the testing dataset. The average number of relevant paragraphs for Task 2 was 1.37 paragraphs for training. The average query length is 35.56 words in the training set and 34.97 in the test set. The average candidate length is 106.86 words in the training set and 105.28 in the test set.

3.3 Approaches

Below are the summaries of the submitted models in Task 2 of COLIEE 2024.

- **AMHR** [8] **(three runs)** proposed two approaches: (1) finetuning a legal-BERT model with triplet loss with labels as positive examples and all other paragraphs as negative examples on the train set provided for task 2. This approach resulted in overfitting. (2) finetuning a monoT5 model pre-trained on the MSMARCO dataset with hard negative mining examples chosen by BM25 and another version of the monoT5 model itself. They choose the top-2 predictions by this model as long as the ratio between their similarity score is less than 6.619 (a hyperparameter found by grid search); otherwise, they choose just the first prediction. The second approach got the best results on task 2, this year.
- **CAPTAIN** [6] **(three runs)** introduces a method that builds upon the state-of-the-art approach used in Task 2 of the 2023 competition. This method incorporates zero-shot and few-shot learning techniques to leverage the knowledge stored in large language models. Initially, they fine-tune a pre-trained monoT5 sequence-to-sequence model using hard negative sampling to produce an output. For each query paragraph, they select the top-k candidates with the highest scores to create zero-shot and few-shot prompting techniques for in-context learning with FlanT5 LLM.
- **JNLP** [5] **(three runs)** fine-tuned MonoT5 on the training set of Task 2 with hard negative sampling. The model MonoT5 is a T5-3B reranker fine-tuned on the MS MARCO passage dataset for 10k steps. They used Flan-T5 and Mixtral for prompting.
- **NOWJ** [7] **(three runs)** proposes two approaches of entailment recognition, using multilingual BERT and monoT5 for the three runs. MonoT5 is a T5-based re-ranking model fine-tuned for the downstream task of classification, while mBERT is a traditional approach for document re-ranker. Multilingual BERT and training the mBERT model with weak labels [10] were our last year's solutions. Therefore, for the first two runs, they fine-tuned the models on this year's dataset.
- **OVGU** [11] **(three runs)** team's proposed approach involves using a chain of pre-trained Custom Legal-BERT models that are fine-tuned on sub-datasets generated using BM25 and a Bi-Encoder to select the top-N candidate paragraphs. To enhance the models' robustness, a binomial test is employed for artifact detection. OpenAI's GPT-3.5-turbo model is used to create adversarial instances for selected training instances with annotation artifacts. The large language model was prompted to switch the previous negative entailment label into a positive one for balancing out the training examples with

annotation artifacts. These instances, along with the top-N candidate paragraph dataset, are further used to fine-tune the models. A chained approach is applied during prediction: If the first model (specialized for high precision) fails to predict a hypothesis with at least one premise as 'Entailed,' the second model is used for that hypothesis. If any hypotheses are missed after using the second model, the BM25 top-ranked premise found for a given hypothesis is labeled as 'Entailed.'

Because last year's winning team used monoT5, in this year, most of the teams utilized monoT5. All the four teams that were ranked from 1st to 4th used monoT5, and achieved promising results.

Table 2. Results attained by all teams on the test dataset of task 2.

Team	run	F1	Prec.	Recall	Team	run	F1	Prec.	Recall
AMHR	mt53bk2r	**0.6512**	0.6364	0.6667	CAPTAIN	fs2	0.6360	0.7281	0.5646
JNLP	07f39	0.6320	0.6967	0.5782	CAPTAIN	zs2	0.6235	0.7700	0.5238
CAPTAIN	zs3	0.6235	0.7700	0.5238	NOWJ	t5	0.6117	0.6181	0.6054
JNLP	join-constr	0.6045	0.6694	0.5510	OVGU	2ovgurun1	0.5962	0.5636	0.6327
NOWJ	weak	0.5946	0.5906	0.5986	JNLP	join	0.5912	0.6378	0.5510
OVGU	2ovgurun2	0.5705	0.5506	0.5918	OVGU	2ovgurun3	0.5532	0.5000	0.6190
NOWJ	bert	0.5197	0.5032	0.5374	MIG	mig1	0.4701	0.5673	0.4014
MIG	mig2	0.4696	0.5800	0.3946	AMHR	lsbk2m42	0.3542	0.3617	0.3469
AMHR	lsbk1.txt	0.3320	0.4100	0.2789	MIG	mig3	0.1364	0.0979	0.2245

