Proceedings of the Tenth International Competition on Legal Information Extraction/Entailment (COLIEE 2023)

in association with
the 19th International Conference
on Artificial Intelligence and Law

June 19, 2023

Preface

This volume contains the papers accepted for presentation at COLIEE 2023, the Tenth Competition on Legal Information Extraction/Entailment, held in conjunction with the International Conference on Artificial Intelligence and Law (ICAIL 2023) in Braga, Portugal, on June 19, 2023. It is a significant highlight that the COLIEE competition has completed a decade of organizing COLIEE, and has helped nurture the development of a world-wide community of research on AI and law. In what began as only a handful of competitors from Japan and Canada, the competition has spread world wide, and has now had as many as 30 different teams from Africa, Asia, Europe, and North American, including more than 25 different countries.

As in previous years, the overall goal of COLIEE is to formulate a challenging legal informatics competition that engages researchers around the world, and helps build a community that would consider all of computer science and Artificial Intelligence methods to tackle the problems of legal reasoning.

This year there are 19 different teams from 7 countries (China, Canada, Japan, Vietnam, India, United States, Taiwan). Each submission was reviewed by at least 3 program committee members. In addition, the organizers provided a summary paper for the COLIEE 2023 tasks and solutions, which is accepted for publication in the main ICAIL conference. We have now completed ten competitions, all of whose selected contributions have been peer-reviewed, published in the ICAIL proceedings or at the Springer Lecture Notes in Artificial Intelligence series, and – most importantly – developed a community of researchers, lawyers, judges, and related communities to discuss the impact and future of technology adoption in for legal systems.

The COLIEE organizers would like to acknowledge the continued support of people and organizations around the planet, including Colin Lachance from Compass Law/Vlex in Canada, who has been particularly support in his work to help develop and extend the case law data for COLIEE, and to Young Yik Rhim of Intellicon in Seoul, who has been our advocate since the beginning of COLIEE. In addition, a number of Japanese colleagues (in addition to the organizing team participants of Ken Satoh, Yoshinobu Kano, and Masaharu Yoshioka) have contributed to the extension and curation of the statute law data for the COLIEE competition.

June 19, 2023
Braga, Portugal

Randy Goebel, University of Alberta, Canada
Yoshinobu Kano, Shizuoka Univesity, Japan
Mi-Young Kim, University of Alberta, Canada
Juliano Rabelo, University of Alberta, Canada
Ken Satoh, National Institute of Informatics, Japan
Masaharu Yoshioka, Hokkaido University, Japan
COLIEE 2023 organizers

Program Committee

Kripabandhu Ghosh	Indian Institute of Science Education and Research (IISER) Kolkata, India
Saptarshi Ghosh	Indian Institute of Technology Kharagpur
Randy Goebel	University of Alberta
Yoshinobu Kano	Shizuoka University
Mi-Young Kim	Department of Computing Science, U. of Alberta, Canada
Nguyen Le Minh	Graduate School of Information Science, Japan Advanced Institute of Science and Technology
Makoto Nakamura	Niigata Institute of Technology
María Navas-Loro	Universidad Politécnica de Madrid
Juliano Rabelo	AMII
Julien Rossi	Amsterdam Business School
Ken Satoh	National Institute of Informatics and Sokendai, Japan
Jaromir Savelka	Carnegie Mellon University
Yunqiu Shao	Tsinghua University
Satoshi Tojo	Asia University
Vu Tran	The Institute of Statistical Mathematics, Japan
Josef Valvoda	University of Cambridge
Sabine Wehnert	Otto-von-Guericke-Universität Magdeburg
Hannes Westermann	University of Montreal
Hiroaki Yamada	Tokyo Institute of Technology
Masaharu Yoshioka	Hokkaido University

Additional Reviewers

Nguyen, Ha-Thanh

Table of Contents

THUIR@COLIEE 2023: Incorporating Structural Knowledge into Pre-trained Language Models for Legal Case Retrieval

Haitao Li
DCST, Tsinghua University
Quan Cheng Laboratory
Beijing 100084, China
liht22@mails.tsinghua.edu.cn

Weihang Su
DCST, Tsinghua University
Quan Cheng Laboratory
Beijing 100084, China
swh22@mails.tsinghua.edu.cn

Changyue Wang
DCST, Tsinghua University
Quan Cheng Laboratory
Beijing 100084, China
changyue20@mails.tsinghua.edu.cn

Yueyue Wu
DCST, Tsinghua University
Quan Cheng Laboratory
Beijing 100084, China
wuyueyue@mail.tsinghua.edu.cn

Qingyao Ai
DCST, Tsinghua University
Quan Cheng Laboratory
Beijing 100084, China
aiqy@tsinghua.edu.cn

Yiqun Liu*
DCST, Tsinghua University
Quan Cheng Laboratory
Beijing 100084, China
yiqunliu@tsinghua.edu.cn

ABSTRACT

Legal case retrieval techniques play an essential role in modern intelligent legal systems. As an annually well-known international competition, COLIEE is aiming to achieve the state-of-the-art retrieval model for legal texts. This paper summarizes the approach of the championship team THUIR in COLIEE 2023. To be specific, we design structure-aware pre-trained language models to enhance the understanding of legal cases. Furthermore, we propose heuristic pre-processing and post-processing approaches to reduce the influence of irrelevant messages. In the end, learning-to-rank methods are employed to merge features with different dimensions. Experimental results demonstrate the superiority of our proposal. Official results show that our run has the best performance among all submissions. The implementation of our method can be found at https://github.com/CSHaitao/THUIR-COLIEE2023.

CCS CONCEPTS

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

KEYWORDS

legal case retrieval, dense retrieval, pre-training

ACM Reference Format:
Haitao Li, Weihang Su, Changyue Wang, Yueyue Wu, Qingyao Ai, and Yiqun Liu. 2023. THUIR@COLIEE 2023: Incorporating Structural Knowledge into Pre-trained Language Models for Legal Case Retrieval. In *Proceedings of COLIEE 2023 workshop, June 19, 2023, Braga, Portugal*. ACM, New York, NY, USA, 6 pages.

1 INTRODUCTION

In countries with case law systems, precedent is an important determinant for the decision of new given cases [13, 25]. Therefore, it takes a substantial amount of time for legal workers to find precedents that support or contradict a new case. With the growing number of digital legal cases, it is increasingly more expensive for legal practitioners to find precedents. Recently, the growing works have raised the awareness that legal search systems will free people from the heavy manual work [1, 2, 16, 17, 24, 30].

In ad-hoc retrieval and open-domain search, contextual language models such as BERT have brought significant performance gains to the first stage of retrieval [28]. Despite their great success, applying language models to legal case retrieval is not trivial with the following main challenges.

Firstly, it is labor-intensive to construct high-quality annotated datasets for legal case retrieval due to the need for legal knowledge. Hence, the current dataset usually has only a few thousand training data, which may lead to over-fitting of the language model. Secondly, legal cases are usually long texts with internal writing logic. To be specific, legal cases usually contain three parts: Fact, Reasoning, and Decision. The Fact section describes the defendant's and plaintiff's arguments, evidence, and basic events. The Reasoning section is the analysis by the judges of the legal issues in the facts. The Decision section is the specific response of the court to all legal issues. Limited by the input length of 512 tokens, existing language models either truncate the redundant content or flatten the input of all structures, making it difficult to understand legal cases properly.

To tackle the above challenges, we propose SAILER [9], which stands for **S**tructure-**A**ware pre-tra**I**ned language model for **LE**gal case **R**etrieval. SAILER utilizes an encoder-decoder architecture to explicitly model the relationships between different structures and learns the legal knowledge implied in the structures through pre-training on a large number of legal cases.

To verify the effectiveness of SAILER, the THUIR team participates in the COLIEE 2023 legal case retrieval task and wins the championship. This paper elaborates on our technical solutions and demonstrates the effectiveness of incorporating structural knowledge into pre-trained language models.

The remainder of the paper is organized as follows: Section 2 introduces the background for legal case retrieval and dense retrieval. Section 3 presents the description, datasets, and evaluation

*Corresponding author

COLIEE 2023, June 19, 2023, Braga, Portugal

Table 1: Dataset statistics of COLIEE Task 1.

	COLIEE 2021		COLIEE 2022		COLIEE 2023	
	Train	Test	Train	Test	Train	Test
# of queries	650	250	898	300	959	319
# of candidate case per query	4415	4415	3531	1263	4400	1335
avg # of relevant candidates/paragraphs	5.17	3.6	4.68	4.21	4.68	2.69

metrics of the COLIEE 2023 legal case retrieval task. In Section 4, the technical details are elaborated. After that, Section 5 introduces the experiment results. Finally, we conclude this paper in Section 6 by summarizing the major findings and discussing future work.

2 RELATED WORK

2.1 Legal Case Retrieval

Legal case retrieval, which aims to identify relevant cases for a given query case, is a key component of intelligent legal systems. A number of deep learning methods have been applied to retrieve precedents with various techniques, such as CNN-based models [26], BiDAF [23], SMASH-RNN [8], etc. Recently, researchers have attempted to achieve performance gains in legal case retrieval with transformer-based language models. For example, Shao et al. [24] propose BERT-PLI, which divides the case into multiple paragraphs and aggregates the scores together with neural networks. Furthermore, researchers have begun to design legal-oriented pre-trained models, such as Lawformer [27] and LEGAL-BERT [3]. However, neither of them design pre-training tasks for legal case retrieval. We believe that the potential of language models for legal case retrieval has not been fully exploited.

2.2 Dense Retrieval

Dense retrieval is a powerful retrieval paradigm that can effectively capture contextual information [5–7, 10, 18, 33]. Generally speaking, dense retrieval maps queries and documents to dense embeddings with a dual encoder. Later, the inner product is applied to measure their relevance. For better performance, researchers have designed pre-trained objectives oriented to web search, which achieve state-of-the-art effectiveness. For example, Zhan et al. [32] propose dynamic negative sampling to further improve performance. Chen et al. propose ARES [5], which attempts to incorporate axioms into the pre-training process.

3 TASK OVERVIEW

3.1 Task Description

The Competition on Legal Information Extraction/Entailment (COLIEE) is an annual international competition whose aim is to achieve state-of-the-art methods for legal text processing. There are four tasks in COLIEE 2023, and we submit systems to task 1.

Task 1 is the legal case retrieval task, which involves identifying supporting cases for the decision of query cases from the entire corpus. Formally, given a query case Q and a set of candidate cases S, this task is to identify all the supporting cases $S_Q^* = \{S_1, S_2, ..., S_n\}$ from a large candidate pool. The supporting cases are also named "noticed cases". For each query, participants can return any number of supporting cases that they consider relevant.

3.2 Data Corpus

The data corpus for Task 1 belongs to a database of case law documents from the Federal Court of Canada provided by Compass Law. Statistics of the dataset are shown in Table 1. From COLIEE 2021, all queries share a large candidate case pool, which is more challenging and realistic. The COLIEE 2023 dataset contains 959 query cases against 4400 candidate cases for training and 319 query cases against 1335 candidate cases for testing.

On further analysis, we find that the average number of relevant documents per query in the training set is 4.68 while the number of relevant documents in the test set is 2.69. Therefore, we predict the top-5 possible relevant cases to calculate the evaluation metrics during training. At testing time, we adopt heuristic post-processing to avoid the performance damage caused by the inconsistent distribution of the training and testing sets. We randomly select 187 queries as the validation set and the remaining 772 queries as the training set.

3.3 Metrics

For COLIEE 2023 Task 1, evaluation measures will be precision, recall, and F-measure:

$$\text{Precision} = \frac{\#TP}{\#TP + \#FP} \tag{1}$$

$$\text{Recall} = \frac{\#TP}{\#TP + \#FN} \tag{2}$$

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

where $\#TP$ is the number of correctly retrieved candidate cases for all query cases, $\#FP$ is the number of falsely retrieved candidate cases for all query cases, and $\#FN$ is the number of missing noticed candidate paragraphs for all query cases. It is worth noting that micro-average (evaluation measure is calculated using the results of all queries) was used rather than marco-average (evaluation measure is calculated for each query and then takes average) in the evaluation process.

4 METHOD

In this section, we present the complete solution of the COLIEE 2023 Task 1. To be specific, we first perform a simple pre-processing of the data. Then, we implement traditional retrieval methods and pre-trained language models. Furthermore, we extract multiple features for each query-candidate pair. Learning-to-rank methods are

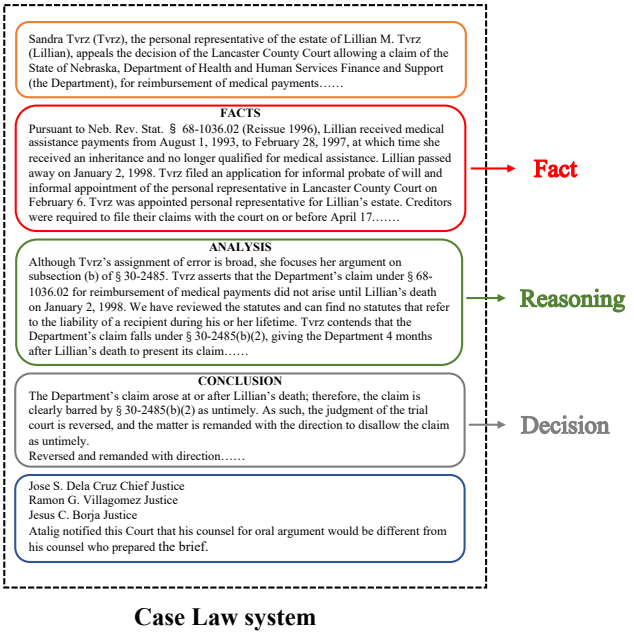


Figure 1: An example of the legal case structure in the Case Law system.

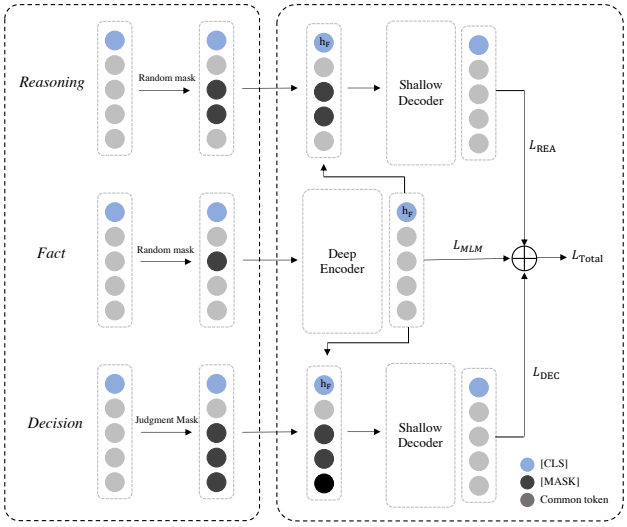


Figure 2: The model design for SAILER, which consists of a deep encoder and two shallow decoders. The Reasoning and Decision section are aggressively masked, joined with the Fact embedding to reconstruct the key legal elements and the judgment results.

employed to aggregate these features for the score. At last, we design heuristic post-processing methods to form the final submission list.

4.1 Pre-processing

Before training, we perform the following pre-processing:

4.1.1 Remove useless information. Firstly, we directly remove the content before character "[1]", which is usually procedural information for that legal case, such as time, court, etc. Then, we remove the placeholders, such as "FRAGMENT_SUPPRESSED" etc. When calculating the similarity, these placeholders are considered as noise. Furthermore, we note that some legal cases contain French text and Langdetect is employed to remove all French paragraphs. For a few documents with a high percentage of French text, we translate them into English to retain the main information.

4.1.2 Summary extraction. A part of the case has the subheading of "Summary". The summary section usually contains the important content of cases. Therefore, we extract the summary by regular matching and concatenate it at the beginning of the processed text.

4.1.3 Reference sentence extraction. Inspired by [15], we are aware that placeholders such as "FRAGMENT_SUPPRESSED", "REFERENCE_SUPPRESSED", "CITATION_SUPPRESSED", are citations or references from other noticed cases. These sentences are directly relevant to the supporting cases. Therefore, for all queries, we keep only the sentences with placeholders to further improve performance. Noticeably, for the candidate cases, we retain the full content.

4.2 Traditional Lexical Matching Models

According to previous findings [1, 15, 19, 20], the traditional lexical matching models are competitive in legal case retrieval tasks. Therefore, we first implement the following lexical matching approach.

4.2.1 TF-IDF. TF-IDF [21] is a classical lexical matching model, which is the combination of term frequency (TF) and inverse document frequency (IDF). Their equations are shown as follows:

$$TF(t_{i,j}) = \frac{n_{i,j}}{\sum_k n_{k,j}} \tag{4}$$

$$IDF(t_i) = \log \frac{|D|}{|D_i + 1|} \tag{5}$$

$$TF - IDF = TF \times IDF \tag{6}$$

where D is the total number of documents in the corpus and D_i represents the number of documents containing the word t_i. $n_{i,j}$ denotes the number of words t_i in the document d_j.

4.2.2 BM25. BM25 [22] is a probabilistic relevance model based on bag-of-words. Given a query q and a document d, the formula of BM25 is shown as follows:

$$BM25(d,q) = \sum_{i=1}^{M} \frac{IDF(t_i) \cdot TF(t_i,d) \cdot (k_1+1)}{TF(t_i,d) + k_1 \cdot \left(1 - b + b \cdot \frac{len(d)}{avgdl}\right)} \tag{7}$$

where k_1, b are free hyperparameters, TF represents term frequency and IDF represents inverse document frequency. $avgdl$ is the average length of all documents.

Table 2: Features that we used for learning to rank. The placeholder contains "FRAGMENT_SUPPRESSED", "REFERENCE_SUPPRESSED", "CITATION_SUPPRESSED".

Feature ID	Feature Name	Description
1	query_length	Length of the query
2	candidate_length	Length of the candidate paragraph
3	query_ref_num	Number of placeholders in the query case
4	doc_ref_num	Number of placeholders in the candidate case
5	BM25	Query-candidate scores with BM25 (k_1 = 3.0 , b = 1.0)
6	QLD	Query-candidate scores with QLD
7	TF-IDF	Query-candidate scores with TF-IDF
8	SAILER	Inner product of query and candidate vectors generated by SAILER

4.2.3 QLD. QLD [31] is another efficient probabilistic statistical model which calculates relevance scores by considering the probability of query generation. Given a query q and a document d, the score of QLD is calculated as follows:

$$\log p(q|d) = \sum_{i:c(q_i;d)>0} \log \frac{p_s(q_i|d)}{\alpha_d p(q_i|C)} + n \log \alpha_d + \sum_i \log p(q_i|C) \tag{8}$$

The details can be referred to Zhai et al.'s work[31].

4.3 SAILER

As mentioned above, legal cases usually contain three parts: Fact, Reasoning, and Decision. Figure 1 illustrates an example of the legal case structure. Key information in the Facts will be carefully analyzed in the Reasoning and influence the final decision. Furthermore, the Reasoning and Decision are written based on the extensive domain knowledge of the judges. Incorporating the rich knowledge inherent in the structure into language models is essential for understanding legal cases.

To achieve the above goals, we propose SAILER [9], which is shown in Figure 2. More specifically, SAILER consists of a deep encoder and two shallow decoders. The Fact part is fed to the deep encoder to form a dense vector h_f. Then, h_f is concatenated with the positively masked Reasoning and Decision, respectively, which is fed to the shallow decoder. Since the shallow decoder with limited power, h_f is forced to pay more attention to the useful information in the Fact which is relevant to the Reasoning and Decision sections.

To construct the pre-training corpus, we collect 50w legal cases from the U.S. federal and state courts [1]. Then, we extract the corresponding section with regular matching. During the pre-training phase, we optimize the model with the following loss function:

$$L_{Total} = L_{MLM} + L_{REA} + L_{DEC} \tag{9}$$

$$L_{MLM} = -\sum_{x' \in m(F)} \log p(x'|F \backslash m(F)) \tag{10}$$

$$L_{REA} = -\sum_{x' \in m(R)} \log p(x'|[h_F, R \backslash m(R)]) \tag{11}$$

$$L_{DEC} = -\sum_{x' \in m(D)} \log p(x'|[h_F, D \backslash m(D))]) \tag{12}$$

where F, R, D denote Fact, Reasoning and Decision section respectively. $m(F), m(R), m(D)$ are the masked token of the corresponding section. Only a small percentage of the token (0%-30%) in the Fact section is masked since most of the information has to be preserved. The Reasoning and Decision sections have an aggressive masking rate (30%-60%) for a better vector representation.

After pre-training, we employ contrastive learning loss to finetune. More specifically, given a query case q, let d^+ and d^- be relevant and negative cases, the loss function L is formulated as follows:

$$L(q, d^+, d_1^-, ..., d_n^-) = -\log \frac{exp(s(q, d^+))}{exp(s(q, d^+)) + \sum_{j=1}^{n} exp(s(q, d_j^-))} \tag{13}$$

For each query, we take the irrelevant cases from the top 100 cases recalled by BM25 as negative examples.

4.4 Learning to Rank

Following up on previous work [4, 11, 29], learning to rank techniques are used to further improve performance. In this paper, we integrate all features into the final score with Lightgbm. Table 2 shows the details of all the features. We employ NDCG as the ranking optimization objective and select the model that performs best on the validation set for testing.

4.5 Post-processing

After getting the ranking scores, we perform the following post-processing strategy:

4.5.1 Filtering by trial date. Since query cases can only cite cases that are judged before itself, we filter the candidate set according to trial date. Specifically, we extract all the dates in the case, i.e., four digits within a reasonable range. Then, the largest date that appears is regarded as the trial date of the case. This avoids wrong filtering caused by treating other dates as the trial date. If the trial date of the query case is unknown, its candidate set contains all other cases.

[1] https://case.law/

Table 3: Performance of single model on COLIEE 2023 validation set."-" represents the unlimited length.

model	max_length	P@5	R@5	F1 score
BM25(k_1=3,b=1)	512	0.0963	0.1067	0.1012
QLD	512	0.0983	0.1091	0.1035
BERT	512	0.0770	0.0854	0.0809
RoBERTa	512	0.0994	0.1103	0.1046
LEGAL-BERT	512	0.0845	0.0937	0.0888
SAILER	512	0.1315	0.1459	0.1385
TF-IDF	-	0.0898	0.1504	0.1142
BM25(k_1=3,b=1)	-	0.1465	0.1625	0.1541
QLD	-	0.1411	0.1565	0.1484

4.5.2 Filtering query cases. We note that the average number of times that query cases are noticed is 0.056 in the training set. Therefore, after getting the relevant cases for each query, we delete all query cases included in it.

4.5.3 Dynamic cut-off. It is noticeable that the number of cases relevant to each query case is variable. Therefore we employ dynamic cut-off to identify the relevant cases for each query. We define l as the minimum number of noticed cases and h as the maximum number of noticed cases. After that, we take the highest score S as the basis, and only cases with scores greater than $p \times S$ are returned. Grid search is performed on the validation set to determine the optimal value of p, l, h.

5 EXPERIMENT

We conduct experiments to verify the effectiveness of our proposed method. Specifically, this section investigates the following research questions:

- **RQ1**: What are the advantages of SAILER over the previous pre-trained and lexical matching models?
- **RQ2**: How do different post-processing strategies affect final performance?

5.1 Implementation Details

For traditional lexical matching models, we implement them with the pyserini toolkit [2]. We notice that BM25 does not perform well with the default parameters, so we set $k_1 = 3.0$ and $b = 1.0$.

For pre-training, the masking rate of the encoder is 0.15, and the masking rate of decoders is 0.45. We pre-train up to 10 epochs using AdamW [14] optimizer, with a learning rate of 1e-5, batch size of 72, and linear schedule with warmup ratio of 0.1. In the fine-tuning process, the ratio of positive to negative samples is 1:15. We fine-tune up to 20 epochs using the AdamW [14] optimizer, with a learning rate of 5e-6, batch size of 4, and linear schedule with warmup ratio 0.1. All the experiments in this work are conducted on 8 NVIDIA Tesla A100 GPUs.