3.4 Results and Discussion

The F1-measure is used to assess performance in this task. The actual results of the submitted runs by all participants are shown in Table 2, from which it can be seen that the AMHR team attained the best results. CAPTAIN used last year's winner model, which is based on a fine-tuned monoT5, and their model was ranked second. The first ranked model also used fine-tuned monoT5, but they used a hyperparameter value as a threshold of the similarity score, and got the best result this year.

4 Task 3 - Statute Law Information Retrieval

4.1 Task Definition

Statute law task consists of two different tasks. One is the statute law information retrieval task (Task 3), and the other is the entailment task (Task 4). Statute law information retrieval task is a preprocess of entailment task, which retrieves a subset of Japanese Civil Code articles that can be used to judge whether the given statement can be entailed by the entire Civil Code.

Since this task is a preprocess of the entailment task, it is important to include all necessary articles in the returned results. Therefore, we use the F2 measure, which is a variation of the F1 measure that puts more emphasis on recall. In addition, since we also encourage participants to submit more articles for the difficult queries without reducing the overall results, we use the macro average of the F2 measure as the official evaluation measure.

In the last COLIEE (COLIEE 2023) there was a submission using GPT-4 and we discussed whether to exclude the submission from the official results due to lack of reproducibility and contamination problems (e.g., GPT-4 is frequently updated and one cannot guarantee reproducibility, and models trained with undisclosed data may have contamination problems).

In order to exclude the submission of such closed-source models, we introduce the following rules for the submission of tasks.

> Participants should clearly mention what dataset was used (for example: pretrained by Wikipedia dump data as of 2022xxxx, fine-tuned by...) for reproducibility purposes. Participants can use any external data, but it is assumed that they do not use the test dataset and/or something which could directly contain the correct answers of the test dataset (e.g., published results from Japanese Bar Law Exams).

4.2 Statute Law Dataset

We use the Japanese Civil Code with the official English translation for this task. However, if there is no official English translation for a part of this code, we exclude the articles of these parts. As a result, we used a subset of the Japanese Civil Code with 768 articles. Questions are selected from the Japanese bar exam related to this subset and provided in two languages. Japanese version uses original questions and English translated version are provided by the organizers. For the task training data, we also provide sets of relevant articles for Task 3 and entailment results for Task 4.

The training data was constructed by using previous COLIEE data (1097 questions) and new questions (109 questions) were selected from the 2023 bar exam. Of these 109 questions, 88 questions have one relevant article, and 21 questions require two relevant articles.

4.3 Approaches

There are 20 submitted runs from 8 teams. In these submissions, due to the different interpretations made by the participants, there are three varieties of submissions classified by the use of the Large Language Model (LLM).

1. Submissions using LLM whose model is publicly available, but trained with undisclosed training data.
2. Submissions using LLM trained only on disclosed training data.
3. Submissions without explicit use of LLMs (BERT, LegalBERT, ...).

Some participants assume that the LLM whose model is publicly available is good for reproducibility. However, these models do not meet the requirement of disclosed training data. Some participants assume that the use of LLM is prohibited and submit entries without LLM.

Below is a brief summary of the submissions. To clarify 1 and 2, we add underline to the external resource whose model is publicly available but trained with undisclosed training data or its related resources.