For learning to rank, we set the learning rate to 0.01, the number of leaves to 20, and the early stopping step to 100. The boosting_type is "gbdt" and the objective is "lambdarank". During post-processing, l/h are eventually 4/6 respectively, and p is set to 0.84.

[2]https://github.com/castorini/pyserini

Table 4: Ensemble with different post-processing strategies

model	P@5	R@5	F1 score
Ensemble	0.1863	0.2032	0.1944
+Filtering by trial date	0.2070	0.2290	0.2175
+Filtering query cases	0.2092	0.2314	0.2197
+Dynamic cut-off	0.2177	0.2385	0.2276

Table 5: Final top-5 of COLIEE 2023 Task 1 on the test set.

Team	Submission	Precision	Recall	F1
THUIR	thuirrun2	0.2379	0.4063	0.3001
THUIR	thuirrun3	0.2173	0.4389	0.2907
IITDLI	iitdli_task1_run3	0.2447	0.3481	0.2874
THUIR	thuirrun1	0.2186	0.3782	0.2771
NOWJ	nowj.d-ensemble	0.2263	0.3527	0.2757

5.2 Experiment Result

To answer **RQ1**, we compare the performance of different single models and analyze the strengths and weaknesses of pre-trained language models. Table 3 shows the performance comparison of the different methods. We can get the following observations:

- When the input lengths of the models are the same, the performance of RoBERTa [12] is approximate to that of BM25 and QLD. Since there are no pre-training tasks designed for dense retrieval, LEGAL-BERT [3] does not achieve competitive performance.
- Benefiting from the expert knowledge inherent in the structure of legal cases, SAILER outperforms traditional lexical matching models and pre-trained language models under the same conditions.
- However, the performance of BM25 and QLD is further improved when the input length is not limited. The traditional lexical matching model is still competitive under long-text legal cases. The input length limits the further understanding of the legal instrument by language models. In the future, we will continue to explore the performance of language models based on Longformer for legal case retrieval.

To answer question **RQ2**, we employ different post-processing strategies on the score of ensemble. From the experimental results in Table 4, we can obtain the following observations:

- Compared with the effectiveness of single models, learning to rank incorporates multiple features and achieves further performance improvements.
- All three post-processing strategies facilitate performance improvement. Narrowing the candidate set for each query via the strategy of filtering by trial date achieves the best boosting effect.

The final top-5 results of COLIEE 2023 Task 1 are illustrated in Table 5. Our run2 has the best performance and is significantly better than other runs. Run 3 and Run 1 are other processing methods with different parameters. Finally, the THUIR team wins the championship.

6 CONCLUSION

This paper presents THUIR Team's approaches to the legal case retrieval task in the COLIEE 2023 competition. Due to the limited training data, we employ a legal-oriented pre-training model to improve performance. Furthermore, diverse pre-processing and post-processing approaches are presented. Also, we utilize learning to rank to merge the different features into the final score. Finally, we win first place in this competition. In the future, we will explore more pre-training objectives suitable for legal case retrieval.

REFERENCES

[1] Sophia Althammer, Arian Askari, Suzan Verberne, and Allan Hanbury. 2021. DoSSIER@ COLIEE 2021: leveraging dense retrieval and summarization-based re-ranking for case law retrieval. *arXiv preprint arXiv:2108.03937* (2021).

[2] Trevor Bench-Capon, Michał Araszkiewicz, Kevin Ashley, Katie Atkinson, Floris Bex, Filipe Borges, Daniele Bourcier, Paul Bourgine, Jack G Conrad, Enrico Francesconi, et al. 2012. A history of AI and Law in 50 papers: 25 years of the international conference on AI and Law. *Artificial Intelligence and Law* 20, 3 (2012), 215–319.

[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] Jia Chen, Haitao Li, Weihang Su, Qingyao Ai, and Yiqun Liu. 2023. THUIR at WSDM Cup 2023 Task 1: Unbiased Learning to Rank. arXiv:2304.12650 [cs.IR]

[5] Jia Chen, Yiqun Liu, Yan Fang, Jiaxin Mao, Hui Fang, Shenghao Yang, Xiaohui Xie, Min Zhang, and Shaoping Ma. 2022. Axiomatically Regularized Pre-training for Ad hoc Search. (2022).

[6] Qian Dong, Yiding Liu, Suqi Cheng, Shuaiqiang Wang, Zhicong Cheng, Shuzi Niu, and Dawei Yin. 2022. Incorporating Explicit Knowledge in Pre-trained Language Models for Passage Re-ranking. *arXiv preprint arXiv:2204.11673* (2022).

[7] Yixing Fan, Xiaohui Xie, Yinqiong Cai, Jia Chen, Xinyu Ma, Xiangsheng Li, Ruqing Zhang, Jiafeng Guo, et al. 2022. Pre-training methods in information retrieval. *Foundations and Trends® in Information Retrieval* 16, 3 (2022), 178–317.

[8] 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*. 795–806.

[9] Haitao Li, Qingyao Ai, Jia Chen, Qian Dong, Yueyue Wu, Yiqun Liu, Chong Chen, and Qi Tian. 2023. SAILER: Structure-aware Pre-trained Language Model for Legal Case Retrieval. arXiv:2304.11370 [cs.IR]

[10] Haitao Li, Qingyao Ai, Jingtao Zhan, Jiaxin Mao, Yiqun Liu, Zheng Liu, and Zhao Cao. 2023. Constructing Tree-based Index for Efficient and Effective Dense Retrieval. arXiv:2304.11943 [cs.IR]

[11] Haitao Li, Jia Chen, Weihang Su, Qingyao Ai, and Yiqun Liu. 2023. Towards Better Web Search Performance: Pre-training, Fine-tuning and Learning to Rank. *arXiv preprint arXiv:2303.04710* (2023).

[12] 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 preprint arXiv:1907.11692* (2019).

[13] Daniel Locke and Guido Zuccon. 2022. Case law retrieval: problems, methods, challenges and evaluations in the last 20 years. *arXiv preprint arXiv:2202.07209* (2022).

[14] Ilya Loshchilov and Frank Hutter. 2018. Fixing weight decay regularization in adam. (2018).

[15] Yixiao Ma, Yunqiu Shao, Bulou Liu, Yiqun Liu, Min Zhang, and Shaoping Ma. 2021. Retrieving legal cases from a large-scale candidate corpus. *Proceedings of the Eighth International Competition on Legal Information Extraction/Entailment, COLIEE2021* (2021).

[16] 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*. 2342–2348.

[17] Yixiao Ma, Yueyue Wu, Weihang Su, Qingyao Ai, and Yiqun Liu. 2023. CaseEncoder: A Knowledge-enhanced Pre-trained Model for Legal Case Encoding. arXiv:2305.05393 [cs.IR]

[18] Yingqi Qu, Yuchen Ding, Jing Liu, Kai Liu, Ruiyang Ren, Wayne Xin Zhao, Daxiang Dong, Hua Wu, and Haifeng Wang. 2021. RocketQA: An Optimized Training Approach to Dense Passage Retrieval for Open-Domain Question Answering. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 5835–5847.

[19] 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.

[20] 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 *JSAI International Symposium on Artificial Intelligence*. Springer, 196–210.

[21] Juan Ramos et al. 2003. Using tf-idf to determine word relevance in document queries. In *Proceedings of the first instructional conference on machine learning*, Vol. 242. Citeseer, 29–48.

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

[23] Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. 2016. Bidirectional attention flow for machine comprehension. *arXiv preprint arXiv:1611.01603* (2016).

[24] 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*. 3501–3507.

[25] Yunqiu Shao, Yueyue Wu, Yiqun Liu, Jiaxin Mao, and Shaoping Ma. 2023. Understanding Relevance Judgments in Legal Case Retrieval. *ACM Transactions on Information Systems* 41, 3 (2023), 1–32.

[26] Vu Tran, Minh Le Nguyen, and Ken Satoh. 2019. Building legal case retrieval systems with lexical matching and summarization using a pre-trained phrase scoring model. In *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Law*. 275–282.

[27] Chaojun Xiao, Xueyu Hu, Zhiyuan Liu, Cunchao Tu, and Maosong Sun. 2021. Lawformer: A pre-trained language model for chinese legal long documents. *AI Open* 2 (2021), 79–84.

[28] Xiaohui Xie, Qian Dong, Bingning Wang, Feiyang Lv, Ting Yao, Weinan Gan, Zhijing Wu, Xiangsheng Li, Haitao Li, Yiqun Liu, et al. 2023. T2Ranking: A large-scale Chinese Benchmark for Passage Ranking. *arXiv preprint arXiv:2304.03679* (2023).

[29] Shenghao Yang, Haitao Li, Zhumin Chu, Jingtao Zhan, Yiqun Liu, Min Zhang, and Shaoping Ma. 2022. THUIR at the NTCIR-16 WWW-4 Task. *Proceedings of NTCIR-16. to appear* (2022).

[30] Weijie Yu, Zhongxiang Sun, Jun Xu, Zhenhua Dong, Xu Chen, Hongteng Xu, and Ji-Rong Wen. 2022. Explainable legal case matching via inverse optimal transport-based rationale extraction. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 657–668.

[31] ChengXiang Zhai. 2008. Statistical language models for information retrieval. *Synthesis lectures on human language technologies* 1, 1 (2008), 1–141.

[32] Jingtao Zhan, Jiaxin Mao, Yiqun Liu, Jiafeng Guo, Min Zhang, and Shaoping Ma. 2021. Optimizing dense retrieval model training with hard negatives. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1503–1512.

[33] Jingtao Zhan, Jiaxin Mao, Yiqun Liu, Min Zhang, and Shaoping Ma. 2020. RepBERT: Contextualized text embeddings for first-stage retrieval. *arXiv preprint arXiv:2006.15498* (2020).

CAPTAIN at COLIEE 2023: Efficient Methods for Legal Information Retrieval and Entailment Tasks

Chau Nguyen*
Phuong Nguyen*
Thanh Tran*
Dat Nguyen*
An Trieu*
Tin Pham
Anh Dang
Le-Minh Nguyen†
{chau.nguyen,phuongnm,thanhtc,nguyendt,antrieu,tinpham,hoanganhdang,nguyenml}@jaist.ac.jp
Japan Advanced Institute of Science and Technology
Ishikawa, Japan

ABSTRACT

The Competition on Legal Information Extraction/Entailment (COLIEE) is held annually to encourage advancements in the automatic processing of legal texts. Processing legal documents is challenging due to the intricate structure and meaning of legal language. In this paper, we outline our strategies for tackling Task 2, Task 3, and Task 4 in the COLIEE 2023 competition. Our approach involved utilizing appropriate state-of-the-art deep learning methods, designing methods based on domain characteristics observation, and applying meticulous engineering practices and methodologies to the competition. As a result, our performance in these tasks has been outstanding, with first places in Task 2 and Task 3, and promising results in Task 4.

CCS CONCEPTS

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

KEYWORDS

COLIEE competition, Legal Text Processing, CAPTAIN Team

ACM Reference Format:
Chau Nguyen, Phuong Nguyen, Thanh Tran, Dat Nguyen, An Trieu, Tin Pham, Anh Dang, and Le-Minh Nguyen. 2023. CAPTAIN at COLIEE 2023: Efficient Methods for Legal Information Retrieval and Entailment Tasks. In *Proceedings of COLIEE 2023 workshop, June 19, 2023, Braga, Portugal*. 10 pages.

*Authors contributed equally to the paper
†The corresponding Author

COLIEE 2023, June 19, 2023, Braga, Portugal

1 INTRODUCTION

The COLIEE competition [11] (Competition on Legal Information Extraction/Entailment) is an annual event that focuses on the automated processing of legal texts. The competition involves two types of data: case law and statute law. For each type of data, there are two tasks: legal information retrieval and legal information entailment.

Task 1 involves retrieving cases that support a given case law, which is a crucial aspect of legal practice, used by attorneys and courts in decision-making. Task 2 also uses case law data, but requires models to find paragraphs that entail the decision of a given case. Tasks 3 and 4 involve using statute law data and tackling challenges such as retrieval and entailment. It is worth noting that Task 1 can be considered as an initial step towards Task 2, and Task 3 can serve as a preliminary stage for Task 4. Nonetheless, it is not obligatory to perform Task 1 before Task 2 or to undertake Task 3 prior to Task 4.

This year, we participated in Task 2, Task 3, and Task 4. The CAPTAIN team has succeeded in achieving competitive results in the COLIEE 2023 competition by utilizing appropriate deep learning techniques, suitable methods based on domain characteristics, and robust engineering practices and methodologies.

In Task 2, to tackle the challenging task of legal case entailment in COLIEE 2023, we propose an approach based on the pre-trained MonoT5 sequence-to-sequence model, which is fine-tuned with hard negative mining and ensembling techniques. The approach utilizes a straightforward input template to represent the point-wise classification aspect of the model and captures the relevance score of candidate paragraphs using the probability of the "true" token (versus the "false" token). The ensembling stage involves hyperparameter searching to find the optimal weight for each checkpoint. The approach achieves state-of-the-art performance in Task 2 this year, demonstrating the effectiveness of the proposed techniques.

Our solution for Task 3 addresses the issue of *data diversity* across many legal categories (e.g., rights of retention, ownership, juridical persons). We observed that the queries and articles in the Civil Code include multiple categories, and the limited annotated data and lengthy pairs of questions and articles make it challenging

for deep learning models to identify general patterns, often leading to local optima. To overcome this issue, we focused on constructing sub-models to learn specific aspects of the legal domain and designing an ensemble method to combine these sub-models to create a generalized system. We assumed that each local optimum of the legal retrieval system is biased towards particular categories, whereas our goal is to develop a system that covers all categories and utilizes the strengths of all sub-models through ensemble methods.

In Task 4, we experimented with three approaches, which involve identifying the entailment relationship between legal articles and a query. The first approach is to fine-tune language models with a novel augmentation mechanism. The second approach is based on condition-statement extraction, which involves breaking down legal articles and the query into components, identifying the entailing relationship based on the matching between the condition part and the statement part of the article and the query. The third approach is a hybrid approach that uses SVM ensembled with the condition-statement extraction method.

The CAPTAIN team proposed deep learning methods for these tasks and conducted detailed experiments to evaluate their approaches. The proposed methods could serve as a useful reference for researchers and engineers in the field of automated legal document processing.

2 RELATED WORKS

2.1 Case Law

In COLIEE 2020, the Transformer model and its modified versions were widely used. TLIR [33] and JNLP [20] teams used them for Task 1 to classify candidate cases with two labels (support/non-support). Team cyber [7] encoded candidate cases and base cases in TfIdf space and used SVM to classify, which helped them obtain the top rank in Task 1. For Task 2, JNLP [20] utilized the same approach as in Task 1 for the weakly-labeled dataset, which resulted in outperforming team cyber [7] and winning Task 2.

In COLIEE 2021, teams used various BERT-based IR models to tackle Task 1, which is a case retrieval problem. TLIR team [15] achieved first place by applying an ensemble of multiple BERT models and selecting the highest result among them. Other teams also utilized BERT models and additional techniques such as text enrichment and similarity measures like Word Mover's Distance to improve their performance. In Task 2 of COLIEE 2021, the winning team NM [30] utilized several models including monoT5-zero-shot, monoT5, and DeBERTa, and evaluated an ensemble of monoT5 and DeBERTa models. TR team [31] used hand-crafted similarity features and a classical random forest classifier, while UA team [9] used BERT pre-trained on a large dataset and a language detection model to remove French text. Siat team [12] proposed a pre-training task on BERT with dynamic N-gram masking to obtain a BERT model with legal knowledge.

In COLIEE 2022, in Task 1, the winning team (UA [25]) used a Transformer-based model to generate paragraph embeddings, then calculated the similarity between paragraphs of the query and the positive and negative cases. They used a Gradient Boosting classifier to determine if those cases should be noticed or not. The other teams used a combination of traditional and neural-based techniques, such as document and passage-level retrieval, adding external domain knowledge by extracting statute fragments mentioned in the cases, and re-ranking based on different statistical and embedding models. In Task 2, the winning team (NM [29]) used monoT5, which is an adaptation of the T5 model. During inference, monoT5 generates a score that measures the relevance of a document to a query by applying a softmax function to the logits of the tokens "true" and "false". They also extended a zero-shot approach and fine-tuned T5-base and T5-3B models for legal question-answering. The other teams used a combination of LegalBERT, BM25, and knowledge representation techniques such as Abstract Meaning Representation (AMR) to capture the most important words in the query and corresponding candidate paragraphs.

2.2 Statute Law

The approaches used in Task 3 of COLIEE 2020 involved various methods such as TfIdf, BM25, BERT models, and different word embedding techniques. The teams used different combinations of these techniques to classify articles as relevant or not based on the given legal queries. The LLNTU team [32] achieved the highest performance using the BERT model for article classification. The approaches used in Task 4 involved various NLP techniques and models such as BERT [4], RoBERTa [14], GloVe [24], and LSTM [8]. The winning team, JNLP [20], employed BERT-based models fine-tuned with Japanese legal data and TfIdf to achieve the best performance. Other approaches also used rule-based ensembles, SVM [3], and attention mechanisms with word embeddings to tackle the legal text classification task.

In Task 3 of the COLIEE 2021 competition, The OvGU team [37] took first place by using a variety of BERT models and data enrichment techniques, with the best run using Sentence-BERT [27] embedding and TfIdf. The HUKB [38] system uses a BERT-based IR system with Indri [34] and constructs a new article database. JNLP team [17] uses multiple BERT models and an ensemble approach for generating results. TR team [31] uses Word Mover's Distance to calculate similarity and UA team [9] uses ordinary IR modules, with the best run using BM25. In Task 4, the winning approach by HUKB [38] utilized an ensemble of BERT models with data augmentation. JNLP team [19] used a bert-base-japanese-whole-word-masking model with TfIdf-based data augmentation and proposed Next Foreign Sentence Prediction (NFSP) and Neighbor Multilingual Sentence Prediction (NMSP) for Task 4. OvGU [37] employed an ensemble of graph neural networks, where each node represented a query or article, and TR used existing models, including T5-based [26] ensemble, Multee [35], and Electra [2]. UA team [9] utilized BERT with semantic information. KIS team [5] extended their previous work using predicate-argument structure analysis, legal dictionary, negation detection, and an ensemble of modules for explainability.

In the COLIEE 2022 Task 3, the top-performing teams used various methods for answering legal questions. HUKB [39] and OvGU [36] utilized information retrieval (IR) models with different variations in terms of document databases and sentence embeddings. JNLP [1] proposed two separate deep learning models for answering ordinal questions and use-case questions, while LLNTU [13] used a BERT-based approach with no clear adjustments for the current year's task. Finally, UA [25] relied on IR models, specifically TfIdf and BM25. In Task 4, HUKB [39] proposed a method for selecting

relevant parts from articles and employed an ensemble of BERT with data augmentation. JNLP [1] compared different pre-trained models and data augmentation techniques. KIS [6] employed an ensemble of rule-based and BERT-based methods with data augmentation and person name inference. LLNTU [13] used the longest uncommon subsequence similarity comparison model, while OvGU [36] employed graph neural networks on both the queries and articles, and additionally incorporating textbook knowledge.

3 METHODS

3.1 Task 2. Case Law Entailment Task

The second task of COLIEE 2023 is legal case entailment: Given a fragment of a base case and a set of potential judicial-decision-related paragraphs of past cases, determine which candidate paragraphs entail the fragment of the base case. The dataset for this year's competition consists of 625 training cases with entailment labels and 100 test cases. The training set has an average ratio of 35-to-1 candidates per case and 1 entailment paragraph per case. As the ratio of candidates per case is high and the size of the dataset is limited, the semantic and structural complexity of legal documents makes this a particularly challenging task.

In order to overcome the challenge of this task, we utilize the pre-trained MonoT5 [23] sequence-to-sequence models on the MS MARCO [21] passage ranking dataset. MonoT5 is a novel approach to document ranking by fine-tuning the pre-trained T5 models with modified training data for the point-wise classification task, which outperformed the traditional BERT re-ranker approach. The process to convert a sequence-to-sequence T5 model to a point-wise classification MonoT5 model is straightforward with the following input template:

"Query: [CASE] Document: [CANDIDATE] Relevant:"

The point-wise aspect of the model is represented by taking the probability of 2 tokens "true" and "false" of the decoded outputs. The relevance score of the candidate to the fragment of the base case is captured by the probability of the "true" token after normalization by the softmax function. After obtaining the relevance score of each candidate, the scores are sorted to form the final ranking list.

Although MonoT5 has been used in past competitions [28], our approach introduces new engineering techniques in order to improve the performance of pre-trained checkpoints on legal entailment tasks, which we describe in the following sections. Figure 1 provides an overview of our approach for Task 2.

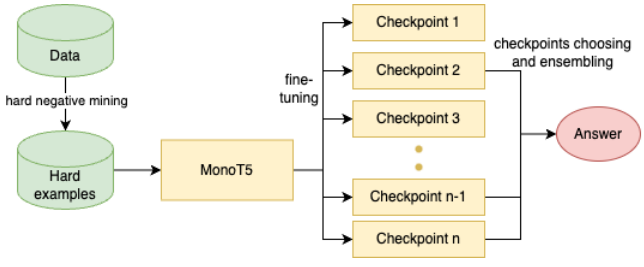


Figure 1: Our approach for Task 2

3.1.1 Fine-tuning with hard negative mining. To form the training dataset for fine-tuning, we split the original dataset into training/validation/test segments and use the validation set to select the best checkpoints for later stages. We select the pre-trained MonoT5-large checkpoint on MS MARCO [21] as our initial weight for fine-tuning. Note that the pre-trained MonoT5-3B checkpoints are more powerful and outperformed the MonoT5-large variant on MS MARCO but unfortunately, we are not able to fine-tune these weights due to their high computation cost. We initialize the MonoT5-large model with the pre-trained weights and perform hard negative mining as follows. Firstly, we sample the top 10 negative examples of each base case's fragment (i.e., the top 10 fragments with a negative label that are most similar to the base case's fragment), sorted by the retrieval scores with BM25 to form the first training set and perform fine-tuning. After that, we re-sample the top 10 negative examples from the fine-tuned model from the previous step to form the final training set and fine-tune the original pre-trained weights with this dataset. This procedure has the purpose of mining the hard examples to further push the performance of the pre-trained models as these models have already achieved good performance in past competitions. After fine-tuning, we keep the 5 best checkpoints ordered by the metric score on the validation set for the next ensembling pipeline.

3.1.2 Ensembling. We perform ensembling the best checkpoints with hyperparameter searching on the validation set to get the ensembled score for each candidate. Each checkpoint is given an initial weight range from $[0, 1]$ and the final weight is calculated by normalizing the initial weight with the sum of all initial weights. We use grid search to find the optimal weight for each checkpoint and save the ensembled scores for the final stage.

3.1.3 Prediction. After obtaining the candidate's scores for each base case, we sort the candidate list by score and select the top k candidates. We keep the candidate with the highest score and for each remaining top candidate, we include the candidate in the final prediction if the difference between their relevance score to the highest score is smaller than a pre-defined margin m. We perform a grid search on the values of k and m on the validation set to select the best values.

3.2 Task 3. The Statute Law Retrieval Task

Problem statement. The objective of Task 3 in this competition is to extract a subset of Japanese Civil Code Articles that are relevant to a given legal bar exam question, denoted by Q. Participants are required to select articles, denoted by A_i $(1 \leq i \leq n)$, that are related to the given question Q. This task serves as a precursor to Task 4, which requires participants to infer the legality of Q based on the relevant legal articles.

3.2.1 Approach overview. Following the previous approach [17, 20], we also utilize the power of the pre-trained language model and some negative sampling techniques to fine-tune a model which can retrieve the relevant articles from a legal corpus. In detail, given an input query, top k relevant articles (e.g., $k = 150$ following [17]) are chosen based on term matching (e.g., TfIdf or BM25). Then, these articles are paired with the query and fed to a pre-trained language model to find the relevant articles.

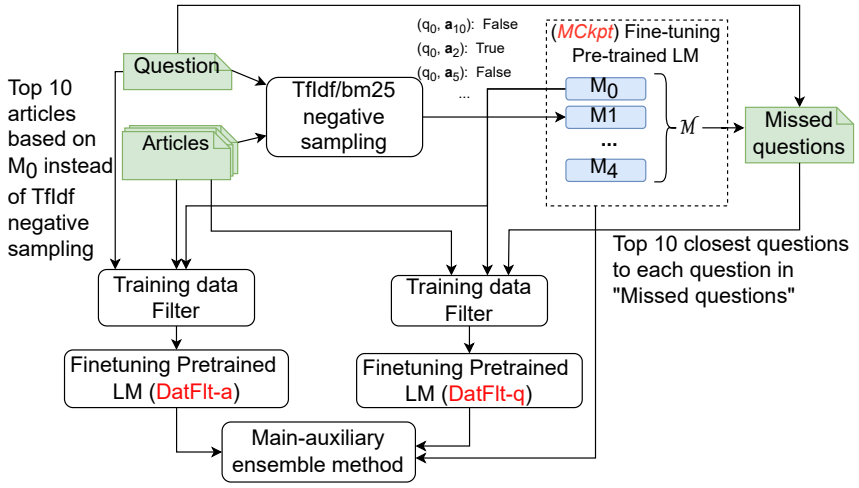


Figure 2: Our approach for Task 3

3.2.2 Approach details. As mentioned before, this year, our solution is focused on the problem of *data diversity* in many categories. Based on our observation, we found that the queries and articles in the Civil Code include many categories. In Figure 3, we show the different performance between checkpoints in the training process (which are local optimums). This result showed a large margin among the checkpoint's performance.

Therefore, in this work, we also introduce a simple yet effective method to ensemble model checkpoints in local optima to make a generalized system finally. We assume that each local optimum (fine-tuned model found in section 3.2.1) is biased to some categories, while our target is to build a system that can explore all the categories via sub-models and aggregate the strengths of these sub-models. To this end, we introduce some approaches to construct the sub-models to learn different aspects and design an effective way to combine them (Figure 2).

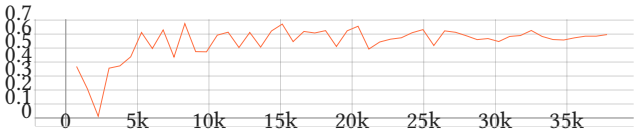


Figure 3: Model performance (macro F2) on the development set with five epochs (about 35K steps).

Model checkpoints (MCkpt). As we mentioned, we assume that each checkpoint in the training process (Figure 3) is biased to some categories. Therefore, we combine the top h best model checkpoints ($\mathcal{M} = \{M_i | 0 \leq i < h\}$) to collect these best states of a pre-trained language model based on the development set.

Pre-trained language models (Plm). We hypothesize that using different pre-trained language models also supports the overall system to cover different aspects of data. Therefore, we design the combination method using two pre-trained language models *cl-tohoku/bert-base-japanese-whole-word-masking* and *monot5-large-msmarco-10k*.

Data filter (DatFlt). In this approach, we aim to let the model learn within different data distributions to strengthen the robustness of the sub-model. To this end, we keep the original validation set and construct a new training set by two following strategies:

- *Tackling missed **q**uery (DatFlt-q)*: By using the fine-tuned mechanism described in section 3.2.1, we cached top h model checkpoints in the training process (setting *MCkpt* in Figure 2, $\mathcal{M} = \{M_{i|0 \leq i < h}\}$). We found a set of questions that are typically *"missed"* by these model checkpoints (i.e., the questions in which the fine-tuned models, M_i, did not find any relevant articles in the legal corpus). For each missed question, we filter the top 10 nearest questions to them in the training set (using cosine similarity of BERT `[CLS]` vector) to construct a new training dataset. In particular, the new training set related to these questions is generated and we use it to continue to train with the best model checkpoint (M_0). To this end, we expect that the new training set has a similar topic/category to the *missed* queries, which makes the model M_0 learn different features.
- *Highly relevant **a**rticles in negative sampling (DatFlt-a)*: In difference with the previous approach, this approach constructs a new training data set by filtering the highly relevant articles based on a fine-tuned model (e.g., top 10 nearest articles based on M_0) for the negative sampling step instead of the Tfldf scores. We expect this bootstrapping mechanism to boost the model and make it easier to separate similar articles in the legal corpus.

Main-auxiliary ensemble method. Based on our experiments, we found that the overall system's performance will be hurt if we unite all the articles of all sub-models found in the previous step (because this standard combination method decreases the precision for each question sample). Therefore, we choose one sub-model (best performance based on the development set, M_0) as the main model, and we consider other sub-models as auxiliary models. Then we filter *missed* questions of this model and union corresponding

relevant articles of these questions from auxiliary models for the final result.

3.3 Task 4. The Legal Textual Entailment Task

We propose three approaches to tackle Task 4 of the COLIEE competition, which focuses on legal textual entailment. The objective of this task is to identify the entailment relationship between legal articles and a given query, where the answer is binary: "YES" (if the article entails the query) or "NO" (if the article does not entail the query).

3.3.1 First approach: Online data augmentation. In the previous work [20], the authors proposed a mechanism to augment training data and fine-tune a pre-trained model in the legal domain. In line with this work, we introduce a mechanism named *online data augmentation*, which re-samples the training data based on a pre-trained masked-language model (e.g., *cl-tohoku/bert-base-japanese-whole-word-masking*). We argue that the main challenge in this task is the sparse annotated data distribution which is hard for the fine-tuning process. While in Task 3, each question can consider up to 100 or 150 relevant articles for negative samples, in this task, the number of relevant articles is usually only one or two. Therefore, the augmentation data approach is typically an appropriate solution to create a generalized system.

In terms of our data augmentation mechanism, given a question input, we used a pre-trained language model with the task recovery masked words [4] to generate similar questions. In particular, in the training process, we randomly masked some words in a question and replaced them with a random selection from the top k highest probabilities of candidate words output from a pre-trained language model. The generated question is then paired with the summaries of given relevant articles, and the generated pair is used for the fine-tuning process.

3.3.2 Second approach: Condition-statement extraction. The method relies on the notion that a legal article can consist of multiple sentences, each of which discusses a particular statement subject to certain conditions. Additionally, a statement may have an exception if certain other conditions are met. An instance of this would be the phrase following "provided, however" in Table 1, which represents the "other conditions" that contradict the primary statement "the person's residence is deemed to be the person's domicile". In particular, we break down a legal article into multiple components, where each component consists of two parts: the condition part and the statement part. Similarly, we break down the query into two parts (condition and statement). Our approach involves determining the entailing relationship based on the "matching" (or entailing) between the condition part of the article and the query, as well as the statement part of the article and the query. We use a Semantic Role Labeling (SRL) model and heuristics to identify the relevant components in the legal articles and the queries, and then we employ a matching algorithm to determine the entailment relationship. Our approach shares similarities with the methods proposed in [22] and [10]. Nevertheless, our approach, which employs SRL to detect condition-statement pairs, has the potential to provide accurate and efficient results, especially when dealing with complex legal texts, which often contain lengthy and intricate sentences and structures. The overview of the approach is described in Figure 4.

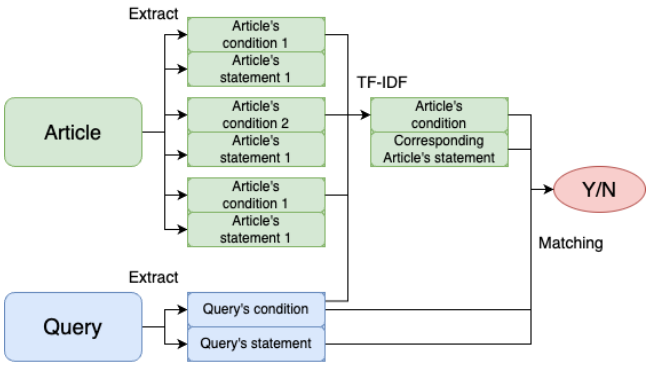


Figure 4: The condition-statement approach for Task 4

Extract (condition, statement) pairs from an article. The approach involves extracting (condition, statement) pairs from each point in an article, which usually consists of one sentence. To achieve this, the article is first split into sentences using the nltk library[1]. If a sentence does not have an exception, a Semantic Role Labeling technique is applied to identify the main verb, its arguments, and other components. Two heuristics are used: (i) the main verb is identified as the verb with the SRL labeling that covers most of the words in the input sentence, and (ii) the statement is extracted from the sentence where the first token is the first token of the main verb or argument, and the last token is the last token of the main verb or argument. The condition is then constructed by concatenating the remaining parts of the sentence that do not belong to the statement (see Pair 1 in Table 1). In case the sentence has an exception (indicated by the phrase "provided, however"), we split it into two parts. The part before the phrase "provided, however" is processed as described above. The part after the phrase "provided, however" will be processed to extract a condition or statement, and will be negated (i.e., add the word "not" following [18]) if necessary (see Pair 2 and 3 in Table 1). For the query, we only consider the last sentence to extract the statement, and we consider other parts as the condition.

Inference of YES/NO answer To begin with, we employ TfIdf to select a condition from the extracted list of conditions that closely corresponds to the query's condition. After that, we evaluate the similarity between the retrieved condition and the query's condition, as well as the similarity between the corresponding statements. This is done by checking whether the number of negations in both the retrieved and the query's conditions and statements is odd or even. If both the condition and statement match, we answer YES, otherwise, the answer is NO.

3.3.3 Third approach: SVM ensembles with condition-statement extraction. This method involves using SVM on the data to generate a series of YES/NO predictions, which are then combined with predictions from the condition-statement extraction approach. To

[1] https://github.com/nltk/nltk

Article	Article 23 (1) If a person's domicile is unknown, the person's residence is deemed to be the person's domicile. (2) If a person does not have a domicile in Japan, the person's residence is deemed to be the person's domicile, regardless of whether the person is a Japanese national or a foreign national; *provided, however, that this does not apply if the law of domicile is to be applied in accordance with the provisions of the laws that establish the governing law.*
Pair 1	**Condition:** If a person's domicile is unknown
	Statement: the person's residence is deemed to be the person 's domicile
Pair 2	**Condition:** If a person does not have a domicile in Japan and regardless of whether the person is a Japanese national or a foreign national
	Statement: the person's residence is deemed to be the person 's domicile
Pair 3	**Condition:** If a person does not have a domicile in Japan and regardless of whether the person is a Japanese national or a foreign national and if the law of domicile is to be applied in accordance with the provisions of the laws that establish the governing law
	Statement: the person's residence is **not** deemed to be the person 's domicile

Table 1: Example of (condition, statement) pairs extracted from a legal article. Pair 1 is extracted from the sentence in point (1), pairs 2 and 3 are extracted from the sentence in point (2).

achieve this, we adopt an ensemble method that involves identifying whether a query is a specific-scenario query or a general query, as described in [16]. A specific-scenario query, as opposed to a general query, is a type of query that describes a particular situation or scenario in real life, requiring the ability to identify relevant abstract articles based on understanding the scenario. Specific-scenario queries are detected by identifying uppercase characters used to refer to legal persons or objects in the queries, following a consistent template observed in legal bar queries.

We have found that the condition-statement extraction method tends to perform well with general queries, while SVM is better suited for specific-scenario queries. Consequently, when the two approaches produce different answers for a given query, we determine the final answer based on the query type: if it is a specific-scenario query, the final answer follows SVM's prediction, while for general queries, the final answer follows the condition-statement extraction method's prediction.

4 EXPERIMENTS

In this section, we describe in detail the result of our methods introduced in Section 3 in the development process and the final result from the COLIEE committee.

4.1 Task 2. Case Law Processing

4.1.1 Dataset and baselines. The statistics of the dataset provided by COLIEE competition are shown in Table 2. We compare our approach against the following baselines:

- BM25: We use the BM25 Lucene implementation in the Anserini[2] open-source toolkit with the default parameters setting. We select the top k candidates as the prediction with k selected by grid search on the validation set.
- PT MonoT5-[large/3B]: We use the pre-trained MonoT5-large[3] and MonoT5-3B [4] checkpoints as zero-shot baselines.
- PT BERT-large: We use a pre-trained BERT-large re-ranker[5] on MS MACRO as an alternative approach to MonoT5. We use the implementation and the checkpoints released by the authors.

	Train	Validation	Test
Size	525	100	100
Case ID range	001 - 525	526 - 625	626 - 725
Candidates / Case Ratio	35.31	32.5	37.4
Entailments / Case Ratio	1.17	1.18	1.2

Table 2: Dataset statistics

4.1.2 Submissions. Our submissions are described below:

- FT MonoT5-large RS (Run name: `mt5l-e2`): The fine-tuned MonoT5-large using random negative sampling strategy.
- FT MonoT5-large HS (Run name: `mt5l-ed`): The fine-tuned MonoT5-large using our hard negative sampling strategy.
- Ensemble (Run name: `mt5l-ed4`): The ensembled top 5 checkpoints using the procedure in 3.1.2.

Except for BM25, we generate the prediction for all the baselines and submissions using the procedure in 3.1.3.

4.1.3 Experimental results. Table 3 shows the validation F1 score of our methods compared with the baselines. Table 4 shows the results of the teams participating in the competition. It can be seen that our methods for this task outperform others by a considerable gap in the F1 score, indicating the appropriateness of our formulation for Task 2.

Method	**Validation F1**
Baselines	
+ BM25	61.47
+ PT MonoT5-large	68.62
+ PT MonoT5-3B	68.31
+ PT BERT-large	53.21
Our methods	
+ FT MonoT5-large RS (`mt5l-e2`)	75.23
+ FT MonoT5-large HS (`mt5l-ed`)	79.29
+ Ensemble (`mt5l-ed4`)	**80.18**

Table 3: Our methods in Task 2 compared with baselines

[2] https://github.com/castorini/pyserini
[3] castorini/monot5-large-msmarco
[4] castorini/monot5-3b-msmarco
[5] Luyu/bert-base-mdoc-bm25

Run	F1 (%)	Precision (%)	Recall (%)
CAPTAIN.mt5l-ed	**74.56**	78.70	**70.83**
CAPTAIN.mt5l-ed4	72.65	78.64	67.50
THUIR.thuir-monot5	71.82	**79.00**	65.83
CAPTAIN.mt5l-e2	70.54	75.96	65.83
THUIR.thuir-ensemble_2	69.30	73.15	65.83
JNLP.bm_cl_1_pr_1	68.18	75.00	62.50
IITDLI.iitdli_task2_run2	67.27	74.00	61.67
JNLP.cl_1_pr_1	65.45	72.00	60.00
UONLP.test_no_labels	63.87	64.41	63.33
THUIR.thuir-ensemble	60.91	67.00	55.83
NOWJ.non-empty	60.79	64.49	57.50
NOWJ.hp	60.36	65.69	55.83
IITDLI.iitdli_task2_run3	53.04	55.45	50.83
LLNTU.task2_llntukwnic	18.18	20.00	16.67

Table 4: Results of Task 2 in the competition

By looking at the F1 results, we can see that the fine-tuned MonoT5 on the legal entailment dataset outperforms the pre-trained MonoT5 on MS MARCO [21] by a significant margin. The introduced hard negative mining procedure proved to be a more effective method for fine-tuning than the vanilla random sampling strategy. The ensemble approach has the best performance on the validation set but is inferior to a single model fine-tuned MonoT5 on the test set, which is likely the result of overfitting when performing hyperparameter grid searching. And finally, our results show that the sequence-to-sequence approach has better performance in textual entailment tasks compared to the BERT re-ranker approach.

4.2 Task 3. The Statute Law Retrieval Task

4.2.1 Dataset. Similar to the previous year's format, the dataset of this task consists of 996 questions, a legal corpus (Civil Code) with 768 articles, and 1272 pairs of questions and relevant articles (positive samples). The examples of this dataset are shown in Table 1. For the development process, we choose questions that have an ID starting with `R02` (81 questions) or `R03` (109 questions) as a validation set and conduct model/settings evaluations on this subset.

4.2.2 Submissions. We describe three settings we submitted in the COLIEE Task-3 submission time.

- `CAPTAIN.bjpAll`: We ensemble all the sub-models from the pre-trained language model *cl-tohoku/bert-base-japanese-whole-word-masking*. This submission is the ensemble result of five checkpoints in *MCkpt* setting, five checkpoints in *DatFlt-q* setting, and the best checkpoint in *DatFlt-a* setting.
- `CAPTAIN.allEnssMissq`: We ensemble the sub-models from the pre-trained language model *cl-tohoku/bert-base-japanese-whole-word-masking* and *monot5-large-msmarco-10k*. This submission is the ensemble result of the best checkpoint in *MCkpt* setting and *monot5-large-msmarco-10k* fine-tuned model.
- `CAPTAIN.allEnssBoostraping`: We ensemble the sub-models from the pre-trained language model *cl-tohoku/bert-base-japanese-whole-word-masking* and *monot5-large-msmarco-10k*. This submission is the ensemble result of the best checkpoint in *MCkpt-a* setting and *monot5-large-msmarco-10k* fine-tuned model.

4.2.3 Experiments and Results. We conducted experiments to estimate the effectiveness of our proposed methods based on the result of the development set and compared it with the best result [39] of COLIEE 2022. Especially for a fair comparison with their result, we only optimize model hyper-parameters based on the `R02` sub-set and evaluate the result on `R03` sub-set. The best models found in the development process are used to predict the output of the official test set `R04`.

Hyper-parameters. In the fine-tuning process, we used five epochs, the learning rate is selected in $\{1e^{-5}, 2e^{-5}\}$, the maximum token length is 512 sub-word tokens and batch size is selected in $\{16, 32\}$. The top 5 best model checkpoints are saved based on the macro F2 scores of development set `R02`.

Development results. Our main development results are shown in Table 5. Firstly, we develop our system based on two main pre-trained language models: monoT5 versions *castorini/monot5-large-msmarco* [6], *castorini/monot5-large-msmarco-10k*[7] and Japanese pre-trained language model *cl-tohoku/bert-base-japanese-whole-word-masking*[8]:

- *monoT5 pre-trained model*: This pre-trained model is specialized for text retrieval tasks trained on the MACRO large-scaled dataset. Based on the suggestion of the authors [23], we used version *castorini/monot5-large-msmarco-10k* for zero-shot learning and ensemble with version *castorini/ monot5-large-msmarco* fine-tuned on the COLIEE dataset. The result of the development set proved that incorporating two versions boosts the performance by about 2% F2 score (Table 5). Therefore, we used the ensemble result of this model as the main prediction when incorporated with other settings.
- *Japanese pre-trained model*: We mainly evaluate our proposed method on a pre-trained language model (*cl-tohoku/bert-base-japanese-whole-word-masking*) because this model did not learn any text retrieval task. This makes it straightforward to measure the effectiveness of our method on legal text retrieval tasks (two last rows in Table 5). In the comparison between three settings *DatFlt-q*, *DatFlt-a*, *MCkpt* separately, we found that *DatFlt-q* setting shows the strength in the Recall metric but low on the Precision. Since this setting is strengthened in *"missed"* questions, which makes the model reduces the number of *missed* questions as much as possible. In contrast to setting *DatFlt-a*, which prefers to return the accurate articles that make the Precision typically higher than *DatFlt-q*.
- *Ensemble model*: We combine the result of monoT5 with Japanese pre-trained models by the *main-auxiliary ensemble method* (described in section 3.2) with monoT5 play as the

[6] https://huggingface.co/castorini/monot5-large-msmarco
[7] https://huggingface.co/castorini/monot5-large-msmarco-10k
[8] https://huggingface.co/cl-tohoku/bert-base-japanese-whole-word-masking

main model. Besides, we also combine all the sub-models of our proposed methods for comparison (last rows in Table 5). These results show that the effective incorporation between our methods and monoT5 achieved significant improvement in the overall system.

Run	F2 (%)	P (%)	R (%)
Evaluate on R02 questions			
monoT5-L-10k (zs)	68.29	68.93	69.14
monoT5-L (ft)	68.29	68.93	69.14
monoT5-L (ft) + monoT5-L-10k (zs)	70.33	71.81	70.99
DatFlt-q	69.35	53.81	**80.86**
DatFlt-a	69.08	58.09	77.78
MCkpt	68.38	61.21	72.84
DatFlt-q + monoT5 (allEnssMissq)	**76.36**	**74.28**	78.40
DatFlt-a + monoT5 (allEnssBoostraping)	76.29	73.66	78.40
DatFlt-q + DatFlt-a + MCkpt (bjpAll)	70.45	63.37	75.31
Evaluate on R03 questions			
HUKB2 [39]	82.04	81.80	84.05
monoT5-L-10k (zs)	79.13	82.19	79.62
monoT5-L (ft)	80.05	83.10	80.54
monoT5-L (ft) + monoT5-L-10k (zs)	81.28	81.61	83.29
DatFlt-q	75.84	59.06	**88.84**
DatFlt-a	76.90	62.58	87.92
MCkpt	80.68	76.07	85.06
DatFlt-q + monoT5 (allEnssMissq)	**85.52**	84.56	87.45
DatFlt-a + monoT5 (allEnssBoostraping)	85.01	**85.47**	86.35
DatFlt-q + DatFlt-a + MCkpt (bjpAll)	82.15	75.84	87.81

Table 5: Results of Task 3 on the development set. The notations (ft), (zs) refer to *fine-tuned* on the COLIEE dataset and *zero-shot* learning, respectively.

Official test results. Table 6 shows the full test results of Task 3 provided by the COLIEE organization. Our results got first and second places and also competitive in third place from other teams. The results demonstrate that our proposed methodology exhibits a high degree of generalizability and holds significant promise for applications in text retrieval tasks. Besides, we also got the best performance on most other metrics, such as MAP, R@5, R@10, and R@30, which is clear evidence of the effectiveness of our method. In addition, in the two settings `allEnssMissq` and `allEnssBoostraping`, because we combine two different pre-trained models with monoT5 model playing as a primary model for prediction, the evaluation scores on four metrics (MAP, R@5, R@10, and R@30) are exactly same. In the submission `bjpAll`, (equal to *DatFlt-q + DatFlt-a + MCkpt*), the ranking scores of our proposed method still work effectively because all the models of these settings are the different checkpoints of the same pre-trained model in the fine-tuning process, which share the same semantic space.

4.3 Task 4. The Legal Textual Entailment Task

4.3.1 Dataset. The dataset we use for experiments in Task 4 is similar to the dataset used for Task 3, but for the task of textual entailment. It contains 996 questions, and a legal corpus (Civil Code) with 768 articles. For the validating and testing step, we used the questions in R01 (111 questions), R02 (81 questions), and R03

Run	F2	P	R	MAP	R5	R10	R30
CAPTAIN.Missq	**75.69**	**72.61**	79.21	69.21	75.38	83.85	88.46
CAPTAIN.Boost	74.70	71.62	78.22	69.21	75.38	83.85	88.46
JNLP3	74.51	64.52	**82.18**	70.99	80.00	83.85	90.00
CAPTAIN.bjpAll	74.15	70.63	77.72	**84.64**	**87.69**	**90.77**	**96.15**
NOWJ.ensemble	72.73	68.23	76.73	78.99	78.46	80.77	89.23
HUKB1	67.25	62.79	70.79	73.97	75.38	83.85	93.08
JNLP2	66.28	64.22	70.30	68.61	73.08	79.23	88.46
HUKB3	66.25	65.02	68.32	74.14	74.62	84.62	93.08
JNLP1	65.71	66.50	67.82	68.65	73.08	79.23	88.46
LLNTUgigo	65.35	73.27	64.36	76.43	80.00	88.46	91.54
HUKB2	64.85	67.82	65.84	74.14	74.62	84.62	93.08
LLNTUkiwiord	63.26	70.30	62.38	76.25	82.31	86.92	93.08
UA.Tfldf_threshold2	56.42	62.05	56.44	65.51	66.92	79.23	84.62
UA.Tfldf_threshold1	55.45	63.37	54.46	65.51	66.92	79.23	84.62
UA.BM25	55.01	63.37	53.96	64.86	66.92	79.23	86.92
NOWJ.ttlJP	06.37	05.78	06.93	07.16	06.15	07.69	13.08
NOWJ.COENJP	01.82	01.49	01.98	02.46	01.54	03.85	06.15

Table 6: Results on the official test of Task 3. "CAPTAIN.Missq" stands for CAPTAIN.`allEnssMissq`, "CAPTAIN.Boost" stands for CAPTAIN.`allEnssBoostraping`.