- **AMHR** [8] **runs)** uses BM25 to select the top 50 hits and re-ranks the results using monot5-3b-msmarco (language model tuned with MS MARCO for ranking) fine-tuned for COLIEE task 3 to select the top 5 results. They use 3 variants of LLMs (FLAN-T5 and FLAN-alpaca) to select the final relevant articles.
- **BM24 (one run)** uses AnglE-llama-7b-nli (AnglE embedding calculated by using LLaMA) as the text embedding model for semantic retrieval. They fine-tune the system using COLIEE task data (1 and 3), the Supreme Court of Canada Bulk Decisions dataset, and the Semantic Textual Similarity (STS) dataset. They use GPT3.5 to generate similar sentences for the STS.
- **CAPTAIN** [6] **(three runs)** uses three different settings to ensemble the results. The first system (bjpAll) uses BERT-base-Japanese, which is tuned for COLIEE task 3. 4 best checkpoint models are used to generate ensemble results. The second system (bjpAllMonoP) uses MonoT5 (language model tuned with MS MARCO for ranking) fine-tuned for COLIEE task 3 to generate the results, and filters out the results with prompting technique using LLM (Flan T5). They also ensemble the results obtained by the first system. The third system (bjpAllMonoT5) applies the same prompting technique to the bjpAll results to filter out the results. They ensemble the results from the first and second systems.
- **JNLP** [5] **(three runs)** uses BERT-base-Japanese, which is fine-tuned for COLIEE task 3, and they ensemble the predictions of many checkpoints to produce a ranked list. From the ranked list, they use different LLMs to generate final results. For the first system (Mistral), they use the prompt technique of LLM (Mistral) to select the final results. The second system uses RankLLaMA (language model tuned with MS MARCO for ranking based on LLaMA2) to calculate the score for each paring of legal questions and top 5 relevant articles. The third system (constr-join) uses LLM (Orca and Qwen) to get a more concise list from the ranked list. They also include retrieval results from run Mistral to improve recall.
- **NOWJ** [7] **(three runs)** uses a multitask approach to train the BERT for Sequence Classification model using COLIEE Task 3 and Task 4. The results from this model are ensemble with the corresponding scores from the lexical-based BM25 model.
- **PSI (one run)** does not provide a short description.
- **TQM** [4] **(three runs)** uses MonoT5 (language model tuned with MS MARCO for ranking), fine-tuned for COLIEE task 3 for run1. For run2 and run3 they use LightGBM to integrate the results of different models. Light-

Table 3. Evaluation results of submitted runs (Task 3) showing only best runs from each team.

Submission ID	return	retrieved	F2	Precision	Recall	MAP
JNLP.constr-join *	188	**99**	**0.807**	0.709	**0.870**	0.801
CAPTAIN.bjpAllMonoT5	168	96	0.800	0.732	0.845	**0.815**
TQM-run1 #	140	89	0.782	**0.785**	0.800	0.790
NOWJ-25mulreftask-ensemble #	202	96	0.772	0.690	0.835	0.756
AMHR02	185	95	0.749	0.651	0.825	0.740
UA-anglE	233	91	0.711	0.610	0.800	0.700
BM24-1 *	425	94	0.539	0.282	0.795	-
PSI01 ?	109	9	0.086	0.090	0.085	0.231

Table 4. Number of questions with average F2

Average F2	0–0.2	0.2–0.4	0.4–0.6	0.6–0.8	0.8–1.0
number of questions	15	7	21	14	52

Table 5. Evaluation results for 45 questions with anonymized symbols ("A" and "B")

Submission ID	return	retrieved	F2	Precision	Recall	MAP
AMHR02	87	42	0.669	0.561	0.756	0.726
JNLP.constr-join	83	39	0.662	0.586	0.722	0.735
CAPTAIN.bjpAll	95	42	0.647	0.497	0.756	0.742
TQM-run1	56	33	0.628	0.678	0.633	0.719

GBM is a gradient boosting framework that uses tree-based learning algorithms. For run2, they use BM25, Legal BERT, and MonoT5 for integration. For run3, they apply post-processing to the run2 results.

- **UA** [1] **(three runs)** uses Universal AnglE Embedding for the text embedding model for semantic retrieval for 2 runs. The first run (anglE) uses whole articles to compute the embedding and the second run (angleE_chunk) uses single sentences for the embeddings. Cosine similarity is used to calculate the scores to find the relevant articles. The third run (mp_net) uses the sentence transformer model MP-net, which is fine-tuned for task 3.

4.4 Results and Discussion

Table 3 shows the evaluation results of all submissions. Submission IDs with "*" use an LLM whose model is publicly available but trained with undisclosed training data. Those with "#" do not use any LLM.

We confirm that the top performance systems achieve higher average F2 compared to the previous COLIEE. The best performance system is JNLP.constr-join, but it uses LLM with undisclosed training data. The best performance system that satisfies the rule condition is CAPTAIN.bjpAllMonoT5. TQM-run1 is the best performance system among the submissions without LLM. This shows that there was room to improve retrieving performance without using LLM. However, since the submission with LLM can better handle the questions that require semantic matching (e.g., questions with anonymized symbols, such as "A" and "B"), the recall is lower than that of the submission with LLM.

Table 4 shows the number of questions with its average F2 score. Almost one half of the questions (52 questions) have an average F2 greater than 0.8. However, we still have 15 questions whose average F2 is lower than 0.2. Out of these 15 questions, 9 questions use anonymized symbols. This ratio is comparatively higher than the overall average (45 questions out of 109 total). However, the recent development of LLM may have improved performance on these questions. Table 5 shows selected evaluation results for 45 questions with anonymized symbols for the best performance results per team. The best performing system is AMHR02, which uses LLM to select articles. It is important to understand the characteristics of the system through such a detailed analysis of question types.

Finally, we discuss the appropriateness of the rules introduced this year as informally discussed with a number of participants. During the discussion, we found that it is difficult to do an in-depth investigation of the training data used in the system. Therefore, for the next time, we would like to allow the use of any LLM whose model is publicly available and is trained before the Japanese Bar Exam. This is a simple rule to guarantee that the model is good for reproducibility, but avoids encoding answers published.

5 Task 4 - Statute Law Textual Entailment and Question Answering

5.1 Task Definition

Task 4 requires the determination of entailment relationships between a given problem sentence and article sentences. Competitor systems should answer "yes" or "no" regarding the given problem sentences and given article sentences. Participants could use any external data, except that they can not use the test dataset and/or something which could directly contain the correct answers of the test dataset to avoid any "contamination" even in the pretraining/fine-tuning datasets of any software they used. This is because this task is intended to be a pure textual entailment task. We also required the participants to make their system reproducible as per an open academic standard, i.e., they should describe which methods and what datasets were used to enable a reproducible result. Note that this contamination/reproducibility issue does not allow the use of black box LLMs like ChatGPT. To encourage deeper analysis, we asked the participants to submit their outputs when using any fragment of the training dataset (H30, R01, and R02), in addition to the formal runs.

5.2 Dataset

Our training dataset and test dataset are the same as for Task 3. Questions related to Japanese civil law were selected from the Japanese bar exam. The organizers provided a data set used for previous campaigns as training data (1097 questions) and new questions selected from the 2024 bar exam as test data (109 questions).

5.3 Approaches

We describe approaches for each team as follows, shown as a header format of **Team Name (number of submitted runs)**. The slash-separated italic names indicate corresponding huggingface IDs.