(106 questions). The remaining questions are used for the training process.

4.3.2 Submissions. In Task 4, we submit 3 runs as follows:

- CAPTAIN.gen: The predictions of the first approach (*online data augmentation*)
- CAPTAIN.run1: The predictions of the second approach (condition-statement extraction)
- CAPTAIN.run2: The predictions of the third approach (SVM ensembles with condition-statement extraction)

4.3.3 Experiments and results. For the first approach experiments, we use the pre-trained model *cl-tohoku/bert-base-japanese-whole-word-masking* to re-sample the training data. The training data, in this case, involves remaining data and augmented data generated randomly by *online data augmentation* mechanism. For the second and the third approaches, we only use the original data.

The results of our approaches for Task 4 with the validation data are shown in Table 7. The results of Task 4 in the competition are shown in Table 8.

Method	R01	R02	R03
CAPTAIN.run1	60.36	50.62	64.22
CAPTAIN.run2	60.36	51.85	66.05
CAPTAIN.gen	58.56	67.90	56.88

Table 7: Performance (accuracy) of Task 4 validation sets: R01, R02, R03

Through the results, we can see that the approach using the *online data augmentation* mechanism has stable results on all testing and validating datasets. This shows that the data generated through *online data augmentation* mechanism provided the model useful information to decide the final result. On the other side, extracting condition, and statement pairs from articles also gives positive results on the validation dataset. The performance of the *condition and statement extracting* approach is close to the performance of

Run	Accuracy (%)
JNLP3	**78.22**
TRLABS_D	78.22
KIS2	69.31
UA-V2	66.34
AMHR01	65.35
LLNTUdulcsL	62.38
HUKB2	59.41
CAPTAIN.gen	58.42
CAPTAIN.run1	57.43
NOWJ.multi-v1-jp	54.46
CAPTAIN.run2	52.48
NOWJ.multijp	52.48
NOWJ.multi-v1-en	48.51

Table 8: Results of Task 4 in the competition

online data augmentation. However, when combined with the SVM model, the performance of the *condition and statement extracting* approach is decreased as compared to the results in the validating dataset. The SVM model seems to be biased with the validation dataset, since that causes the reduction of performance on official testing questions.

5 CONCLUSIONS

This paper describes our methods for addressing Task 2, Task 3, and Task 4 in COLIEE 2023. We have utilized appropriate deep learning techniques and applied rigorous engineering practices and methodologies to the competition, resulting in exceptional performance in these tasks. Going forward, we plan to further explore the properties of legal documents and queries to gain deeper insights and develop more effective techniques.

ACKNOWLEDGMENTS

This work is partly supported by AOARD grant FA23862214039. We express our gratitude to the reviewers for their valuable and constructive feedback.

REFERENCES

[1] Minh-Quan Bui, Chau Nguyen, Dinh-Truong Do, Nguyen-Khang Le, Dieu-Hien Nguyen, Thi-Thu-Trang Nguyen, Minh-Phuong Nguyen, and Le-Minh Nguyen. 2022. Using deep learning approaches for tackling legal's challenges (COLIEE 2022). In *Sixteenth International Workshop on Juris-informatics (JURISIN)*.

[2] Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators. *CoRR* abs/2003.10555 (2020). arXiv:2003.10555 https://arxiv.org/abs/2003.10555

[3] Corinna Cortes and Vladimir Vapnik. 1995. Support-vector networks. *Machine learning* 20 (1995), 273–297.

[4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *CoRR* abs/1810.04805 (2018). arXiv:1810.04805 http://arxiv.org/abs/1810.04805

[5] Masaki Fujita, Naoki Kiyota, and Yoshinobu Kano. 2021. Predicate's argument resolver and entity abstraction for legal question answering: Kis teams at coliee 2021 shared task. In *Proceedings of the COLIEE Workshop in ICAIL*.

[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: JSAI-isAI 2022 Workshop, JURISIN 2022, and JSAI 2022 International Session, Kyoto, Japan, June 12–17, 2022, Revised Selected Papers*. Springer, 138–153.

[7] Westermann Hannes, Savelka Jaromir, and Benyekhlef Karim. 2020. Paragraph Similarity Scoring and Fine-Tuned BERT for Legal Information Retrieval and Entailment. *COLIEE 2020* (2020).

[8] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.

[9] 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*.

[10] Mi-Young Kim, Juliano Rabelo, and Randy Goebel. 2019. Statute law information retrieval and entailment. In *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Law*. 283–289.

[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*. Springer, 51–67.

[12] J Li, X Zhao, J Liu, J Wen, and M Yang. 2021. Siat@ coliee-2021: Combining statistics recall and semantic ranking for legal case retrieval and entailment. In *Proceedings of the COLIEE Workshop in ICAIL*.

[13] M. Lin, S.C. Huang, and H.L. Shao. 2022. Rethinking attention: An attempting on revaluing attention weight with disjunctive union of longest uncommon subsequence for legal queries answering. I. *Proceedings of the Sixteenth International Workshop on Juris-informatics (JURISIN 2022)* (2022).

[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 preprint arXiv:1907.11692* (2019).

[15] Yixiao Ma, Yunqiu Shao, Bulou Liu, Yiqun Liu, Min Zhang, and Shaoping Ma. 2021. Retrieving legal cases from a large-scale candidate corpus. *Proceedings of the Eighth International Competition on Legal Information Extraction/Entailment, COLIEE2021* (2021).

[16] Chau Nguyen, Nguyen-Khang Le, Dieu-Hien Nguyen, Phuong Nguyen, and Le-Minh Nguyen. 2022. A Legal Information Retrieval System for Statute Law. In *Recent Challenges in Intelligent Information and Database Systems: 14th Asian Conference, ACIIDS 2022, Ho Chi Minh City, Vietnam, November 28-30, 2022, Proceedings*. Springer, 370–382.

[17] Ha-Thanh Nguyen, Phuong Minh Nguyen, Thi-Hai-Yen Vuong, Quan Minh Bui, Chau Minh Nguyen, Binh Tran Dang, Vu Tran, Minh Le Nguyen, and Ken Satoh. 2021. Jnlp team: Deep learning approaches for legal processing tasks in coliee 2021. *arXiv preprint arXiv:2106.13405* (2021).

[18] Ha-Thanh Nguyen, Vu Tran, Phuong Minh Nguyen, Quan Minh Bui, Chau Minh Nguyen, Binh Tran Dang, Hai-Yen Thi Vuong, Ken Satoh, and Minh Le Nguyen. 2021. ParaLaw Nets - Cross-lingual Sentence-level Pretraining for Legal Text Processing.

[19] Ha-Thanh Nguyen, Vu Tran, Phuong Minh Nguyen, Thi-Hai-Yen Vuong, Quan Minh Bui, Chau Minh Nguyen, Binh Tran Dang, Minh Le Nguyen, and Ken Satoh. 2021. ParaLaw Nets–Cross-lingual Sentence-level Pretraining for Legal Text Processing. *Proceedings of the COLIEE Workshop in ICAIL (2021)* (2021).

[20] Ha-Thanh Nguyen, Hai-Yen Thi Vuong, Phuong Minh Nguyen, Binh Tran Dang, Quan Minh Bui, Sinh Trong Vu, Chau Minh Nguyen, Vu Tran, Ken Satoh, and Minh Le Nguyen. 2020. JNLP Team: Deep Learning for Legal Processing in COLIEE 2020. *COLIEE 2020* (2020).

[21] 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. *choice* 2640 (2016), 660.

[22] Truong-Son Nguyen, Le-Minh Nguyen, Satoshi Tojo, Ken Satoh, and Akira Shimazu. 2018. Recurrent neural network-based models for recognizing requisite and effectuation parts in legal texts. *Artificial Intelligence and Law* 26 (2018), 169–199.

[23] Rodrigo Nogueira, Zhiying Jiang, Ronak Pradeep, and Jimmy Lin. 2020. Document Ranking with a Pretrained Sequence-to-Sequence Model. In *Findings of the Association for Computational Linguistics: EMNLP 2020*. Association for Computational Linguistics, Online, 708–718. https://doi.org/10.18653/v1/2020.findings-emnlp.63

[24] Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. 1532–1543.

[25] Juliano Rabelo, Mi-Young Kim, and Randy Goebel. 2023. Semantic-based classification of relevant case law. 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, 84–95.

[26] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research* 21, 1 (2020), 5485–5551.

[27] 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*. Association for Computational Linguistics. https://arxiv.org/abs/1908.10084

[28] Guilherme Rosa, Luiz Bonifacio, Vitor Jeronymo, 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. *Proceedings of the Sixteenth*

International Workshop on Juris-informatics (JURISIN 2022) (2022).

[29] 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. *Sixteenth International Workshop on Juris-informatics (JURISIN)* (2022).

[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] 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*.

[32] 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.

[33] Yunqiu Shao, Bulou Liu, Jiaxin Mao, Yiqun Liu, Min Zhang, and Shaoping Ma. 2020. Thuir@ coliee-2020: Leveraging semantic understanding and exact matching for legal case retrieval and entailment. *COLIEE 2020* (2020).

[34] Trevor Strohman, Donald Metzler, Howard Turtle, and W Bruce Croft. 2005. Indri: A language model-based search engine for complex queries. In *Proceedings of the international conference on intelligent analysis*, Vol. 2. Washington, DC., 2–6.

[35] Harsh Trivedi, Heeyoung Kwon, Tushar Khot, Ashish Sabharwal, and Niranjan Balasubramanian. 2019. Repurposing entailment for multi-hop question answering tasks. *Proc. of NAACL (2019)* (2019).

[36] Sabine Wehnert, Libin Kutty, and Ernesto William De Luca. 2023. 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, Revised Selected Papers*. Springer, 125–137.

[37] 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.

[38] M. Yoshioka, Y. Suzuki, and Y Aoki. 2021. Bert-based ensemble methods for information retrieval and legal textual entailment in coliee statute law task. *Proceedings of the Eigtht International Competition on Legal Information Extraction/Entailment (COLIEE 2021)* (2021).

[39] 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.

JNLP @COLIEE-2023: Data Augmentation and Large Language Model for Legal Case Retrieval and Entailment

Quan Minh Bui*
quanbui@jaist.ac.jp
Japan Advanced Institute of Science and Technology (JAIST)
Nomi, Ishikawa, Japan

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

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

Dieu-Hien Nguyen
ndhien@jaist.ac.jp
Japan Advanced Institute of Science and Technology (JAIST)
Nomi, Ishikawa, Japan

Khac-Vu-Hiep Nguyen
hiepnkv@jaist.ac.jp
Japan Advanced Institute of Science and Technology (JAIST)
Nomi, Ishikawa, Japan

Trang Pham Ngoc Anh
trangpna@jaist.ac.jp
Japan Advanced Institute of Science and Technology (JAIST)
Nomi, Ishikawa, Japan

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

ABSTRACT

The process of legal case retrieval involves identifying relevant cases that share similarities with a given case, while entailment requires assessing whether a legal statement can logically follow from another. These tasks are challenging due to the intricate nature of legal language and the vast quantity of legal documents. To overcome these difficulties, we propose implementing data augmentation techniques to produce additional training data and employing a large language model such as BART or T5 to capture the nuances of legal language. Specifically, we augment the provided dataset by generating synthetic cases that exhibit similar attributes to the original cases. We subsequently train a large language model on the augmented dataset and employ it to retrieve pertinent cases and determine entailment. Our findings also reveal that certain large language generative models, such as the Flan model have demonstrated potential for performing exceptionally well on the COLIEE task4 dataset. Notably, Flan model achieved state-of-the-art results on the COLIEE2023 and 2022 task 4 test sets.

KEYWORDS

Legal, Deep Learning, Contrastive Learning, Transformer Model, Large Language Model, Prompt Tuning

ACM Reference Format:
Quan Minh Bui*, Dinh-Truong Do*, Nguyen-Khang Le*, Dieu-Hien Nguyen, Khac-Vu-Hiep Nguyen, Trang Pham Ngoc Anh, 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, June 19, 2023, Braga, Portugal.* ACM, New York, NY, USA, 10 pages.

COLIEE 2023, June 19, 2023, Braga, Portugal

1 INTRODUCTION

The Competition on Legal Information Extraction and Entailment (COLIEE) is a well-known international competition organized each year with the goal of applying machine learning algorithms and techniques in the analysis and understanding of legal documents. Two main applications of using machine learning in this domain are entailment and information retrieval. The goal of entailment is to answer the question of whether a given proposition is true or false based on a piece of evidence. Moreover, information retrieval involves searching through a corpus of documents and ranking them according to their relevance to a query. The applications of this technique have not been widely used and are still being explored, but it has the potential to greatly assist attorneys in reducing the time and effort spent searching for legal documents or material required for a certain trial.

The utilization of deep learning models for the processing of lengthy and multi-language legal documents poses a number of challenges. The first of these pertains to the difficulty in acquiring and preparing a substantial and varied dataset suitable for training, which can be attributed to the technical and disparate nature of legal terminology. The second challenge is associated with the considerable processing power required by deep learning models to effectively comprehend and analyze the subtle nuances of legal language, with the added complexity of multi-language models further exacerbating the issue. In light of previous research [7, 24], our approach involves the implementation of data cleaning techniques to eliminate extraneous noise from the document, such as French paragraphs and paragraphs of insufficient length. Furthermore, we

*Equal contribution
+Corresponding author

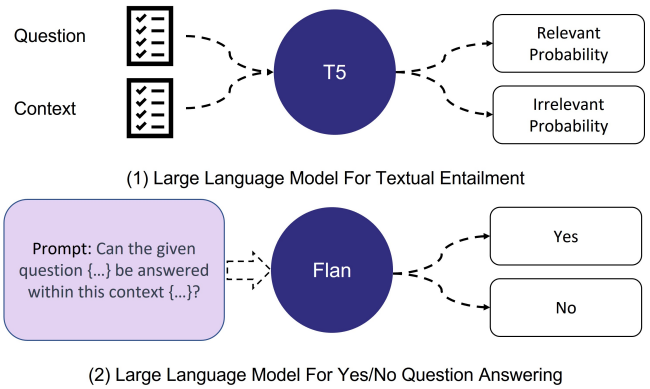


Figure 1: Application of Large Language Model

employ data augmentation strategies to generate new data points by applying a range of transformations to the original data.

Large language models (LLMs) [5, 17, 25] have gained significant attention in recent years due to their ability to perform various natural language processing tasks. These models use a deep neural network architecture and are trained on massive amounts of text data to generate human-like language output. According to the 1, there are 2 applications of LLMs that we apply in COLIEE 2023:

(1) Textual Entailment: using large language models to classify the probability of relevance between two documents.
(2) Prompt tuning for Yes/No question answering: Using various prompts to determine their effectiveness in the legal domain for a given question-context pair.

In the legal domain, these applications can be particularly useful for tasks such as document classification. LLMs can learn to understand and generate legal language more accurately and efficiently by providing legal-specific prompts or some training examples. This can save time and resources for legal professionals who need to process large amounts of legal text.

2 TASK DESCRIPTION

2.1 Task 1: The Legal Case Retrieval Task

This task focuses on finding the "noticed" cases for a new case from a set of supporting cases in the case law corpus. The "noticed" cases (or supporting cases) are the cases that support the decision-making process for the new case. More formally, given a new case Q and a set of candidate cases $C = \{C_1, C_2, ..., C_n\}$, the task is to extract the subset of supporting cases S from the candidate set C. In this task, micro average precision, recall, and F-measure are used for evaluation. The detailed formulas are expressed as follows:

$$Precision = \frac{|\text{Correctly retrieved cases}|}{|\text{Retrieved cases}|} \tag{1}$$

$$Recall = \frac{|\text{Correctly retrieved cases}|}{|\text{Correct cases}|} \tag{2}$$

$$\text{F-measure} = \frac{2 \times \textbf{Precision} \times \textbf{Recall}}{\textbf{Precision} + \textbf{Recall}} \tag{3}$$

2.2 Task 2: The Legal Case Entailment Task

In this task, a base case and its entailed fragment are given. The target is to identify a specific paragraph from a relevant case that entails the fragment of the base case. In a formal way, given the base case C, the entailed fragment F of the base case C, and a list of paragraphs $P = \{P_1, P_2, ..., P_n\}$ of a relevant case to the base case C, the target of this task is to identify a subset P' from P which contains the paragraphs that entail the fragment F. The evaluation used in this task is similar to Task 1.

2.3 Task 3: The Statue Law Retrieval Task

This task focuses on retrieving the relevant documents for an input "Yes/No" question and then using these documents to answer the input question. Given a legal bar exam question Q and Japanese Civil Code Articles $S = \{S_1, S_2, .., S_n\}$, the task requires one to collect a subset E from S that can support answering the question Q. The input questions are collected from Japanese Legal Bar exams and translated into English along with all the Japanese Civil Law articles. In this task, macro average precision, recall and F2-measure are used instead of micro average measures in Task 1 and Task 2. In addition, Mean Average Precision and R-precision can be used for further discussion on the submitted results. The formulas of the measures are shown below:

$$Precision = \text{Average of} \frac{|\text{Correctly articles for each query}|}{|\text{Retrieved articles for each query}|} \tag{4}$$

$$Recall = \text{Average of} \frac{|\text{Correctly retrieved articles for each query}|}{|\text{Correct articles for each query}|} \tag{5}$$

$$\text{F2-measure} = \frac{5 \times \textbf{Precision} \times \textbf{Recall}}{4 \times \textbf{Precision} + \textbf{Recall}} \tag{6}$$

2.4 Task 4: The Legal Textual Entailment Data Corpus

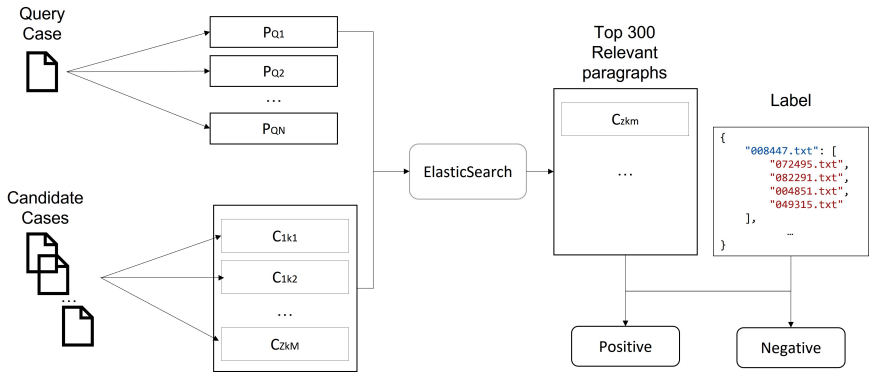
In this task, the articles retrieved from Task 3 will be used to answer the input "Yes/No" question. Given a question Q and a set of retrieved articles $S = \{S_1, S_2, ..., S_n\}$, the target is to determine the answer "Yes" or "No" for question Q. For evaluation, accuracy is used to measure the correctness of the predictions provided by the models.

$$Accuracy = \frac{|\text{Correctly predicted queries}|}{|\text{All queries}|} \tag{7}$$

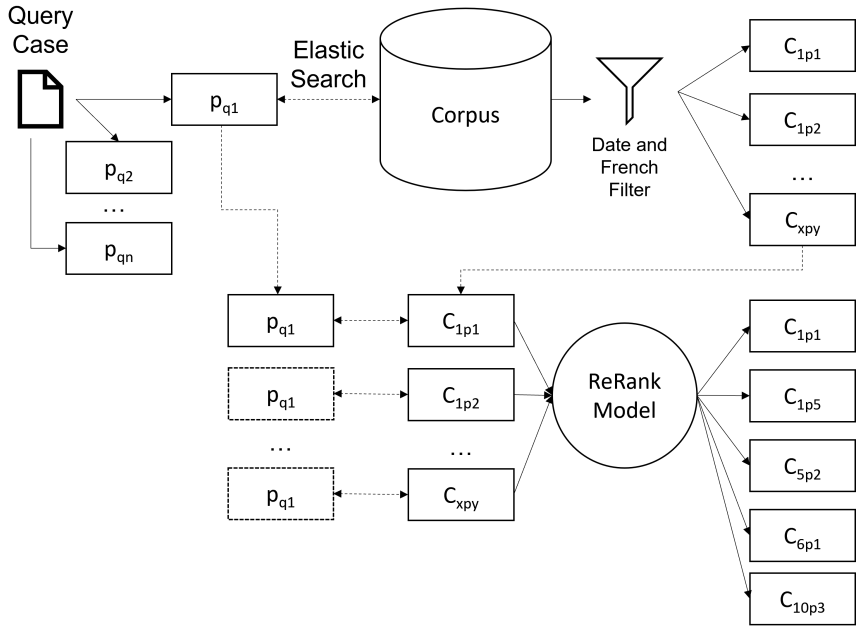
3 METHODOLOGY

3.1 Task1

The hypothesis is advanced that a given paragraph from the candidate's case has pertinence to a paragraph in the query case presently undergoing analysis. Based on this supposition, the query and candidate cases are segmented into paragraph-level components, denoted as $Q = p_{q1}, p_{q2}, ..., p_{qN}$ and $C_k = p_{k1}, p_{k2}, ..., p_{kM}$ respectively for the query and candidate cases. To determine the appropriate candidates, a semantic ranking approach is employed using bi-encoder similarity to identify the closest $p_{qn} - p_{km}$ pairs.



(a) Task 1 Data Augmentation



(b) Illustration of Task 1 Prediction Phase

Figure 2: Task 1 Pipeline Demonstration

3.1.1 Bi-encoder Similarity. Bi-Encoders are a type of deep neural network model that generates a sentence embedding for a given input sentence. This is achieved by passing the input sentence through the Bi-Encoder, resulting in the creation of a sentence embedding. In the context of sentence similarity or text classification tasks, sentence embeddings have proven to be an effective approach. Specifically, given two input sentences q_{qn} and q_{km}, we can pass them independently through the deep learning model to obtain sentence embeddings u and v, respectively. Once these embeddings are obtained, they can be compared using Euclidean distance, a commonly used similarity metric in natural language processing applications. Through this approach, we can effectively compare the similarity between two sentences and extract valuable insights for a variety of tasks, such as sentiment analysis, text classification, and information retrieval.

3.1.2 Training Phase. In order to compare the similarity between two input sentences, we employed:

- Sentence_Transformer [26], a popular deep learning model for natural language processing tasks.
- Contrastive model trained by data augmentation.

For Sentence_Transformer, we used *sentence-transformers/all-mpnet-base-v2*[1] the pre-trained model based on *microsoft/mpnet-base*[2] and fine-tuned in on a 1B sentence pairs dataset.

The present study employs data augmentation techniques to enhance the Task 1 dataset and generate training data for the contrastive model. The procedure outlined in Figure 2a involves querying Elasticsearch to retrieve 300 candidate paragraphs corresponding to a given query case paragraph. The paragraphs are

[1] https://huggingface.co/sentence-transformers/all-mpnet-base-v2
[2] https://huggingface.co/microsoft/mpnet-base

then assigned positive or negative labels based on whether their IDs belong to a predetermined set of labels.