- **AMHR** [8] **(three runs)** used approximately 80 prompts, all on the *google/flan-t5-xxl* model, on each question in the training dataset. The best 25 prompts on the training dataset are used to vote on an answer for each question in the test set, where their vote is based on their accuracies on the training dataset, and their accuracies on articles similar (by *sentence-transformers/sentence-t5-x1*, **sentence-transformers/paraphrase-Mini LM -L6-v2**, without fine-tune, and BM25) to the articles used by the test set problem. **AMHR. ensemble0** is the same except the top 50 prompts are used, and the prompts' votes are less based on their previous accuracies. **AMHR.single** is the same except only the best single performing prompt on the train set is selected, without article similarity considered.
- **CAPTAIN** [6] **(three runs)** employs data augmentation that summarizes statute law via *google/flan-t5-xxl* with prompting and filters the good summaries via heuristic rules, generates new pairs of 'Query' and 'Statute Law' by using summary instead of original statute law with various heuristic rules, and fine-tune *google/flan-t5-xxl*. **CAPTAIN2** consists of augmentation and fine-tuning. **CAPTAIN1** uses few shot prompting (using Dense Passage Retrieval for demonstration selection) as input of the model, and then fine-tunes with the augmented data. **CAPTAIN3** generates CoT prompting (by using *google/flan-t5-xxl* for reasoning training data) then ensemble all model.
- **HI (Hybrid Intelligence)** [9] **(three runs) HI1** used *declare-lab/flan-alpaca-gpt4-xl* with zero-shot prompting. **HI2** manually crafted Abstract Dialectical Frameworks (ADF) knowledge representations of a small set of legal articles, ascribing factors to these ADFs for each exam question by zero-shot *declare-lab/flan-alpaca-gpt4-xl*, comparing the logical output of the ADF to the claim in the exam question. **HI3** translated articles into additional ADFs for all articles using GPT3.5-turbo.
- **JNLP** [5] **(three runs)** *JNLP1* and **JNLP2** prompted different large language models (Wqen (their original model), Mistral, Flan-Alpaca, and Flan-T5) and ensemble the results with majority voting, **JNLP1** took the top-1 prompt from Flan-Alpaca while **JNLP1** took the top-2; **JNLP3** prompted Flan-T5 and Mistral, and ensemble the results with the Dawid-Skene label model.

- **KIS** [3] **(three runs) KIS1** employed fine-tuning, few shot learning, retrieval-augmented generation, and a novel method that incorporates character count instructions. additionally, the results were ensemble with rule-based methods. **KIS2** is different from **KIS1** in a unique approach where few shot's data were replaced with outputs generated by GPT-4. **KIS3** used fine-tuning only.
- **NOWJ** [7] **(three runs)** leveraged LLMs in inference phase only. **NOWJ.pandap46** utilized *TheBloke/Panda-7B-v0.1-GPTQ*, used the test set of COLIEE 2023 as the validation set to find the best model and legal prompt. **NOWJ.flant5-panda** combined *google/flan-t5-xl* with panda results following bagging approach. **NOWJ.bagging** combined results from 5 different runs (Panda and Flant5 with different prompts) following the major voting approach.
- **OVGU** [11] **(three runs)** used *MoritzLaurer/DeBERTa-v3-base-mnli-fever-anli*. **4OVGUrun1** and **4OVGUrun3** fine-tuned it by the task 4 COLIEE dataset and a specially created dataset designed to address issues of Word Overlap and Contradiction Word Artefacts, while **4OVGUrun2** fine-tuned it solely with the task 4 dataset. **4OVGUrun1** and **4OVGUrun2** were input datasets of a premise, a hypothesis, and a boolean feature that determines if the hypothesis is a complete subsequence of the premise; **4OVGUrun3** used features to assess the word overlap between premise and hypothesis.
- **UA** [1] **(three runs) UA_stack** used zero-shot learning on *google/flan-t5-xxl* with PromptSource[3] for finding potential good prompts, added one positive and one negative example from the training data as part of each prompt and experimented on the rest of the training set (barring the two examples) to find good prompts, chose the top 3 prompts that gave a good performance on the training data, finally performed zero-shot inference with all three prompts and voting between them. **UA_GPT** followed the same process as **UA_stack** but instead of the top 3 prompts chose the top prompt which is a GPT-3 style prompt. **UA_encoder_decoder** fine-tuned the last two layers from both the decoder and decoder of flan-t5-xxl.