We have $Q = \{p_{q1}, p_{q2}, ..., p_{qN}\}$ be the original dataset consisting of N samples, where each sample q_i represents a paragraph of text. The augmented dataset, Q', is generated by applying a transformation function, g(x), to each sample in the original dataset. Specifically, for each sample q_i in Q, we use Elasticsearch to retrieve 300 candidate paragraphs, denoted as $C = \{c_{p1}, c_{p1}, \ldots, c_{p300}\}$, based on a given query case paragraph. Let P_E be a set of positive labels corresponding to relevant paragraphs corresponding to irrelevant paragraphs. For each candidate paragraph c_{pi} in C, we assign this sample as c_{pi_pos} whether its ID belongs to P_E; otherwise c_{pi_neg}. The augmented dataset Q' is then defined as $Q' = \{(p_{qi}, c_{pi_{pos}}, c_{pi_{neg}}), (p_{qi+1}, c_{pi+1_{pos}}, c_{pi+1_{neg}}), ..., (p_{qN}, c_{pN_{pos}}, c_{pN_{neg}})\}$

For the contrastive model, we use Triplet Margin Loss [3] to fine-tune on Legal-BERT [9], a variant of BERT trained on the legal domain. Triplet Margin Loss [3] is a popular loss function in machine learning used to learn effective feature representations for tasks such as face recognition and image retrieval. The goal of this loss function is to learn embeddings in which similar samples are closer together and dissimilar samples are further apart in the feature space. Triplet Margin Loss is based on the idea of triplets, which consist of an anchor sample, a positive sample that is similar to the anchor, and a negative sample that is dissimilar to the anchor. The loss function penalizes embeddings that do not satisfy the condition that the distance between the anchor and the positive sample is smaller than the distance between the anchor and the negative sample by a margin. Triplet Margin Loss has shown promising results in various applications and has become a widely used technique in the field of deep learning. After obtaining the anchor, positive, and negative samples from the data augmentation step, we can fine-tune the Legal-BERT using the Formula 8:

$$L(a, p, n) = max\{d(a_i, p_i) - d(a_i, n_i) + margin, 0\} \tag{8}$$

3.1.3 Prediction Phase. As we can see in Figure 2b, for each paragraph denoted as p_{qi} in the given query case $Q = p_{q1}, p_{q2}, ..., p_{qN}$, Elasticsearch is utilized to search for the top n candidate paragraphs that have the highest BM25 score. Additionally, we have implemented a regulation to filter out any unreasonable candidates. Since a query solely refers to cases that have already been judged before the query case itself, we extract dates from the law cases. Based on the assumption that the current query cannot refer to a future case, we have defined the regulation, represented by Formula 9:

$$Rank_1 = \{c | c \in Rank_0 \wedge max(dates(c)) \leq max(dates(q))\} \tag{9}$$

In this context, $dates(c)$ denotes the collection of dates that have been present in the document c, while $Rank_0$ refers to the set of candidate paragraphs retrieved through the application of Elasticsearch. Furthermore, $max(dates(c))$ corresponds to the date associated with the case c. There is some noise in the documents, such as French-language content. Consequently, we aim to implement a secondary filtering mechanism 10 to exclude French-language content from the provided document since all the transformer variants we used are for English only.

$$Rank_2 = \{p_{qi} | p_{qi} \in Rank_1, lang(p_{qi}) \neq fr\} \tag{10}$$

In this formula, The function $lang(p_{qi})$[3] returns the language of the paragraph p_{qi}, and $\neq fr$ checks whether the language is not equal to French.

Finally, the sentence_transformer or contrastive model is utilized to re-rank the candidate paragraphs. The most relevant case is identified based on the top 5 paragraphs that possess the smallest Euclidean score.

3.2 Task2

One of the key challenges faced by IR systems is to ensure high accuracy and relevance in the search results presented to the users. Traditionally, IR systems rely on a single model, such as the vector space model, to represent documents and queries and to compute the relevance scores of documents to a given query. However, a single model may not capture all the complexities and nuances of the information and may lead to suboptimal results.

Ensemble [7, 10, 33] methods have emerged as a promising technique to address this challenge and to improve the performance of IR systems. Ensemble methods combine the outputs of multiple individual models or classifiers to arrive at a more accurate and robust prediction. By using an ensemble of diverse IR models, the resulting system can take advantage of the strengths of each model, while mitigating their weaknesses. In this way, ensemble methods can lead to a more accurate, effective, and reliable information retrieval system.

During the prediction phase, we utilized N transformer models, denoted as $T_1, T_2, ..., T_N$, respectively, where each model is associated with a specific loss function. For each query-candidate paragraph pair (q, c), we fed the pair into each of the N models to obtain the corresponding similarity scores $s_1(q, c), s_2(q, c), ..., s_N(q, c)$, where each $s_i(q, c)$ represents the similarity score computed by the $i-th$ model. The final similarity score for the pair, denoted as $s(q, c)$, is then calculated as:

$$s(q, C) = \sum_{i=1}^{N} s_i(q, c) \tag{11}$$

It should be noted that customization of the number of models employed and the associated loss functions for each model can be performed in accordance with the unique characteristics of the task and dataset. For the present study, two distinct training strategies were utilized in the training of the transformer models. As illustrated in Figure 3, the varied training approaches are capable of extracting differing information from the supplied data, and we anticipate that this technique will improve the discriminatory capacity of the transformer models.

3.3 Task3

3.3.1 Pre-processing. In statute laws, articles can contain many items, each of which may be divided into multiple samples. Samples are created by separating the common part of the article from the item, allowing for greater clarity and organization of the legal text. By breaking down articles into samples based on their items, legal

[3] https://pypi.org/project/langdetect/

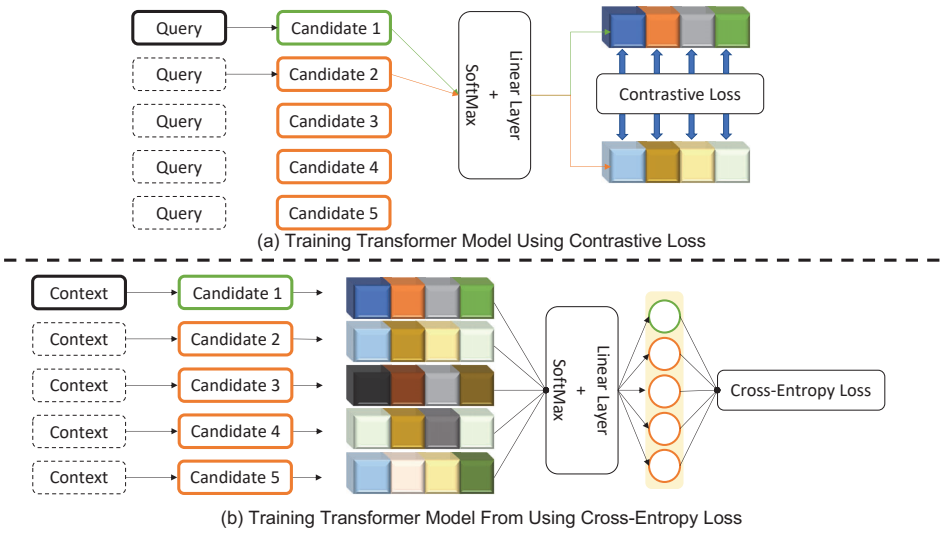


Figure 3: Contrastive loss and Cross-entropy loss illustration

professionals can better understand the details and nuances of the law. Table 1 shows an example of an article containing multiple items and the split article created by pre-processing. Table 2 shows an example of original and split articles where an item in an article contains multiple samples. We refer to the corpus of samples created by breaking down the original articles as the "split" corpus. We experimented with the original article corpus and the "split" corpus.

3.3.2 Retrieval approaches. Our approach to the statute law retrieval task explores and combines the following types of models

- **Lexical-based retrieval methods** like BM25 have the strength of being computationally efficient and easy to implement. They rely on the frequency and distribution of words in a document to determine their relevance to a query, making them effective for simple keyword-based searches. Additionally, lexical-based retrieval methods can handle large datasets and are highly scalable, making them suitable for search engines and other information retrieval systems. However, they may not perform well when faced with complex queries or when the relevance of a document to a query is not solely determined by its textual content
- **Dense retrieval methods** have the strength of being able to capture semantic relationships between words and phrases, making them effective for handling complex queries and understanding the meaning of a document beyond its textual content. They use neural networks to represent documents and queries in a high-dimensional vector space, allowing for efficient similarity search and ranking of results. Dense retrieval methods can also be trained on large amounts of data, making them highly adaptable to different tasks and domains
- **Hybrid retrieval methods**, which combine dense and sparse retrieval methods, have the strength of leveraging the advantages of both approaches. By using sparse methods to identify relevant documents based on keyword matching and then using dense methods to rank them based on semantic similarity, hybrid methods can improve the effectiveness of retrieval systems. This allows for efficiently handling complex queries and capturing semantic relationships between words and phrases while being computationally efficient.
- **Zero-shot language models** have the strength of being able to perform language-related tasks without being explicitly trained on them. This is achieved through their ability to generalize knowledge learned from one task to another related task. They can understand natural language input and produce relevant output without specific training, allowing for the efficient use of resources and the ability to perform a wide range of tasks. Additionally, zero-shot language models can handle multiple languages and produce multilingual output, making them highly versatile and useful for cross-lingual applications.
- **Large language models (LLMs)** have the strength of being able to generate high-quality natural language text that closely mimics human writing style and patterns. They can understand natural language input, including its nuances, idiomatic expressions, and context, and generate coherent and relevant output. LLMs can learn from large amounts of data, capturing complex relationships between words and phrases and performing a wide range of language-related tasks, including language translation, summarization, and question-answering. They can also be fine-tuned to specific domains and applications, making them adaptable and effective for various use cases.

3.4 Task 4

In the past few years, pre-trained language models (PLMs), such as BERT [13], have demonstrated their effectiveness in various domains, including the legal domain [1, 20]. The common approach for utilizing PLMs involves the "pre-training and fine-tuning" learning

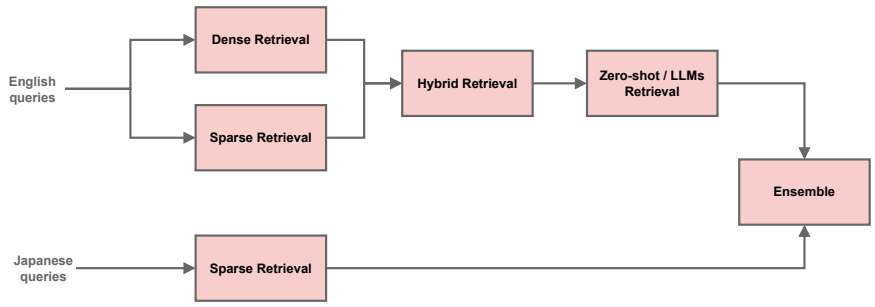


Figure 4: Example of a parametric plot ($\sin(x), \cos(x), x$)

Orginal Article:
Article 1
(1) Private rights must be congruent with the public welfare.
(2) The exercise of rights and performance of duties must be done in good faith.
(3) Abuse of rights is not permitted.
Split Article:
Article 1 Private rights must be congruent with the public welfare.
Article 1 The exercise of rights and performance of duties must be done in good faith.
Article 1 Abuse of rights is not permitted.

Table 1: Example of split Article 1

Orginal Article:
Article 450 (1) If an obligor has the obligation to provide a guarantor, that guarantor must:
(i) be a person with capacity to act; and
(ii) have sufficient financial resources to pay the obligation.
Split Article:
Article 450 If an obligor has the obligation to provide a guarantor, that guarantor must: be a person with capacity to act;
Article 450 If an obligor has the obligation to provide a guarantor, that guarantor must: have sufficient financial resources to pay the obligation.

Table 2: Example of split Article 450 - Item (1)

paradigm. In this paradigm, it often requires fine-tuning the PLM to adapt it to specific downstream tasks, such as textual entailment [23, 39].

Large language models (LLMs) are PLMs with a large number of parameters ranging from several billion to several hundred billion (e.g. the 11B-parameter T5-xxl [25] and the 175B-parameter GPT3 [6]). When performing LLMs, researchers have found that LLMs show unique abilities compared to medium-sized language models, including the ability to follow instructions [29]. This means that by fine-tuning LLMs on multi-task datasets with natural language instructions, the LLMs can perform well on unseen tasks that are also described in the form of instructions [22]. As a result, numerous studies have investigated LLMs' performance in zero-shot settings of downstream tasks, *e.g*, textual entailment [36]. In this section, we assess the performance of LLMs in a zero-shot setting for the legal textual entailment task of COLIEE 2023. Figure 5 shows an overview of our method using LLMs to perform Task 4. This consists of three main steps: Prompt Collecting, LLMs Running, and Label Extracting.

3.4.1 Prompt Collecting. Previous studies have shown that the selection of an appropriate prompt/instruction is crucial for LLMs' performance in zero-shot settings [16, 27]. Hence, we first construct a set of prompts to perform the legal textual entailment task. To accomplish this, we gather all the prompts from the GLUE tasks available in the PromptSource library [2]. We then convert these prompts to JSON format (as shown in Listing. 1) and obtain a set of 56 prompts that could serve as input instructions for the LLMs.

```
[
  {
    "id": 0,
    "label": [
      "True",
      "False"
    ],
    "prompt": "< relevant_articles >
              Question: < query > True or
              False?"
  },
  ...
]
```

Listing 1: Prompt set for Task 4

3.4.2 LLMs Running. In order to input the query Q and relevant articles S (derived from Task 3's results) into the LLMs, we first replace the <*query*> tag in the prompt with the query text Q, and the <*relevant_articles*> tag with the text of the relevant articles S. After that, we load the LLMs using the Huggingface[4] library and provide the modified prompt as input. The LLMs generate output text, which we then use to extract the answer.

[4] https://huggingface.co/

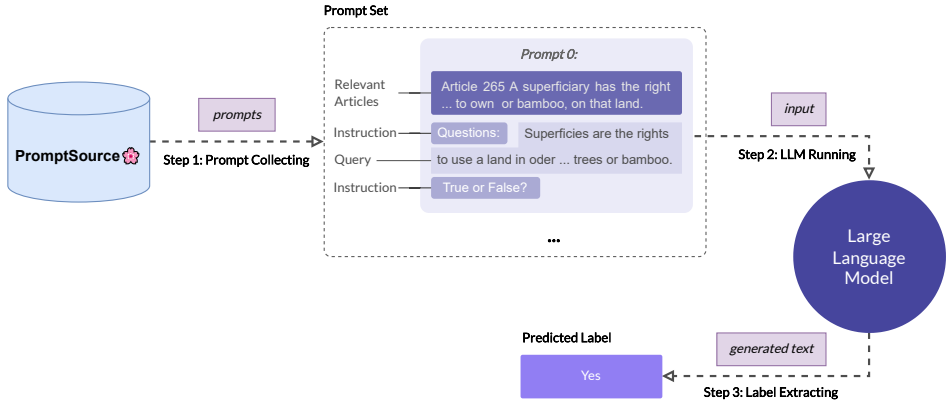


Figure 5: Overview of our method in Task 4.

Table 3: Contrastive dataset analysis

	Query	Positive	Negative
Average Length	84.956	92.091	90.522
Max Length	1947	1952	2259
Min Length	9	8	9
Total	5050500		

Table 4: Task1 COLIEE2023 competition result

Team	F1	Precision	Recall
THUIR	0.3001	0.2379	0.4063
IITDLI	0.2874	0.2447	0.3481
NOWJ	0.2757	0.2263	0.3527
JNLP (ours)	0.2604	0.2044	0.3586

3.4.3 Label Extracting. To extract the predicted answer from the generated text, we create a simple label mapping function based on the prompt labels. This function searches for negative labels (e.g *"false"*, *"incorrect"*, etc.) within the generated text. If any of these labels are present, the function returns 0. If not, it returns 1.

After evaluating the performance of our approach on the provided dataset, we select the prompt that yields the highest accuracy score. During the inference phase, we utilize this prompt to generate the final predicted labels.

4 EXPERIMENT AND RESULT

4.1 Task1

4.1.1 Data Augmentation Result. As shown in Table 3, we can generate more than 5 million triple anchor-positive-negative for training a transformer model.

4.1.2 Prediction Phase Result. As explicated in the Methodology section, the quantity of candidate paragraphs procured through Elasticsearch constitutes a crucial factor. Given that the quantity of candidate paragraphs may vary in magnitude, it can substantially impact the effectiveness of the contractive model in accurately identifying relevant cases. Through experimentation, it is discernible that the optimal number of candidate paragraphs for each query paragraph is either 3 or 5.

Based on the results presented in Table 4, our team was able to secure the 4th position in Task 1 of the COLIEE competition by utilizing the contrastive model and including three candidate paragraphs for each query paragraph.

4.2 Task2

For the ensemble phase, we use the following variant of the transformer model:

(1) Legal-BERT [9]
(2) XLM-RoBERTa [12]
(3) Longformer [4]
(4) DeBERTa [15]

The Legal-BERT model was trained using two distinct strategies, namely, contrastive and cross-entropy. Conversely, for other models, solely the cross-entropy strategy was employed, and two versions of these models, namely, base and larger, were trained. Notably, the larger version contains a higher number of parameters.

Based on the outcomes of our experiments, the performance of transformer models can be impacted by the imbalanced distribution of positive and negative data. Consequently, to address this issue, we conducted an experiment in which the ratio of positive to negative instances was set to 1:4.

Upon analyzing the results presented in Table 5, it can be inferred that the performance of the contrastive model alone was quite poor when evaluated on the Task2 test set. This highlights the limitations of a single model in handling complex natural language processing tasks. However, through the use of an ensemble approach, which involves combining the predictions of multiple models, we were able to significantly enhance our performance on this task. This is exemplified by our achievement of 3rd place in the COLIEE competition for Task 2, which is indicative of the effectiveness of our ensemble approach. The success of this technique can be attributed to its ability to harness the strengths of multiple models and overcome the weaknesses of individual models.

Table 5: Task2 COLIEE2023 competition result

Team	F1	Precision	Recall
CAPTAIN	0.7456	0.7870	0.7083
THUIR	0.7182	0.7900	0.6583
JNLP(CL_LPM)[1]	0.6818	0.7500	0.6250
JNLP(CL_PM)[2]	0.6545	0.7200	0.6000
JNLP(CL)[3]	0.5182	0.5700	0.4750

[1] Using contrastive model and large-size pre-trained models
[2] Using contrastive model and medium-size pre-trained models
[3] Using only contrastive model

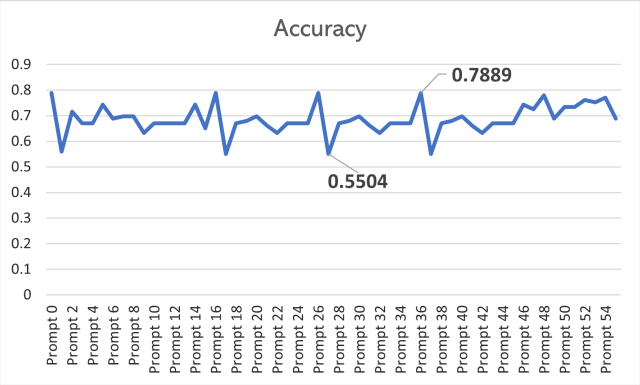


Figure 6: Effect of prompt on the performance.

4.3 Task3

4.3.1 Experimental Setup. The experiments are conducted on the COLIEE 2022 test set. The models used in the experiments are as follows:

- **BM25, Dense/Hybrid Retrieval**: We use the implementation from Pyserini [18].
- **Dense model**: We use the model ColBERT [31] for dense retrieval.
- **Zero-shot model**: We experiment with models MonoT5 [21], BART-NLI [37]
- **Large Language Models**: We experiment with models T0. [30], mT0 [19].

4.3.2 Results. Table 6 shows the performance of various methods on the COLIEE 2022 test set. The experiments indicate that the application of the BM25 model is more effective in the Japanese language as compared to English. In addition, the Hybrid retrieval method exhibits slightly superior performance in contrast to the Sparse and Dense retrieval methods. Furthermore, results obtained from the Split corpus are markedly more effective than those obtained from the original corpus. Regarding the Zero-shot models, it is evident that the BART-NLI model yields unsatisfactory results, whereas MonoT5 displays impressive performance, particularly the version equipped with 3 billion parameters. In contrast, the T0pp and mT0 models, despite having more parameters, demonstrate poor results. Furthermore, the study shows that an ensemble of BM25, utilizing different k1 and b parameters, produces certain improvements. Finally, the most outstanding results are obtained by the ensembles of MonoT5 in English and BM25 in Japanese.

4.4 Task 4

4.4.1 Experimental Setup. The experiments are conducted on various LLMs, which are split into two groups based on their fine-tuning dataset: Flan-based [11] and P3-based [19, 29]. The following are brief descriptions of the LLMs used:

- **Flan-based models**: flan-t5-xxl and flan-alpaca-xxl were initialized with t5-xxl [25] and ul2 [35] model checkpoints respectively, and fine-tuned on the Flan dataset consisting of 1.8K NLP tasks focused on instructions and chain-of-thought reasoning [11]. Flan-alpaca-xxl is an expanded version of flan-t5-xxl, which was further fine-tuned on the Stanford Alpaca synthetic instruction [34].
- **P3-based models**: t0pp and bloomz-7b1 were instruction-tuned versions of t5-xxl [25] and bloom-7b1 [32] respectively, on the instruction P3 dataset which contains of 2073 prompts for 177 datasets [29]. Additionally, mt0-xxl and mt0-xxl-mt were two instruction-tuned versions of the mt5 model [38], on the cross-lingual task mixture dataset xP3 [19].

Mt0-* models were used for performing the Japanese version of Task 4, while the other models were used for the English version.

4.4.2 Main results. Table 7 displays the results of our experiments on the COLIEE test sets for the years 2022 and 2023. Results on the test set of 2022 show that zero-shot LLMs outperform non-LLMs methods that are based on medium-sized PLMs in rows (1) and (2). This demonstrates that the LLMs can perform reasonably well on the textual entailment of legal domain if provided with appropriate instructions. In addition, we observed that larger model parameter sizes did not necessarily result in better performance, as shown by the results in rows (5) and (6). This observation is consistent with previous works [8, 28]. Moreover, by comparing the flan-t5-xxl (7) and t0pp (8) models, which use the same base models but different instruction fine-tuning data (Flan [11] and P3 [29]), we can conclude that using appropriate instruction fine-tuning data can significantly affect the performance of LLMs on downstream tasks.

For the COLIEE 2023 competition, we submitted three runs using different LLMs: a flan-t5-xxl (7), a flan-ul2 (6), and a flan-alpaca (5). The flan-aplpaca-xxl model achieves the highest scores among all submissions. This demonstrates the effectiveness of using LLMs on the legal textual entailment task.

4.4.3 Impact of different prompts. The performance of all collected prompts on the COLIEE 2022 test set with *flan-t5-xxl-zero-shot* setting is presented in Figure 6. The results show a significant gap of 23% in accuracy score between the best-performing prompt and the worst-performing one. This highlights the importance of selecting an appropriate prompt when using LLMs for the legal textual entailment task.

Table 6: Statue law retrieval performance of various methods measured on COLIEE 2022 Task 3 test set

Method		Language	Split Corpus	F2	Precision	Recall	#Correct	#Return
Sparse	BM25 (Top-1)	English		74.53	79.82	73.93	87	109
	BM25 (Top-1)	English	x	74.53	79.82	73.93	87	109
	BM25 (Top-1)	Japanese		75.66	81.65	75.03	89	109
	BM25 (Top-1)	Japanese	x	76.99	82.57	76.41	90	109
Dense	ColBERT (Top-1)	English		57.38	60.55	57.03	66	109
Hybrid	Hybrid (Top-1)	English		75.44	80.73	74.85	88	109
	Hybrid (Top-10)	English		35.21	10.46	92.39	114	1090
Zero-shot models and LLMs (applied on candidates retrieved by Hybrid)	BART_NLI	English		46.96	38.53	51.61	60	114
	MonoT5-Base	English		69.84	74.31	69.34	81	109
	MonoT5-Large	English		75.19	79.82	74.69	87	109
	MonoT5 (3 billions params)	English		76.67	80.73	76.22	88	109
	MonoT5 (3 billions params)	English	x	77.18	81.65	76.68	89	109
	mT0-xxl	English		37.67	37.39	39.53	47	71
	T0pp	English		33.17	18.45	55.66	69	467
Tuning BM25	BM25 (Top-1) (k1=0.99, b=0.4)	Japanese	x	77.91	83.49	77.32	91	109
	BM25 (Top-1) (k1=0.99 b=0.75)	Japanese	x	78.42	84.40	77.78	92	109
	BM25 Ensemble(k1, b) ∈ {(0.01, 0.4), (0.01, 0.75), (0.99, 0.4), (0.99, 0.75)}	Japanese	x	79.86	83.03	80.38	96	126
Ensemble	MonoT5-Ensemble (English) + BM25 (Japanese)	English + Japanese	x	**82.69**	**82.26**	84.20	101	136
	MonoT5-Ensemble (English) + BM25-Ensemble (Japanese)	English + Japanese	x	82.61	81.04	**84.97**	103	150

Table 7: Accuracy scores on the test sets of the legal textual entailment task of COLIEE 2022, 2023. The best single model for each year is highlighted in bold.