5.4 Results and Discussion

Table 6 shows the COLIEE 2024 Task 4 formal run results. The Formal Run (R05) column shows the result of the COLIEE 2024 formal run using the latest Japanese legal bar exam (Year R05). The columns R02, R01, and H30 are the results using the past formal run datasets, which we required participants to submit, in order to compare different datasets for reference due to the smallness of our datasets. Note that these datasets were already made public as part of our training dataset.

The lower part of the table shows runs with "*" as a suffix of the run names, which used external services where its detailed architecture, training datasets, and model weights are not available, resulting in non-reproducible outputs which are prohibited in our participation call.

[3] https://github.com/bigscience-workshop/promptsource.

Table 6. Accuracies of Task 4 Results. ∗ indicates runs using not fully disclosed models, + indicates runs with preprocessing by such models.

Team	Formal Run		Past Formal Runs		
	# Correct	R05	R02	R01	H30
BaseLine (Yes to all)	60	0.5505	0.5309	0.5315	0.5143
# Correct /# Total		60/109	43/81	59/111	36/70
CAPTAIN2	90	0.8257	0.7901	0.7568	0.8429
JNLP1 ∗	89	0.8165	0.7901	0.6937	0.7429
UA_slack	87	0.7982	0.7407	0.7117	0.7429
UA_encoder_decoder	87	0.7982	0.8395	0.7207	0.7571
CAPTAIN1	86	0.7890	0.8148	0.7748	0.8286
CAPTAIN3	86	0.7890	0.8395	0.7207	0.7286
JNLP2 ∗	86	0.7890	0.8272	0.7297	0.7857
UA_gpt	85	0.7798	0.7901	0.6847	0.7571
AMHR.ensembleA50	84	0.7706	0.8148	0.3784	0.6571
AMHR.single	84	0.7706	0.7901	0.3874	0.6714
HI1	82	0.7523	0.7284	0.6667	0.7000
NOWJ.pandap46 ∗	82	0.7523	N/A	N/A	N/A
AMHR.ensembleA0	80	0.7339	0.7778	0.4234	0.7000
JNLP3 ∗	80	0.7339	0.7901	0.6126	0.6571
NOWJ.flant5-panda ∗	80	0.7339	N/A	N/A	N/A
NOWJ.bagging ∗	78	0.7156	N/A	N/A	N/A
OVGU1 +	77	0.7064	0.7531	0.6937	0.6714
KIS2 +	76	0.6972	0.6543	0.6036	0.6429
OVGU3 +	76	0.6972	0.7654	0.6306	0.7000
OVGU2 +	70	0.6422	0.6790	0.6396	0.6000
KIS1	67	0.6147	0.6420	0.6847	0.6286
HI3	64	0.5872	0.6296	0.6306	0.6000
HI2	63	0.5780	0.7531	0.6937	0.7143
KIS3	62	0.5688	0.5926	0.6306	0.6429

The best runs by team **CAPTAIN2** used an LLM (flan-T5) with data augmentation and heuristic rules, while all runs in Task 4 used LLMs in some form. Comparing the results of the past test data (R02, R01, and H30), we found that the scores changed but one of the runs of the CAPTAIN team was top ranked.

There is still concern about the usage of LLMs. For example, it is not clear in what way the GPT-based generative AIs could handle logical reasoning. A possibility is that they can apply superficially similar descriptions which include the use of logical reasoning, so they do not directly handle logic but indirectly reflect the use of logic in existing descriptions and their combinations, i.e., their huge stack of similar contents led to providing approximate answers and marginally related evidence. Because Task 4 is intended to be a pure textual entailment task, superficial similarities without logical reasoning would not make much sense, thus we need further investigations about the capability of the generative AIs on logical reasoning. However, as a practical legal application, it can be useful when there are, to some extent, similar contents available as previous existing cases. For our future work, we need new task designs which provide a framework for the explainability of results and to evaluate the explainability of the solvers in more practical task settings.