Description	Submission Name	Base model	Params	2022	2023
(1) Best of 2022 [14]	KIS2	BERT-based model	< 1B	0.6789	_
(2) 2nd best of 2022 [40]	HUKB-1	BERT-based model	< 1B	0.6697	_
(3) 2nd best of 2023	KIS2	_	_	_	0.6931
(4) flan-alpaca-xxl-zero-shot (ours)	JNLP3	t5-xxl	11B	**0.7889**	**0.7822**
(5) flan-ul2-zero-shot (ours)	JNLP2	ul2	20B	**0.7889**	0.7525
(6) flan-t5-xxl-zero-shot (ours)	JNLP1	t5-xxl	11B	**0.7889**	0.7525
(7) t0pp-zero-shot (ours)	_	t5-xxl	11B	0.7339	0.6732
(8) bloomz-7b1-zero-shot (ours)	_	bloom-7b1	7B	0.6422	0.5940
(9) mt0-xxl-zero-shot (ours)	_	mt5-xxl	13B	0.7155	0.7128
(10) mt0-xxl-mt-zero-shot (ours)	_	mt5-xxl	13B	0.7247	0.6435

5 CONCLUSION

In conclusion, the use of data augmentation and large language models has shown promising results for improving the performance of natural language processing models on the COLIEE legal dataset. Data augmentation and data cleaning techniques have been successful in generating additional training data and improving the robustness of NLP models. Large language models, such as Flan or T5, have shown impressive performance on COLIEE competition tasks, including legal document retrieval and legal question answering. By using these models on the COLIEE dataset, significant improvements in accuracy and efficiency have been achieved.

REFERENCES

[1] Yusuf Arslan, Kevin Allix, Lisa Veiber, Cedric Lothritz, Tegawendé F Bissyandé, Jacques Klein, and Anne Goujon. 2021. A comparison of pre-trained language models for multi-class text classification in the financial domain. In *Companion Proceedings of the Web Conference 2021*. 260–268.

[2] Stephen Bach, Victor Sanh, Zheng Xin Yong, Albert Webson, Colin Raffel, Nihal V. Nayak, Abheesht Sharma, Taewoon Kim, M Saiful Bari, Thibault Fevry, Zaid Alyafeai, Manan Dey, Andrea Santilli, Zhiqing Sun, Srulik Ben-david, Canwen Xu, Gunjan Chhablani, Han Wang, Jason Fries, Maged Al-shaibani, Shanya Sharma, Urmish Thakker, Khalid Almubarak, Xiangru Tang, Dragomir Radev, Mike Tian-jian Jiang, and Alexander Rush. 2022. PromptSource: An Integrated Development Environment and Repository for Natural Language Prompts. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*. Association for Computational Linguistics, Dublin, Ireland, 93–104. https://doi.org/10.18653/v1/2022.acl-demo.9

[3] Vassileios Balntas, Edgar Riba, Daniel Ponsa, and Krystian Mikolajczyk. 2016. Learning local feature descriptors with triplets and shallow convolutional neural networks.. In *Bmvc*, Vol. 1. 3.

[4] Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150* (2020).

[5] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems* 33 (2020), 1877–1901.

[6] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin

Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language Models are Few-Shot Learners. In *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (Eds.), Vol. 33. Curran Associates, Inc., 1877–1901. https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf

[7] Quan Minh Bui, Chau Nguyen, Dinh-Truong Do, Nguyen-Khang Le, Dieu-Hien Nguyen, Thi-Thu-Trang Nguyen, Minh-Phuong Nguyen, and Minh Le Nguyen. 2023. JNLP Team: Deep Learning Approaches for Tackling Long and Ambiguous Legal Documents in COLIEE 2022. 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, 68–83.

[8] Diedre Carmo, Marcos Piau, Israel Campiotti, Rodrigo Nogueira, and Roberto Lotufo. 2020. Ptt5: Pretraining and validating the t5 model on brazilian portuguese data. *arXiv preprint arXiv:2008.09144* (2020).

[9] 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).

[10] Viktoriia Chekalina and Alexander Panchenko. 2021. Retrieving comparative arguments using ensemble methods and neural information retrieval. *Working Notes of CLEF* (2021).

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

[12] Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116* (2019).

[13] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

[14] 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: JSAI-isAI 2022 Workshop, JURISIN 2022, and JSAI 2022 International Session, Kyoto, Japan, June 12–17, 2022, Revised Selected Papers*. Springer, 138–153.

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

[16] Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *arXiv preprint arXiv:2205.11916* (2022).

[17] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461* (2019).

[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 Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2021)*. 2356–2362.

[19] Niklas Muennighoff, Thomas Wang, Lintang Sutawika, Adam Roberts, Stella Biderman, Teven Le Scao, M Saiful Bari, Sheng Shen, Zheng-Xin Yong, Hailey Schoelkopf, Xiangru Tang, Dragomir Radev, Alham Fikri Aji, Khalid Almubarak, Samuel Albanie, Zaid Alyafeai, Albert Webson, Edward Raff, and Colin Raffel. 2022. Crosslingual Generalization through Multitask Finetuning. arXiv:2211.01786 [cs.CL]

[20] Ha-Thanh Nguyen, Minh-Phuong Nguyen, Thi-Hai-Yen Vuong, Minh-Quan Bui, Minh-Chau Nguyen, Tran-Binh Dang, Vu Tran, Le-Minh Nguyen, and Ken Satoh. 2022. Transformer-Based Approaches for Legal Text Processing: JNLP Team-COLIEE 2021. *The Review of Socionetwork Strategies* 16, 1 (2022), 135–155.

[21] Rodrigo Nogueira, Zhiying Jiang, Ronak Pradeep, and Jimmy Lin. 2020. Document Ranking with a Pretrained Sequence-to-Sequence Model. In *Findings of the Association for Computational Linguistics: EMNLP 2020*. Association for Computational Linguistics, Online, 708–718. https://doi.org/10.18653/v1/2020.findings-emnlp.63

[22] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Gray, 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. In *Advances in Neural Information Processing Systems*, Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (Eds.). https://openreview.net/forum?id=TG8KACxEON

[23] Juliano Rabelo, Mi-Young Kim, and Randy Goebel. 2021. The application of text entailment techniques in coliee 2020. 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, 240–253.

[24] Juliano Rabelo, Mi-Young Kim, and Randy Goebel. 2023. Semantic-based classification of relevant case law. 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, 84–95.

[25] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research* 21, 1 (2020), 5485–5551.

[26] Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084* (2019).

[27] Laria Reynolds and Kyle McDonell. 2021. Prompt programming for large language models: Beyond the few-shot paradigm. In *Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems*. 1–7.

[28] 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).

[29] Victor Sanh, Albert Webson, Colin Raffel, Stephen Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Arun Raja, Manan Dey, M Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal Nayak, Debajyoti Datta, Jonathan Chang, Mike Tian-Jian Jiang, Han Wang, Matteo Manica, Sheng Shen, Zheng Xin Yong, Harshit Pandey, Rachel Bawden, Thomas Wang, Trishala Neeraj, Jos Rozen, Abheesht Sharma, Andrea Santilli, Thibault Fevry, Jason Alan Fries, Ryan Teehan, Teven Le Scao, Stella Biderman, Leo Gao, Thomas Wolf, and Alexander M Rush. 2022. Multitask Prompted Training Enables Zero-Shot Task Generalization. In *International Conference on Learning Representations*. https://openreview.net/forum?id=9Vrb9D0WI4

[30] Victor Sanh, Albert Webson, Colin Raffel, Stephen Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Arun Raja, Manan Dey, M Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal Nayak, Debajyoti Datta, Jonathan Chang, Mike Tian-Jian Jiang, Han Wang, Matteo Manica, Sheng Shen, Zheng Xin Yong, Harshit Pandey, Rachel Bawden, Thomas Wang, Trishala Neeraj, Jos Rozen, Abheesht Sharma, Andrea Santilli, Thibault Fevry, Jason Alan Fries, Ryan Teehan, Teven Le Scao, Stella Biderman, Leo Gao, Thomas Wolf, and Alexander M Rush. 2022. Multitask Prompted Training Enables Zero-Shot Task Generalization. In *International Conference on Learning Representations*. https://openreview.net/forum?id=9Vrb9D0WI4

[31] Keshav Santhanam, Omar Khattab, Jon Saad-Falcon, Christopher Potts, and Matei Zaharia. 2022. ColBERTv2: Effective and Efficient Retrieval via Lightweight Late Interaction. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, Seattle, United States, 3715–3734. https://doi.org/10.18653/v1/2022.naacl-main.272

[32] Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, et al. 2022. Bloom: A 176b-parameter open-access multilingual language model. *arXiv preprint arXiv:2211.05100* (2022).

[33] Xiaotong Sun, Ying Qu, Lianru Gao, Xu Sun, Hairong Qi, Bing Zhang, and Ting Shen. 2021. Ensemble-based information retrieval with mass estimation for hyperspectral target detection. *IEEE Transactions on Geoscience and Remote Sensing* 60 (2021), 1–23.

[34] Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto. 2023. Stanford alpaca: An instruction-following llama model. *GitHub repository* (2023).

[35] Yi Tay, Mostafa Dehghani, Vinh Q Tran, Xavier Garcia, Dara Bahri, Tal Schuster, Huaixiu Steven Zheng, Neil Houlsby, and Donald Metzler. 2022. Unifying language learning paradigms. *arXiv preprint arXiv:2205.05131* (2022).

[36] Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V Le. 2022. Finetuned Language Models are Zero-Shot Learners. In *International Conference on Learning Representations*. https://openreview.net/forum?id=gEZrGCozdqR

[37] Jamaal Hay Wenpeng Yin and Dan Roth. 2019. Benchmarking Zero-shot Text Classification: Datasets, Evaluation and Entailment Approach. In *EMNLP*. https://arxiv.org/abs/1909.00161

[38] Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A Massively Multilingual Pre-trained Text-to-Text Transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, Online, 483–498. https://doi.org/10.18653/v1/2021.naacl-main.41

[39] 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.

[40] 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.

A Topic-Based Approach for the Legal Case Retrieval Task

Luisa Pereira Novaes, Daniela Vianna, Altigran da Silva
{luisa.novaes,dvianna,alti}@icomp.ufam.edu.br
Instituto de Computação, Universidade Federal do Amazonas
Manaus, Amazonas, Brazil

ABSTRACT

This paper describes the method developed by the UFAM team in the 10th COLIEE for Task 1, the legal case retrieval task. In a nutshell, we propose a topic-based approach composed of two phases: *filtering* and *ranking*. In the filtering phase, a topic discovery technique is applied to the entire dataset to select an initial set of candidate cases for each query. Then, in the ranking phase, three different ranking functions can be applied to each pair query and candidate from the set producing a sorted list of relevant cases per query. Finally, using a predefined *threshold* (cut-off value), we select the most relevant documents for a given query. Based on this two-phased approach, we implemented three different solutions that achieved the two best precision values for this competition. Of a total of 22 results, our best result was ranked number 12 in the overall ranking.

CCS CONCEPTS

• **Information systems → Rank aggregation**; **Similarity measures**; **Retrieval models and ranking**.

KEYWORDS

topic discovery, legal case retrieval, Doc2Vec, cosine similarity, COLIEE

ACM Reference Format:
Luisa Pereira Novaes, Daniela Vianna, Altigran da Silva. 2023. A Topic-Based Approach for the Legal Case Retrieval Task. In *Proceedings of COLIEE 2023 workshop, June 19, 2023, Braga, Portugal.* ACM, New York, NY, USA, 5 pages.

1 INTRODUCTION

The fundamental characteristic of a Common Law System is that it confers great importance on prior cases. That means that, if similar cases have earlier been decided by a court of law, they have to be studied, and their interpretation has to be taken into consideration. Countries such as the United States of America, Canada, and Australia follow the Common Law System. However, even for countries with judiciary systems based on Civil Law, or statutory law, precedents can influence decisions in court, as is the case of Brazil, where precedents have become a fundamental source of law with the goal of reaching a more fair and standardized judiciary system.

COLIEE 2023, June 19, 2023, Braga, Portugal

Considering the ever-growing number of new legal cases, the task of finding comparable cases can be time-consuming and costly, impacting the agility and fairness of legal systems worldwide. The ongoing digital transformation of the legal domain has opened an opportunity for the application of new technologies in law, including artificial intelligence solutions for a variety of tasks such as legal case retrieval and legal text entailment. Even though document retrieval is a well-studied field, when proposing solutions for the legal domain specifics challenges arise, as is the case of the complexity and uniqueness of the legal language (legalese) and the fact that legal documents are often long-length documents.

The problem of searching for legal decisions to support or disprove a case is addressed by the Competition on Legal Information Extraction and Entailment (COLIEE) through its Legal Case Retrieval task (Task 1). This paper presents the approach developed by the UFAM team for Task 1 of the 10th COLIEE. Our approach has two phases: *filtering* and *ranking*. For the filtering phase, a topic discovery model is used to select an initial set of candidate cases considerably reducing the number of documents to be analyzed in the next phase. Then, in the ranking phase, three different ranking function can be applied to further refine the selection and determine the final candidates for each query. The first function measures the cosine similarity between each query case and the candidate cases. The second function relies solely on the probability of a query's dominant topic being assigned to a candidate document. Finally, the third function combines the first and second functions considering both the cosine similarity and the percentage of topic contribution.

Experiments conducted on the training set demonstrate that the topic discovery method used during the filtering phase positively impacts the final result by significantly reducing the size of the candidate set. For the ranking phase, the function based on cosine similarity produced the best results.

The remainder of this article is organized as follows. In Section 2 we discuss related work. The legal case retrieval task, Task 1 of the COLIEE competition, is described in Section 3. Section 4 introduces the topic-based approach proposed in this work. An experimental evaluation of our method is presented in Section 5. Finally, conclusions and future research directions are discussed in Section 7.

2 RELATED WORK

Retrieving similar legal cases is a crucial task in the legal domain and has been extensively explored by researchers and practitioners. The team from University of Alberta (UA) emerged as the Task 1 winner of COLIEE 2022 by proposing an approach based on measuring the similarity between paragraphs of legal cases and generating feature vectors based on these similarities, followed by using a classifier to determine whether the cases should be flagged or not [4]. The use of topic discovery techniques in the task of legal document retrieval

is not new in the COLIEE competition. In the 2017 edition, Nanda et al. [3] utilized topic models to categorize documents into clusters of topics as part of their efforts to measure the similarity between queries and documents. This involved assigning a topic vector to each document, which could then be compared to the query's topic vector, resulting in the selection of the initial top-n documents that are most similar to the query. These top-n documents were further processed using a semantic similarity model to identify the most relevant documents for the given query. In the recent COLIEE 2022 edition, the TUWBR team approached this problem with the assumption that there is a topical overlap between query and notice cases. They converted the case documents to queries and used a BM25 model for ranking the results [1]. Their results, however, were not very expressive.

3 TASK DESCRIPTION

3.1 Task 1: Legal Case Retrieval

The legal case retrieval task involves reading a new case, denoted as *Query*, and extracting relevant supporting cases, denoted as $S_1, S_2, \ldots S_n$ from a comprehensive corpus of case law. Notice that, differently from traditional information retrieval tools, in which the most common type of queries are keyword-based queries, in this task, a query is a text document (legal case).

3.2 Dataset

The 2023 dataset for this task primarily consists of cases from the Federal Court of Canada provided by Compass Law. The training set comprises 4400 documents in total. Among them, 959 query cases and 4310 documents were used as queries and/or supporting relevant cases. 290 documents were not used as a query or as a supporting case document. The maximum number of relevant supporting cases per query was 34, and 1 was the minimum. On average, there are approximately 4.7 relevant cases per query case in the provided training set. In the test set, 1334 documents in total were provided, being 319 queries with no relevant documents revealed. For the legal case retrieval task, the goal is to find all the relevant documents for each of the 319 queries from the given test set.

3.3 Metrics

For this task, the evaluation measure was precision, recall and F-measure, with F-measure calculated by:

$$\text{F-measure} = (2 \times Precision \times Recall)/(Precision + Recall) \quad (1)$$

where Precision is the number of correctly retrieved cases for all queries divided by the number of retrieved cases for all queries, and Recall is the number of correctly retrieved cases for all queries divided by the number of relevant cases for all queries.

4 METHODOLOGY

In this section, we introduce the topic-based approach proposed in this work to address the legal case retrieval task (Task 1). The approach has two phases: filtering and ranking. In Section 4.1 we detail the filtering phase in which a topic discovery approach is used to generate a reduced set of legal case candidates for each query. Then, in Section 4.2 we present a ranking method responsible for determining the final list of relevant legal cases per query (ranking phase).

4.1 Filtering Phase: Topic Discovery

Inspired by the work presented in [5], as a first step to address the legal case retrieval task, a topic discovery model, called *BERTopic* [2], is trained using the entire training dataset presented in Section 3.2, which also includes all queries. *BERTopic* finds dense clusters of documents by leveraging embeddings and a class-based TF-IDF (c-TF-IDF) procedure. Precisely, *BERTopic* has three distinct steps: generation of document embeddings using a pre-trained language model; dimensionality reduction of document embeddings to optimize the clustering phase; then, extraction of topic representations using c-TF-IDF. The assumption is that semantically similar documents are a strong indicator of an existing topic, so by applying *BERTopic* to the training set we aim to find a reduced set of candidates for a query given the fact that those documents belong to topics that are somehow relevant to this query. Note that in addition to finding the most relevant topic for a document (dominant topic), *BERTopic* can also return a list with the top k topics that could be assigned to a document with different probabilities. With that, a document may belong to multiple topics with different levels of relevance (probabilities). This allows us to experiment with different values of k when defining whether a document is a candidate for a query. If a query's dominant topic appears in the top k topics assigned to a document, then this document is considered a candidate for this query. Finding the ideal value of k is important to reduce the number of candidates to be considered in the next phase (ranking phase) without discarding any relevant document in the filtering phase.

BERTopic is very versatile. It supports several embedding techniques to perform the embedding step. Since legal cases are often long-length documents, we opted to use Doc2Vec in combination with *BERTopic* to generate a vector representation of each document in our collection. The Doc2Vec model is trained using the same training data as *BERTopic*. (Section 3.2).

4.2 Ranking phase

After the filtering phase, different ranking functions can be applied to the reduced set of candidates finding the most relevant documents to a query. This phase is called the ranking phase. In this work, we have explored three different approaches:

(1) The first approach uses the Doc2Vec model trained in the previous phase, filtering phase (Section 4.1), to generate vector representations (embeddings) for queries and candidates. Then, a cosine similarity function is applied to each pair query embedding and document embedding, generating for each query a sorted list of candidate documents.
(2) The second approach takes into consideration the probability of the query's dominant topic to be assigned to a candidate document. With that, we try to measure the contribution of the query's dominant topic to the candidate document. Those probabilities are generated by *BERTopic* during the clustering step and indicate the probability of a document to be assigned to a topic (or cluster).

(3) The third approach is a combination of the two previous approaches: it adds the score obtained from both, cosine similarity and topic contribution.

After the sorted list of candidates per query is generated using one of the three ranking approaches explained above, a final cut, based on a pre-defined *threshold*, can be applied to this list resulting in the final list of relevant document (supporting cases) per query. The ideal values of k and *threshold* are experimentally defined and will be detailed in Section 5.

5 EVALUATION

5.1 Preprocessing

We carried out a preprocessing stage on the competition data, in which we employed several techniques to clean and normalize the text. Firstly, we removed all punctuation marks to avoid any interference they may cause. Secondly, we removed all stopwords, which are commonly occurring words that do not carry significant meaning and may skew our analysis. Finally, we converted all text to lowercase to ensure consistency in word representation and avoid the creation of multiple representations of the same word due to different capitalizations. These procedures were applied to ensure a cleaner and more accurate representation of the text data, which ultimately improved the performance of our proposed approach. It is important to note that this preprocessing step was applied before training the Doc2Vec and the *BERTopic* models.

5.2 Filtering

Following the completion of the topic discovery phase on the training dataset of 2023 (Section 3.2), a total of 119 distinct topics were identified from the corpus of 4400 documents that underwent analysis. With the topic contribution percentages computed for each document, we were able to discern the dominant topic(s) for every document. To assess the effectiveness of these topics as indicators of relevance, we measured the presence of the dominant topic of a query among the k top topics of the candidates. We call this metric *Topification Recall*. Then, we compute the average of Topification Recall for all queries. This metric is called *Average Topification Recall*. We presented the results for *Average Topification Recall* in Figure 1 varying the value of k.

We observed a strong correlation between the topics of a query and the topics of its relevant decisions. By searching for the dominant topic of the query in the first five dominant topics of the candidates ($k = 5$), we can already find more than 62% of the relevant cases for each query on average. This result shows the efficacy of the topic modeling approach in reducing the size of the candidate set during the filtering phase.

Additionally, we conducted experiments to explore the possibility of incorporating a minimum threshold based on the percentage of contribution of a query's dominant topic to the candidate document. That is, if the query's dominant topic appears in the top k topics of a document but with a percentage below a given threshold, we remove this document from the list of candidates. However, the results obtained using such a threshold did not meet our expectations. As a result, this parameter was not considered in our final proposed approach.

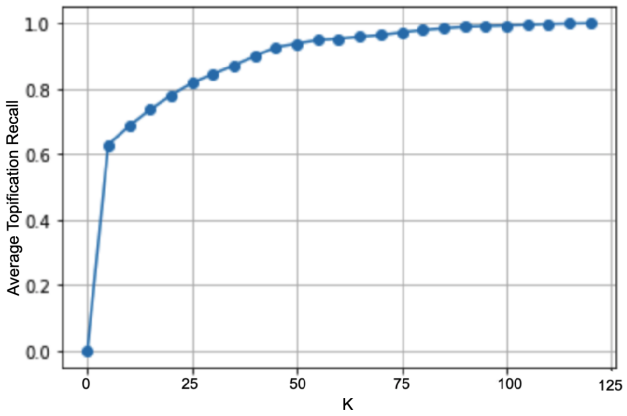


Figure 1: Average topification recall for different values of k.

Up to this point, we have trained a Doc2Vec model and a BERTopic model using the training set from the 2023 dataset during the filtering phase. Now, we can apply these models to the 1334 documents in the test set to find the relevant documents for each of the 319 queries in the test set. The first step is to generate an embedding for each document using the Doc2Vec model and then use the BERTopic model to find the dominant topics for each query and the top k topics for the remaining documents.

5.3 Ranking

Subsequent to the initial filtering phase, the ranking phase is applied to the list of candidates sorting all documents based on some similarity measure (ranking function) between documents and the query. As introduced in Section 4.2, three different ranking functions were explored in this work. The first one is based on the cosine similarity between the embeddings of queries and candidate documents. The second function is based on the percentage of contribution of a query's dominant topic to the candidate document. Finally, the third function, a hybrid function, combines both cosine similarity and percentage contributions of topics. After the ranking function is applied, producing a sorted list of candidate documents, a *threshold* (cut-off value) is applied to select the final list of candidates per query. To find the optimal values of k (cut-off value for the filtering phase) and *threshold* (cut-off value for the ranking phase) we conduct a set of experiments varying the values of k, the ranking function, and finally the values of *threshold*.

Table 1 shows the F-measure results obtained by varying k from 5 to 65 during the filtering phase, then during the ranking phase applying the ranking function based on cosine similarity and varying the values of *threshold* from 0.32 to 0.41. Initially, values of *threshold* varying from 0.05 to 1.00 were considered. However, since we noticed that the best results were achieved in the interval between 0.32 and 0.41, we chose to display only those results in the table. We observed that for the cosine similarity ranking function, the four best F-measure values were obtained using $k = 50$ and *threshold* values of 0.39, 0.38, 0.37, and 0.36. A $k = 50$ means that if a query's dominant topic appears in the top 50 topics assigned to a document, then this document is a candidate to be a relevant case

K \Threshold	0.32	0.33	0.34	0.35	0.36	0.37	0.38	0.39	0.40	0.41
65	0.1459	0.1503	0.1546	0.1549	0.1569	0.1577	0.1578	0.1579	0.1553	0.1512
60	0.1460	0.1504	0.1546	0.1550	0.1570	0.1577	0.1579	0.1579	0.1553	0.1512
55	0.1460	0.1503	0.1546	0.1549	0.1569	0.1576	0.1579	0.1579	0.1553	0.1512
50	0.1460	0.1503	0.1547	0.1550	**0.1570**	**0.1577**	**0.1579**	**0.1579**	0.1553	0.1512
45	0.1459	0.1503	0.1546	0.1548	0.1568	0.1574	0.1577	0.1576	0.1552	0.1512
40	0.1452	0.1494	0.1536	0.1537	0.1557	0.1565	0.1570	0.1574	0.1552	0.1513
35	0.1452	0.1497	0.1537	0.1538	0.1557	0.1564	0.1568	0.1571	0.1549	0.1509
30	0.1451	0.1496	0.1536	0.1536	0.1554	0.1560	0.1564	0.1566	0.1546	0.1506
25	0.1453	0.1499	0.1538	0.1539	0.1557	0.1563	0.1565	0.1567	0.1547	0.1506
20	0.1451	0.1495	0.1534	0.1534	0.1553	0.1558	0.1560	0.1560	0.1539	0.1497
15	0.1444	0.1488	0.1527	0.1525	0.1545	0.1548	0.1547	0.1553	0.1532	0.1489
10	0.1441	0.1487	0.1529	0.1524	0.1541	0.1542	0.1542	0.1545	0.1527	0.1483
5	0.1461	0.1503	0.1542	0.1535	0.1548	0.1544	0.1542	0.1542	0.1522	0.1479

Table 1: F-measures for different values of k, values of *thresholds*, and ranking function cosine similarity.

to this query, and so, it should be in the final list of candidates generated during the filtering phase. Values of *threshold* between 0.36 and 0.39 mean that after computing the cosine similarity between each pair of query and candidate documents, only documents with similarity greater than the threshold will be considered relevant to a given query.

Table 2 presents a comparison between the F-measures obtained with the three different ranking functions – cosine similarity, topic contribution, and hybrid (cosine similarity + topic contribution) –, fixing $k = 50$ and varying the *threshold* values from 0.32 to 0.41. The values of k and *threshold* were selected based on the previous experiments with ranking function cosine similarity (Table 1). The values of F-measure presented in Table 2 confirm that our best approach relies on the ranking function cosine similarity, with $k = 50$ and *threshold* values of 0.36, 0.37, 0.38, and 0.39. In the future, new experiments varying k and *threshold* should be performed to evaluate both ranking functions, topic contribution and hybrid, in different scenarios.

Analyzing the results presented in Table 2 and considering that three different solutions could be submitted to the COLIEE competition, we initially selected the following parameters: $k = 50$, ranking function cosine similarity, and thresholds 0.39, 0.38, 0.37. To confirm our choice of threshold, given the fact that *threshold* 0.36 achieved a F-measure close to the one obtained by *threshold* 0.37, we execute a different set of experiments to evaluate their performance with different sets of data. The experiments are similar to the ones presented so far, but instead of using the entire training data to chose the optimal value of *threshold*, we divide the training data in two sets, train (70%) and test (30%), evaluating the impact of *threshold* values in both sets. This experiment is repeated five times, each time building different train and test sets. The conclusion was that a *threshold* of 0.36 were more consistent than *threshold* 0.37 for the majority of the experiments. In summary, three different solutions were submitted, all of them with $k = 50$, ranking function cosine similarity, and *threshold* values of 0.36, 0.38, and 0.39.

6 OVERALL RESULTS

As this was our inaugural participation in the competition, we are pleased to report that our team achieved a notable performance, securing a 12th place out of 22 finishes (Table 3). Our best result was achieved using $k = 50$, cosine similarity, and *threshold* = 0.36. Followed by the solutions with *threshold* = 0.38 and *threshold* = 0.39, respectively. Notably, our submissions stood out in terms of precision, with rankings of 1st, 2nd, and 4th. This encouraging result bolsters our belief in the effectiveness of topic-based approach for the filtering phase. However, it is worth mentioning that after receiving the labels, we discovered that by selecting different parameters, we could have achieved an even better result, with an F1 score of 0.2685. This insight highlights the importance of continuous refinement and optimization in our ranking phase as we strive to improve our performance in future competitions.

We observed that for some queries our solutions do not generate a final list of candidates. This is because even with our best ranking function, cosine similarity, the *threshold* ends up eliminating all possible candidates due to low similarity between query and candidates. Even though applying the *threshold* led to empty sets for some queries, it is still better to use the threshold than returning the n best-ranked candidates bellow the threshold, since it negatively impacts precision.

7 CONCLUSIONS AND FUTURE WORK

In this work, we presented a topic-based approach to address the legal case retrieval task, Task 1 of the COLIEE competition. The proposed approach has two phases: filtering and ranking. In the filtering phase a topic discovery method is applied to select an initial set of candidates to a given query. Then, in the ranking phase, similarity measures (ranking function) are applied to the initial candidate set, followed by a cut-off using an experimentally defined threshold, generating the final list of relevant documents to a given query. A variety of experiments were performed using the training dataset to define the optimal parameters for the proposed approach.

The results demonstrated the effectiveness of the topic-based approach used during the filtering phase, resulting in an smaller initial candidate set without compromising the recall. However, it also highlighted that there is room for improvement in the ranking phase, which can leads to more expressive results in the future.

Ranking Function	0.32	0.33	0.34	0.35	0.36	0.37	0.38	0.39	0.40	0.41
Cosine Similarity	0.1460	0.1503	0.1547	0.1550	**0.1570**	**0.1577**	**0.1579**	**0.1579**	0.1553	0.1512
Topic Contribution	0.0839	0.0839	0.0845	0.0841	0.0843	0.0841	0.0840	0.0843	0.0843	0.0845
Hybrid	0.0961	0.0956	0.0946	0.0945	0.0938	0.0930	0.0933	0.0925	0.0928	0.0930

Table 2: F-measure $k = 50$, different values of *thresholds*, and ranking function cosine similarity, topic contribution, and hybrid.

Team	File	F1	Precision	Recall
THUIR	thuirrun2.txt	0.3001	0.2379	0.4063
THUIR	thuirrun3.txt	0.2907	0.2173	0.4389
IITDLI	iitdli_task1_run3.txt	0.2874	0.2447	0.3481
THUIR	thuirrun1.txt	0.2771	0.2186	0.3783
NOWJ	nowj.d-ensemble.json	0.2757	0.2263	0.3527
NOWJ	nowj.ensemble.json	0.2756	0.2272	0.3504
IITDLI	iitdli_task1_run1.txt	0.2738	0.2107	0.3912
IITDLI	iitdli_task1_run2.txt	0.2681	0.2063	0.3830
JNLP	jnlp_cl_3_dates.json	0.2604	0.2044	0.3586
NOWJ	nowj.bestsingle.json	0.2573	0.2032	0.3504
UA	pp_0.8_10_3.csv	0.2555	0.2847	0.2317
UFAM	**task1_2023_k50t036_3.json**	**0.2545**	**0.2975**	**0.2224**
JNLP	jnlp_sb_3_dates.json	0.2511	0.1971	0.3458
JNLP	jnlp_cl_5_dates.json	0.2493	0.1931	0.3516
UA	pp_0.7_9_2.csv	0.2390	0.3045	0.1967
UA	pp_0.65_10_3.csv	0.2345	0.2400	0.2293
UFAM	**task1_2023_k50t038_2.json**	**0.2345**	**0.3199**	**0.1851**
UFAM	**task1_2023_k50t039_1.json**	**0.2156**	**0.3182**	**0.1630**
YR	task1_yr_run1.txt	0.1377	0.1060	0.1967
YR	task1_yr_run2.txt	0.1051	0.0809	0.1502
LLNTU	task1_llntucliiss_2023.json	0.0000	0.0000	0.0000
LLNTU	task1_llntu3q4clii_2023.json	0.0000	0.0000	0.0000
Total teams:		8		
Total submissions:		22		

Table 3: COLIEE 2023 Results for Task 1

Based on the experiments, three solutions were submitted to be applied to the test set, with our best solution obtaining the 12nd position in the competition leader board. In terms of precision, our solutions were placed on 1st, 2nd, and 4th.

Given the results obtained during the competition, as a next step, we plan to investigate different ranking functions to be used during the ranking phase. We also plan to explore different language models to be applied both in the filtering phase and the ranking phase. The Doc2Vec model used to generate vector representations for the documents were trained using only the training set from the 2023 dataset. Another possibility would be to train a more robust Doc2Vec model using legal documents from a variety of other sources.

ACKNOWLEDGMENTS

This research was partially supported by Jusbrasil Postdoctoral Fellowship Program, Brazilian funding agency FAPEAM-POSGRAD 2020 (Resolution 002/2020), the Coordination for the Improvement of Higher Education Personnel-Brazil (CAPES) financial code 001, and an individual grant from CNPq (307248/2019-4) to Altigran da Silva.

REFERENCES

[1] Tobias Fink, Gabor Recski, Wojciech Kusa, and Allan Hanbury. 2023. Statute-enhanced lexical retrieval of court cases for COLIEE 2022. arXiv:2304.08188 [cs.IR]
[2] Maarten Grootendorst. 2022. BERTopic: Neural topic modeling with a class-based TF-IDF procedure. arXiv:2203.05794 [cs.CL]
[3] Rohan Nanda, Kolawole John Adebayo, Luigi Di Caro, Guido Boella, and Livio Robaldo. 2017. Legal Information Retrieval Using Topic Clustering and Neural Networks. In *COLIEE@ICAIL*.
[4] 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.
[5] Daniela Vianna and Edleno Silva de Moura. 2022. Organizing Portuguese Legal Documents through Topic Discovery. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Madrid, Spain) *(SIGIR '22)*. Association for Computing Machinery, New York, NY, USA, 3388–3392. https://doi.org/10.1145/3477495.3536329

NOWJ at COLIEE 2023 - Multi-Task and Ensemble Approaches in Legal Information Processing

Thi-Hai-Yen Vuong
University of Engineering and Technology, VNU
Hanoi, Vietnam
yenvth@vnu.edu.vn

Hai-Long Nguyen
University of Engineering and Technology, VNU
Hanoi, Vietnam
long.nh@vnu.edu.vn

Tan-Minh Nguyen
University of Engineering and Technology, VNU
Hanoi, Vietnam
20020081@vnu.edu.vn

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

Thai-Binh Nguyen
University of Engineering and Technology, VNU
Hanoi, Vietnam
20020328@vnu.edu.vn

Ha-Thanh Nguyen
National Institute of Informatics
Tokyo, Japan
nguyenhathanh@nii.ac.jp

ABSTRACT

This paper presents the NOWJ team's approach to the COLIEE 2023 Competition, which focuses on advancing legal information processing techniques and applying them to real-world legal scenarios. Our team tackles the four tasks in the competition, which involve legal case retrieval, legal case entailment, statute law retrieval, and legal textual entailment. We employ state-of-the-art machine learning models and innovative approaches, such as BERT, Longformer, BM25-ranking algorithm, and multi-task learning models. Although our team did not achieve state-of-the-art results, our findings provide valuable insights and pave the way for future improvements in legal information processing.

CCS CONCEPTS

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

KEYWORDS

Legal information processing, COLIEE, NOWJ, Multi-task, Ensemble

1 INTRODUCTION

COLIEE (Competition on Legal Information Extraction/Entailment) is an annual event focusing on advancing legal information processing techniques and applying them to real-world legal scenarios. The competition consists of four tasks, divided into two main categories: case law and statute law competitions. The tasks involve various challenges, such as legal case retrieval, legal case entailment, statute law retrieval, and legal textual entailment. Fortunately, these challenges can be identified and addressed by referring to previous research in the field.

Chalkidis Ilias and Kampas Dimitrios [2] conducted a survey on the early adaptation of Deep Learning in legal analytics, focusing on three main fields: text classification, information extraction, and information retrieval. Their study emphasized the importance of semantic feature representations, which are crucial for the successful application of deep learning in natural language processing tasks. In addition to their analysis, they provided pre-trained legal word embeddings using the WORD2VEC model, which were trained on large corpora containing legislations from various countries, including the UK, EU, Canada, Australia, USA, and Japan.

Nguyen et al. [6] addressed the challenge of representing legal documents for statute law document retrieval. They proposed a general approach using deep neural networks with attention mechanisms and developed two hierarchical architectures with sparse attention, named Attentive CNN and Paraformer. These methods were evaluated on datasets in English, Japanese, and Vietnamese. The results showed that attentive neural methods significantly outperformed non-neural methods in retrieval performance across datasets and languages. Pretrained transformer-based models [12] achieved better accuracy on small datasets but with high computational complexity, while the lighter weight Attentive CNN performed better on large datasets. The proposed Paraformer outperformed state-of-the-art methods on the 2021 COLIEE dataset, achieving the highest recall and F2 scores in the top-N retrieval task 3.

Vuong et al.'s [13] addressed the challenges of case law retrieval, a complex task involving legal case retrieval and legal case entailment. The difficulties stem from the long length of query and candidate cases, the need to identify legal relations beyond lexical or topical relevance, and the effort required to build large and accurate legal case datasets. To tackle these challenges, they proposed a novel approach called the supporting model, which is based on the case-case supporting relation, paragraph-paragraph matching, and decision-paragraph matching strategies. Furthermore, they introduced a method to automatically create a large weak-labeling dataset to overcome the lack of data for training deep retrieval models. Experimental results demonstrated that their solution achieved state-of-the-art results for both case retrieval and case entailment phases.

COLIEE 2023, June 19, 2023, Braga, Portugal

As the first time participating COLIEE, the NOWJ team aims to tackle the 2023 competition by utilizing state-of-the-art machine learning models and innovative approaches. For Tasks 1 and 2, we rely on BERT and Longformer pre-training models without using any external data. Additionally, Task 2 employs an internal data generation method based on Vuong et al [13] method to overcome the lack of data and enhance the legal case retrieval process. For Task 3, our team utilizes a two-phase retrieval system that employs the BM25-ranking algorithm and a BERT-based model for re-ranking, along with additional techniques to handle relations between articles and exploit information from the Legal Textual Entailment Data Corpus tasks. Lastly, for Task 4, we use a multi-task model with a pre-trained Multilingual BERT as the backbone.

Although we did not achieve state-of-the-art results, our findings in this competition provide good insights and pave the way for further improvements in legal information processing.

2 TASK INTRODUCTION

The competition consists of four tasks focusing on case law and statute law. Tasks 1 and 2 address the case law challenges, while tasks 3 and 4 target the statute law problems. Task 1, the legal case retrieval task, requires participating systems to read a new case and extract supporting cases from a case law corpus. Task 2, the legal case entailment task, involves the systems identifying a paragraph from existing cases that entails the decision of a new case. Task 3, part of the statute law competition, requires systems to retrieve relevant Japanese civil code statutes. Finally, Task 4 involves the systems confirming the entailment of a yes/no answer from the retrieved civil code statutes. The competition aims to advance legal information processing techniques and apply them to real-world legal scenarios through the development of innovative and efficient systems.

2.1 Task 1&2: The Legal Case Retrieval and Entailment Task

In the legal case retrieval task, participating systems are required to examine a new legal case Q and extract supporting cases $S_1, S_2, ...S_n$ from a case law corpus to provide backing for the decision of Q. In real-world legal scenarios, lawyers and legal professionals often need to discover pertinent case laws to substantiate their arguments in court. This task emulates the process of identifying precedents and offering legal arguments based on previous cases' analysis. Automating this procedure with an efficient system allows saving time and resources while ensuring accuracy and relevance in supporting cases.

The UA team [7] developed a successful approach for this task in the COLIEE 2022 competition. Their method combined semantic similarity representation at the sentence level and a Gradient Boosting binary classifier trained on 10-bin histograms containing similarity scores between sentences of the query and candidate cases. Additionally, they applied simple pre- and post-processing heuristics to generate the final results. As a result, they achieved the highest ranking among all competitors, outperforming 9 teams with a total of 26 submissions.

For the legal case entailment task, participating systems must pinpoint a specific paragraph from existing cases that involves the decision of a new legal case Q. Given a decision Q for a new case and its relevant counterpart R, the systems must identify which paragraph supports Q's decision. In actual practice, it is crucial for lawyers and other professionals in law to locate precise arguments or reasoning within existing situations while building strong foundations for their cases. Utilizing an efficient system can help save time, guarantee accuracy when automating this procedure, strengthen their arguments by detecting relevant paragraphs from existing ones.

The NM team [9] conducted experiments with zero-shot models in the legal case entailment task. Utilizing large language models, such as GPT-3, they found that scaling the number of parameters in a language model improved the F1 score by more than 6 points compared to their previous zero-shot result. This suggests that larger models may possess stronger zero-shot capabilities, at least for this specific task. Their 3B-parameter zero-shot model outperformed all other models, including ensembles, in the COLIEE 2021 test set and achieved the best performance of a single model in the COLIEE 2022 competition, ranking second only to an ensemble consisting of the 3B model and a smaller version of the same model. Despite the challenges posed by large language models, particularly latency constraints in real-time applications, the NM team demonstrated the practical use of their zero-shot monoT5-3b model in a search engine for legal documents.

2.2 Task 3&4: The Statute Law Retrieval and Entailment Task

For statute law retrieval tasks involving answering yes/no questions about Japanese Civil Code statutes ($S_1, S_2, ..., S_n$) participating systems will read a question (Q) then retrieve relevant data from databases accordingly. This simulates locating specific provisions needed when determining if certain laws apply given various situations encountered during actual practice sessions among lawyers working with clients. This work seeks advice regarding potential outcomes resulting from particular instances requiring clarification between parties involved within disputes occurring throughout professional settings worldwide today where many people interact daily exchanging ideas concerning different aspects life experience overall. In actual practice, lawyers and legal professionals must quickly identify the relevant statutes governing specific situations. Automating this process with an efficient system can save time and ensure accuracy when interpreting and applying laws to various cases.

The HUKB team [14] proposed a method that utilized three different Information Retrieval (IR) systems. Their new IR system was designed to measure the similarity of descriptions of judicial decisions between questions and articles. In addition to this new system, they also employed an ordinal keyword-based IR system (BM25) and a BERT-based IR system that was proposed in COLIEE 2020. Due to the diverse characteristics of these systems, ensembled results provided better recall without sacrificing much precision. The HUKB group's ensemble, which combined their newly proposed IR system and the keyword-based IR system, achieved the best performance for COLIEE 2022 Task 3.

For the statute law entailment task, participating systems are responsible for confirming whether Japanese Civil Code provisions

entail a yes/no answer to question Q. Given Q along with relevant statutes $(S_1, S_2, ..., S_n)$, systems must determine if they support a "YES" ("Q") or "NO" ("not Q") response.

In real-world legal scenarios, it is essential for lawyers and legal professionals to assess the implications of legal provisions on specific cases accurately. Automating this process with an efficient system can help save time, ensure accuracy, and enhance decision-making by effectively evaluating the relevance and implications of these provisions.

The KIS team [4] developed a successful approach for COLIEE 2022 Task 4, which aimed to solve the textual entailment part of the Japanese legal bar examination problems. By employing an ensemble of a rule-based method and a BERT-based method, and utilizing data augmentation and modular ensembling techniques, they improved the correct answer ratio. Their approach integrated additional proposed methods, such as Sentence-BERT for data selection and person name inference for replacing anonymized symbols. As a result, the KIS team achieved the best score among the Task 4 submissions with a correct answer ratio of 0.6789 in accuracy on the formal run test dataset.

3 METHODOLOGY

3.1 Task 1-2: The Legal Case Retrieval and Entailment Task

The relationship between base case - case in the legal case retrieval task and decision - paragraph in the legal case entailment task is relatively similar, the candidate case supports the base case or the paragraph supports the decision. The length of the legal case, paragraphs, and decision makes a significant distinction between these two tasks. Due to the extremely extensive texts in both the query and candidate instances, the work of legal case retrieval is quite difficult. To tackle these challenges, Vuong et al. proposed a novel approach called the supporting model, which is based on the case-case supporting relation, paragraph-paragraph matching, and decision-paragraph matching strategies [13].

On the other hand, the relationship between the paragraphs in the legal case will be lost if only matching at the paragraph/decision - paragraph level. Therefore, we build a case-level matching model to evaluate the support relationship of the base case and candidate cases. By incorporating both local and global attention mechanisms, Longformer is proposed to effectively encode long texts with thousands of tokens, overcome inability to process long documents limitations that other pre-trained language models have [1]. Longformer could capture effectively the legal case documents with the characteristic of long length.

In this study, we build two matching phases to solve the legal case retrieval and entailment task: mono tatching (paragraph/decision level) and panorama matching (case level).

3.1.1 Pre-processing. Firstly, we implemented some simple pre-processing steps:

- Due to the majority of queries being in English, we have chosen to eliminate all French content from the data, even if the legal case that involves or includes French translations.
- Segment case into paragraphs based on common struture of the legal case.
- Extraction of case's year through a rule-based method. Our assumption is that noticed cases could not be more recent than the base case. Thus, these years are used to filter out candidate cases that include dates more recent than the most recent date mentioned in a base case.
- Removal of redundant characters using regex. We removed duplicate endline, space characters and punctuations (exclude period, commas, question marks, etc.).
- Detect important passages by using heuristic. The placement of a paragraph in the legal case document also reveals its importance level such as whether paragraph is in "I. Introduction" or "II. Background", and so on. Beside, some of the paragraphs in base case quote other cases, in where placeholders like "SUPPRESSED" are used of the cited cases, these paragraphs also contain important content and words in comparison with other case law.

3.1.2 Mono Matching - Paragraph/Fragment level. This phase is a combination of two models which perform lexical and semantic matching, respectively:

- Lexical matching: using BM25 calculates the relevance score betwent paragraph/decision - paragraph based on the frequency of query terms in the query and their frequency in the entire collection of documents, as well as other factors such as document length and average query length.
- Semantic matching: to extract the semantic relationship we built a supporting model [13]. Is is fine-tuned with the weak label dataset and the legal case entailment dataset.

Particularly for the legal case retrieval task, the search space is huge with 4400 candidates case in the train set and 1300 candidates cases in the test set. To ensure the model's performance, we initially narrow down the search space by utilizing a lexical model to select potential candidates. For each paragraph of a base case, we retrieved the top 200 candidate paragraphs. We identify candidate cases through the returned candidate paragraphs. If a candidate case appears more than twice, we will retain the highest score. Ultimately, we will retain the cases with the best scores, up to a maximum of k cases.

3.1.3 Panorama Matching - Case level. In this stage, a Longformer model was implemented to compare a base case with candidate cases based on their similarities and relatedness in the panorama. In consideration of the typical average length of legal cases, which is approximately 3000 tokens, it was observed that a (base, candidate) case pair would exceed the token limitation of the Longformer model. To overcome this limitation, it was deemed necessary to curate the input by retaining only the most important paragraphs of the base case. This allowed us to mitigate the input length while still preserving the salient information necessary for effective matching with relevant candidate cases. This curation approach ensured that the Longformer model could successfully process the input data, thereby achieving superior matching results while also enhancing the efficiency of the matching process.

In the process of constructing the training dataset, we initially selected cases that were labeled as noticed to the query and assigned them as positive samples. We then proceeded to identify negative samples by considering cases that were labeled as not noticed for

each query case, but were retrieved by the mono matching phase. As a result, the resulting dataset consisted of pairs of cases (base, candidate) with a label ratio of 1:2 for positive and negative samples, respectively.

3.2 Task 3-4: The Statue Law Retrieval and The Legal Textual Entailment Data Corpus

Due to the good performance on the recall score of the BM25 model on the retrieval task, which is proven on previous works [5] [10] [11], the retrieval problem of task 3 is tackled using two-stage ranking including BM25 ranking model as first stage and BERT-base ranking model as the second stage. The BERT-based ranking model utilizes multi-task learning, which combines the goal of the retrieval (task 3) and textual entailment (task 4) problems.