6 Conclusion

We have summarized the systems and their performance as submitted to the COLIEE 2024 competition. For Task 1, some participants used TF-IDF, BERT, and BM25. In Task 2, many teams used fine-tuned monoT5 and showed similar performances. For Task 3, many teams use BM25 and MS-Marco-based re-ranker. Postprocess using LLM and ensemble technique also improves the performance. Lastly, for Task 4, all runs use LLMs with different ideas to fine-tune them. We intend to further continue to improve dataset quality in future editions of COLIEE so the tasks more accurately represent real-world problems.

This year we introduce the rule to usage of external resources to maintain the reproducibility and avoid the problem of contamination. However, we need to update the rules to improve clarity.

Acknowledgements. This competition would not be possible without the significant support of Colin Lachance from vLex and Compass Law. Our work to create and run the COLIEE competition is also supported by our institutions: the National Institute of Informatics (NII), Shizuoka University and Hokkaido University in Japan, and the University of Alberta and the Alberta Machine Intelligence Institute in Canada. This research is also supported by the Canadian Natural Sciences and Engineering Research Council (NSERC) [including funding reference numbers RGPIN-2022-03469 and DGECR-2022-00369] and Alberta Innovates. This work was also supported by JSPS KAKENHI Grant Numbers, JP22H00543 and JST, AIP Trilateral AI Research, Grant Number JPMJCR20G4. In addition, this research was supported by the Brain Pool program funded by the Ministry of Science and ICT through the National Research Foundation of Korea (RS-2023-00304286).

References

1. Babiker, H., Rahman, M.A., Kim, M.Y., Rabelo, J., Goebel, R.: Legal yes/no question answering through text embedding, fine-tuning, and prompt engineering. In: Proceedings of the Eighteenth International Workshop on Juris-Informatics (JURISIN 2024) (2024)
2. Curran, D., Conwa, M.: Similarity ranking of case law using propositions as features. In: Proceedings of the Eighteenth International Workshop on Juris-Informatics (JURISIN 2024) (2024)

3. Fujita, M., Onaga, T., Kano, Y.: LLM tuning and interpretable CoT: team in COLIEE 2024. In: Proceedings of the Eighteenth International Workshop on Juris-Informatics (JURISIN 2024) (2024)
4. Li, H., Chen, Y., Ge, Z., Ai, Q., Liu, Y., Zhou, Q., Huo, S.: Towards an in-depth comprehension of case relevance for better legal case retrieval. In: Proceedings of the Eighteenth International Workshop on Juris-Informatics (JURISIN 2024) (2024)
5. Nguyen, C., et al.: Pushing the boundaries of legal information processing with integration of large language models. In: Proceedings of the Eighteenth International Workshop on Juris-Informatics (JURISIN 2024) (2024)
6. Nguyen, P., et al.: CAPTAIN at COLIEE 2024: large language model for legal text retrieval and entailment. In: Proceedings of the Eighteenth International Workshop on Juris-Informatics (JURISIN 2024) (2024)
7. Nguyen, T.M., Nguyen, H.L., Nguyen, D.Q., Nguyen, H.T., Vuong, T.H.Y., Nguyen, H.T.: NOWJ@COLIEE 2024: leveraging advanced deep learning techniques for efficient and effective legal information processing. In: Proceedings of the Eighteenth International Workshop on Juris-Informatics (JURISIN 2024) (2024)
8. Nighojkar, A., et al.: AMHR COLIEE 2024 entry: legal entailment and retrieval. In: Proceedings of the Eighteenth International Workshop on Juris-Informatics (JURISIN 2024) (2024)
9. Steging, C., Leeuwen, L.V.: A hybrid approach to legal textual entailment. In: Proceedings of the Eighteenth International Workshop on Juris-Informatics (JURISIN 2024) (2024)
10. Vuong, Y.T.H., et al.: SM-BERT-CR: a deep learning approach for case law retrieval with supporting model. Artif. Intell. Law **31**(3), 601–628 (2023)
11. Wehnert, S., Murugadas, V., Naik, P.V., Luca, E.W.D.: Improving robustness in language models for legal textual entailment through artifact-aware training. In: Proceedings of the Eighteenth International Workshop on Juris-Informatics (JURISIN 2024) (2024)