3.2.1 First-stage BM25 ranking. The BM25 model [8] is a probabilistic relevance-based model that is widely used in the retrieval field. As it primarily operates on statistical processes and computes the relevance score through a single mathematical formula, it has a fast retrieval time in large corpora. The BM25 model is employed in this work to narrow down the candidate articles for the training phase. Additionally, its relevance score is utilized to improve the recall score during the testing phase by ensembling it with the ranking-BERT's relevance score.

According to the experiment, limiting the negative samples by BM25 ranking significantly reduces the training time but still remains the training efficency. Since the BM25 ranking selects the articles that are relevant to the given query in the lexical level, restricting the candidate documents by the top result of BM25 could make the re-ranking model has ability of distinguishing lexical-related and semantic-related candidates. To conclude, employing the pre-ranking BM25 model can make the ranking model more effectively in both training and inference phase.

3.2.2 Multi-task model with BERT. Although BM25 model can achieve a good recall score with a sufficient number of top candidates, its precision score is still very low. Therefore, a re-ranking BERT-based model is utilized for filtering the BM25's candidates and improving the precision score while keeping the recall score consistent. In this work, the re-ranking models'architecture is determined based on the fact that whether a query is considered as *yes* or *no* highly depends on the perspective of the legal experts. This means that when a query is labeled as *yes* (or *no*) by the legal experts, they tend to find the relevant articles that support their perspective. In other words, the relevant candidates are highly correlated with the *yes/no* result. Arcording the above observation, a multi-task model which has two output heads, including the retrieval head and textual entailment head, is employed as the re-ranking model. The model utilized the BERT architecture as the backbone and two output heads were added on top of it.

For conveniently comparing the performance of model when working with the different languages, the **bert-base-multilingual-uncased** pre-train parameters are used as the initialization for the model's back-bone.

3.3 Ensemble Model

Integrating multiple models can lead to a more efficient and effective solution, as it enables us to leverage the individual strengths of each model. Consequently, we establish emsemble methods which handle the intricacies and complexities that may not be addressed by a single model, while also enhancing the overall performance and precision of the system.

3.3.1 Boosting Ensemble. Boosting ensembles can be utilized in the retrieval process to filter out negative samples step-by-step. In each boosting step, a subset of candidates will be eliminated by a ranking model. This approach could combine the advantages of multiple models, which leads to an improvement in the accuracy and efficiency of the retrieval system. For example, with each input query, the retrieval system has to consider n legal documents in the database denoted as a set $S = \{s_1, s_2, ..., s_n\}$. After first boosting step, a subset of S will be removed from the potential candidate set by a ranking model, which forms a new candidate set with m elements denoted as $S_1 = \{s_{i_1}, s_{i_2}, ..., s_{i_m}\}$ where $i_j \in \{1, 2, 3, ..., n\}$ and $m < n$. The process is performed similarly in the remaining boosting step. In this investigation, ranking models from the lexical level to the semantic level are used in the ranking phases, respectively.

3.3.2 Weighted Ensemble. According to our experiments, the Lexical Matching Models often gain a relatively good recall score, while the Semantic Matching Models improve the precision score. So, combining the results of these two models could help to raise the overall F2 score. The relevance score of the semantic matching model is ensembled with the lexical matching model using the equation 1.

$$relevant\_score = \alpha * bm25\_score + \beta * bert\_score \quad (1)$$

The $bm25\_score$ is the relevance score given by the BM25 model (or Lexical Matching Model), $bert\_score$ is the relevance score given by the re-ranking BERT-based model (or Semantic Matching Model). The min-max normalization is performed with both $bm25\_score$, $bert\_score$, and the $relevant\_score$. The min-max normalization is computed as in Equation 2.

$$x' = \frac{x - min(x)}{max(x) - min(x)} \quad (2)$$

where x is an original value, x' is the normalized value. The trail-threshold inference strategy which determines the relevant articles based on the highest relevance score is used. A candidate is considered as relevant article if its relevance score satisfies the Equation 3.

$$\frac{highest\_score - candidate\_score}{highest\_score} <= trail\_threshold \quad (3)$$

The $highest\_score$ is the highest relevance score among all candidates of the given query and the $candidate\_score$ is the relvant score of the considering candidate. All the relevance scores here are the score after the combination and normalization process. For tuning optimal α, β and $trail\_threshold$, a grid-search process is conducted on the development set.

3.3.3 Voting ensemble. In many cases, the voting ensemble method, which uses the predicted results of several models and determines

the final label based on the majority, could be an effective error-correcting process [3]. This approach could significantly improve the overall results by considering the perspectives of multiple models. In this study, the voting ensemble method was employed for one of the submissions, which turned out to be the best run among all runs.

4 EXPERIMENT AND RESULT

4.1 Measuring

The evaluation metrics of Tasks 1 and 2 are precision, recall, and F-measure. All the metrics are micro-average, which means the evaluation measure is calculated using the results of all queries. The definition of these measures is as follows:

$$\text{Precision} = \frac{\text{\# correctly retrieved cases (paragraphs) for all queries}}{\text{\# retrieved cases (paragraphs) for all queries}} \tag{4}$$

$$\text{Recall} = \frac{\text{\# correctly retrieved cases (paragraphs) for all queries}}{\text{\# relevant cases (paragraphs) for all queries}} \tag{5}$$

$$\text{F-measure} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{6}$$

For Task 3, evaluation measures are precision, recall and F2-measure. All the metrics are macro-average (evaluation measure is calculated for each query and their average is used as the final evaluation measure) instead of micro-average (evaluation measure is calculated using results of all queries). The definition of these measures is as follows:

$$\text{Precision} = \text{average of} \frac{\text{\# correctly retrieved articles for each query}}{\text{\# retrieved articles for each query}} \tag{7}$$

$$\text{Recall} = \text{average of} \frac{\text{\# correctly retrieved articles for each query}}{\text{\# relevant articles for each query}} \tag{8}$$

$$\text{F-measure} = \text{average of} \frac{5 \times \text{Precision} \times \text{Recall}}{4 \times \text{Precision} + \text{Recall}} \tag{9}$$

For Task 4, the evaluation measure will be accuracy, with respect to whether the yes/no question was correctly confirmed:

$$\text{Accuracy} = \frac{\text{\# queries which were correctly confirmed as true or false}}{\text{\# all queries}} \tag{10}$$

4.2 Task 1

An official corpus has been provided for the evaluation of legal case retrieval models in COLIEE-2023 Task 1. The corpus relates to a database of mainly Federal Court of Canada case laws from Compass Law. Table 1 outlines the dataset statistics for Task 1, which includes 959 query cases against 4400 candidate cases in the training set and 319 query cases against 1335 candidate cases in the test dataset. The average number of paragraphs per case in the training dataset is 42.29, while the testing dataset has an average of 37.51 paragraphs per case. Further analysis revealed an average of 4.47 noticed cases in the training dataset and 2.69 noticed cases in the testing dataset.

We implement the lexical matching based on BM25 method using Elasticsearch [1]. We experimented five options of top-k candidate cases: $N = 10, N = 20, N = 50, N = 100, N = 200$ with different features. *Recall@k* measure is used for evaluating the list of returned candidates. Recall@k is (Number of correctly predicted articles in the $top - k$ results) / (Total number of gold articles). Table 2 presents experimental results of the lexical matching method. In $k = 10$ and $k = 20$, using only important paragraphs instead of a whole case improved Elasticsearch performance significantly. Specifically, there is a 77% increase in Recall@k for $k = 10$ and a 32% increase for $k = 20$. Additionally, including Year into queries also showed an improvement in Elasticsearch performance. Based on this evaluation, we chose top 200 candidates from Elasticsearch method.

Table 1: Statistics of the dataset for Task 1

	Train	**Test**
# queries	959	319
# candidate cases	4400	1335
# noticed cases per query		
Min	1	1
Max	34	17
Average	4.67	2.69
# paragraphs per case		
Min	2	3
Max	1117	617
Average	42.29	37.51

Table 2: Top-k recall score of the lexical matching method

Top k	**10**	**20**	**50**	**100**	**200**
All paragraphs	0.1783	0.2961	0.4803	0.5997	0.7190
Important paragraphs	0.3076	0.3921	0.5176	0.6161	0.7257
All paragraphs + Year	0.1877	0.3110	0.5026	0.6100	0.7152
Important paragraphs + Year	0.3334	0.4211	0.5319	0.6275	0.7290

Our team has submited 3 runs for the private test. The first run named **NOWJ.bestsingle** employed the mono matching model, which used the boosting ensemble method to eliminate candidates judged as non-potential and retained at most only one candidate for each paragraph in the base case. The second one named **NOWJ.ensemble** used the panorama matching model. The final run is **NOWJ.d-ensemble** used the voting ensemble of results from the two previous runs.

Table 3 illustrates the leaderboard of Task 1, NOWJ is our team. We submitted three official runs and ranked third among all teams. THUIR achieves the best performance in terms of F1 and Recall scores, while UFAM has the highest Precision score. The search space is large, which is 1335 candidate cases against 319 query cases in the testing dataset. Furthermore, the average case length in the training set is 3712.71 words, which prevents models from effectively aggregating information. Further analysis on our results revealed that combining the mono and panorama matching methods improved model's performance.

[1] https://www.elastic.co/

Table 3: Results on the test set of Task 1

#	Team	Run	F1	P	R
Other team's best results					
1	THUIR	thuirrun2	0.3001	0.2379	0.4063
2	IITDLI	iitdli_task1_run3	0.2874	0.2447	0.3481
4	JNLP	jnlp_cl_3_dates	0.2604	0.2044	0.3586
5	UA	pp_0.8_10_3	0.2555	0.2847	0.2317
6	UFAM	task1_2023_k50t036_3	0.2545	0.2975	0.2224
7	YR	task1_yr_run1	0.1377	0.1060	0.1967
8	LLNTU	task1_llntucliiss_2023	0.0000	0.0000	0.0000
Our results					
3	NOWJ	nowj.d-ensemble	0.2757	0.2263	0.3527
3	NOWJ	nowj.ensemble	0.2756	0.2272	0.3504
3	NOWJ	nowj.bestsingle	0.2573	0.2032	0.3504

4.3 Task 2

Table 4 shows the dataset statistics for Task 2, which includes 625 queries in the training dataset and 100 queries in the testing dataset. The average number of paragraphs per query in the training dataset is 35, while the testing dataset has an average of 37.65 paragraphs per query. There is an average of 1.17 entailed fragments in the training dataset and 1.20 entailed fragments in the testing dataset. The average length of a entailed fragment in both dataset is approximately 34 words.

Table 4: Statistics of the dataset for Task 2

	Train	Test
# query case	625	100
# paragraphs per query		
Min	3	3
Max	283	170
Avg	35	37.65
# entailed fragments per query		
Min	1	1
Max	5	4
Avg	1.17	1.20
# words per fragments		
Min	4	8
Max	115	111
Avg	34	33.18

The problem raised on task 2 is the textual entailment of the relevant cases. In this task, we submitted three runs for the private test. The first run, named **NOWJ.non-empty**, employed the mono matching model using the weighted ensemble method. The second one named **NOWJ.hp** and the final run (**NOWJ.hr**) used the boosting ensemble method to eliminate candidates judged as non-potential and then applied the weighted ensemble method to get the final result with different hyperparams setting.

Table 5 illustrates the leaderboard of Task 2. We submitted three official runs and ranked third among all teams. CAPTAIN achieves the best performance in terms of F1 and Recall scores while THUIR has the highest Precision score. Our team utilized BERT-based models to tackle Task 1 and 2 in COLIEE 2023. Since the introduction of BERT in 2018, newer pre trained language models with improved performance have been developed, as GPT-3, T5. These large models have more parameters and are trained on larger datasets. However, deploying large language models could be a challenge due to infrastructure and training cost requirements.

Table 5: Results on the test set of Task 2

#	Team	Run	F1	P	R
Other team's best results					
1	CAPTAIN	mt5l-ed	0.7456	0.7870	0.7083
2	THUIR	thuir-monot5	0.7182	0.7900	0.6583
3	JNLP	jnlp_bm_cl_1_pr_1	0.6818	0.7500	0.6250
4	IITDLI	iitdli_task2_run2	0.6727	0.7400	0.6167
5	UONLP	task2_test_no_labels_2023	0.6387	0.6441	0.6333
7	LLNTU	task2_llntukwnic_2023	0.1818	0.2000	0.1667
Our results					
6	NOWJ	nowj.d-ensemble	0.2757	0.2263	0.3527
6	NOWJ	nowj.ensemble	0.2756	0.2272	0.3504
6	NOWJ	nowj.bestsingle	0.2573	0.2032	0.3504

4.4 Task 3

Firstly, a simple statistical analysis has been conducted for determining the top-k candidates which could be gained from the BM25 model. The table 6 shows the top-k candidates and the corresponding recall score. Based on that result, the top 30 candidates are used for the training phase for reducing the training time but still remaining a good recall score and the top 500 is used for the inference phase.

In the re-ranking phase, the pre-train **bert-base-multilingual-uncased** [2] is employed as the back-bone and two heads corresponding to two learning tasks are added on top. In this investigation, two models are trained with different languages: English and Japanese in 10 epochs. The first model uses the Japanese data for the training phase and the second model mainly uses the English data.

The relevance score of re-ranking BERT is combined with the BM25 score for determining the final relevance score arcording to the equation.

For making the final submission, an ensemble process based on voting is conducted to combine the BM25-score, Multi-Task JP and Multi-Task EN model's relevance score, which achieves the third rank among all teams. The results of the ensemble method, along with the best results from other teams, on the 2023 private test are described in the Table 7. According to the results, our ensemble model achieves better *Precision* than the best run of the second team (JNLP3) but has lower *Recall*.

4.5 Task 4

The problem raised on task 4 is tackled by utilizing the textual entailment head of the multi-task model trained on previous task.

[2] https://huggingface.co/bert-base-multilingual-uncased

Table 6: Recall score of corresponding top-k

Top-k candidates	Recall score
30	0.7784
100	0.8513
200	0.8926
500	0.9487

Table 7: Results on the private test of Task 3

#	Team	Best Run	F2	P	R
Other team's result					
1	CAPTAIN	allEnssMissq	0.7569	0.7261	0.7921
2	JNLP	JNLP3	0.7451	0.6452	0.8218
4	HUKB	HUKB1	0.6725	0.6279	0.7079
5	LLNTU	LLNTUgigo	0.6535	0.7327	0.6436
6	UA	TFIDF_threshold2	0.5642	0.6205	0.5644
Our results					
3	NOWJ	nowj.d-ensemble	0.7273	0.6823	0.7673

Although a multi-task approach can significantly raise the performance on the retrieval task, it does not improve the result of task 4 yet. Further research could be conducted on the utilization of this model for the textual entailment problem. Our team has submited 3 runs for the private test including the first run named **NOWJ.multi-v1-en** employed the English data for the training phase, the second one named **Multi-Task EN (NOWJ.multi-v1-en)** used Japanese data for the training phase and the final run (**NOWJ.multijp**) also utilizes Japanese data with a different inference strategy. The accuracy of our runs and other team's best results is shown at the Table 8.

Table 8: Results on the private test of Task 4

#	Team	Best Run	Accuracy
Other team's result			
1	JNLP	JNLP3	0.7822
2	TRLABS	TRLABS_D	0.7822
3	KIS	KIS2	0.6931
4	UA	UA_V2	0.6634
5	AMHR	AMHR01	0.6535
6	LLNTU	LLNTUdulcsL	0.6238
7	CAPTAIN	CAPTAIN.gen	0.5842
8	HUKB	HUKB1	0.5545
Our results			
9	NOWJ	multi-v1-jp	0.5446
9	NOWJ	multijp	0.5248
9	NOWJ	multi-v1-en	0.4851

5 DISCUSSIONS

In this section, we discuss the overall performance of our approaches across all tasks in the COLIEE 2023 Competition and the lessons learned from our participation.

5.1 Task 1 and 2: Legal Case Retrieval and Entailment

Our approach for Tasks 1 and 2, which relied on BERT and Longformer pre-training models, demonstrated promising results in legal case retrieval and entailment. The use of lexical matching for candidate case retrieval provided a good balance between computational cost and recall performance. However, there is room for improvement in the precision of our results. The use of BERT and Longformer in the matching phase aimed to address this issue by considering important paragraphs of base cases in a pairwise comparison with candidate cases. Although our team ranked third in both tasks, this suggests that further optimization of the models or the incorporation of additional features could improve our system's performance.

5.2 Task 3: Statute Law Retrieval

Our two-phase retrieval system for Task 3, which employed the BM25-ranking algorithm and a BERT-based model for re-ranking, proved effective at narrowing down candidate articles while maintaining a high recall score. The multi-task learning approach combined retrieval and textual entailment tasks to improve the precision of our results. Our ensemble method, based on voting, achieved the third rank among all teams. However, there is still room for improvement in terms of precision and recall balance. Future work could explore other ensemble techniques or refine the multi-task learning model to boost performance.

5.3 Task 4: Legal Textual Entailment

In Task 4, we utilized the textual entailment head of the multi-task model trained from Task 3. While this approach did not lead to significant improvements in performance compared to other teams' best results, it demonstrated the potential applicability of multi-task learning models in legal textual entailment problems. Further research is needed to better understand the limitations of our current model and to identify opportunities for improvement, such as refining the model architecture or incorporating additional features.

5.4 Overall Lessons Learned

Our participation in the COLIEE 2023 Competition has provided valuable insights into the challenges and opportunities in legal information processing. We have gained experience working with state-of-the-art machine learning models and innovative approaches, while also identifying areas for future research and improvement. Some key lessons learned include:

- Understanding the importance of pre-processing and feature engineering in preparing data for legal information processing tasks.
- Recognizing the potential of pre-trained models like BERT and Longformer in addressing various legal information extraction and entailment challenges.
- Appreciating the effectiveness of multi-task learning models in combining different tasks to achieve better performance.

- Realizing the need for further research on domain-specific knowledge incorporation and model optimization to enhance our systems' performance in real-world legal scenarios.

6 CONCLUSIONS

In conclusion, our participation in the COLIEE 2023 Competition has allowed us to explore various state-of-the-art machine learning models and innovative approaches in the field of legal information processing. Although our team did not achieve the best results in all tasks, our performance in Tasks 1 and 2 with BERT and Longformer, as well as our two-phase retrieval system for Task 3 and the multi-task learning model for Task 4, demonstrated the potential of these methods in addressing real-world legal scenarios.

Our experience in this competition has provided valuable insights into the challenges and opportunities in legal information extraction and entailment. We believe that the methods and techniques we have employed can serve as a foundation for further research and improvements in this area. Future work could focus on refining the multi-task learning models, exploring other pre-training models, and incorporating domain-specific knowledge to enhance the performance of our systems.

Overall, our participation in the COLIEE 2023 Competition has been a fruitful learning experience, and we look forward to continuing our research on legal information processing and its applications in real-world scenarios.

ACKNOWLEDGMENTS

Hai-Long Nguyen was funded by the Master, PhD Scholarship Programme of Vingroup Innovation Foundation (VINIF), code VINIF.2022.ThS.050.

REFERENCES

[1] Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150* (2020).

[2] Ilias Chalkidis and Dimitrios Kampas. 2019. Deep learning in law: early adaptation and legal word embeddings trained on large corpora. *Artificial Intelligence and Law* 27, 2 (2019), 171–198.

[3] Thomas G Dietterich. 2000. Ensemble methods in machine learning. In *Multiple Classifier Systems: First International Workshop, MCS 2000 Cagliari, Italy, June 21–23, 2000 Proceedings 1*. Springer, 1–15.

[4] 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: JSAI-isAI 2022 Workshop, JURISIN 2022, and JSAI 2022 International Session, Kyoto, Japan, June 12–17, 2022, Revised Selected Papers*. Springer, 138–153.

[5] Mi-Young Kim, Juliano Rabelo, Kingsley Okeke, and Randy Goebel. 2022. Legal information retrieval and entailment based on bm25, transformer and semantic thesaurus methods. *The Review of Socionetwork Strategies* 16, 1 (2022), 157–174.

[6] Ha-Thanh Nguyen, Manh-Kien Phi, Xuan-Bach Ngo, Vu Tran, Le-Minh Nguyen, and Minh-Phuong Tu. 2022. Attentive deep neural networks for legal document retrieval. *Artificial Intelligence and Law* (2022), 1–30.

[7] Juliano Rabelo, Mi-Young Kim, and Randy Goebel. 2023. Semantic-based classification of relevant case law. 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, 84–95.

[8] Stephen E Robertson, Steve Walker, Susan Jones, Micheline M Hancock-Beaulieu, Mike Gatford, et al. 1995. Okapi at TREC-3. *Nist Special Publication Sp* 109 (1995), 109.

[9] 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. *Sixteenth International Workshop on Juris-informatics (JURISIN)* (2022).

[10] Guilherme Moraes Rosa, Ruan Chaves Rodrigues, Roberto Lotufo, and Rodrigo Nogueira. 2021. Yes, bm25 is a strong baseline for legal case retrieval. *arXiv preprint arXiv:2105.05686* (2021).

[11] 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*. 3501–3507.

[12] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30 (2017).

[13] Yen Thi-Hai Vuong, Quan Minh Bui, Ha-Thanh Nguyen, Thi-Thu-Trang Nguyen, Vu Tran, Xuan-Hieu Phan, Ken Satoh, and Le-Minh Nguyen. 2022. SM-BERT-CR: a deep learning approach for case law retrieval with supporting model. *Artificial Intelligence and Law* (2022), 1–28.

[14] 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.

IITDLI : Legal Case Retrieval Based on Lexical Models

Rohan Debbarma*
Indian Institute of Technology Delhi
India

Pratik Prawar*
Indian Institute of Technology Delhi
India

Abhijnan Chakraborty
Indian Institute of Technology Delhi
India

Srikanta Bedathur
Indian Institute of Technology Delhi
India

ABSTRACT

This paper describes in detail the methods used by IITDLI as a part of its submission to Competition of Legal Information Extraction and Entailment (COLIEE) 2023 for Task 1 (Legal Case Retrieval) and Task 2 (Legal Case Entailment). For Task 1, a retrieval pipeline consisting of term extraction, ranking using lexical model (BM25), year filter and post-processing of results produced excellent results. For Task 2, it was observed that zero shot Mono-T5 trained out of domain still outperforms other traditional and neural retrieval models. For Task 1, we have also explored how the different components of the pipeline incrementally contribute to the performance of the model. It is observed that year filter and term extraction are extremely crucial components of the pipeline sans which the Micro F1 dropped by more than 3 % in validation set. Our submission ranked 2nd among all teams for Task 1 and 4th among all teams for Task 2.

CCS CONCEPTS

• **Applied computing** → **Law**; • **Information systems** → *Retrieval models and ranking*; *Query representation.*

KEYWORDS

legal information retrieval, natural language processing, textual entailment

ACM Reference Format:
Rohan Debbarma, Pratik Prawar, Abhijnan Chakraborty, and Srikanta Bedathur. 2023. IITDLI : Legal Case Retrieval Based on Lexical Models. In *Proceedings of COLIEE 2023 workshop, June 19, 2023, Braga, Portugal.* ACM, New York, NY, USA, 8 pages.

1 INTRODUCTION

In the legal sector, the amount of data being generated is increasing day by day. Management and analysis of such an overwhelming amount of data manually becomes tedious. With the increase in the volume of legal documents, the demand for automated and semi-automated systems to help the legal professionals has also increased. In order to facilitate research in this field, the Competition of Legal Information Extraction and Entailment (COLIEE) was established. This competition focuses on four different aspects :- Case Law Retrieval, Case Law Entailment, Statute Law Retrieval and Legal Textual Entailment Data Corpus. In this paper, we provide insights and details regarding our approach to two of those problems, the Case Law Retrieval (Task 1) and the Case Law Entailment (Task 2) tasks.

In this case, the use of text retrieval systems in the field of legal domain becomes important. In both tasks, lawyers would only look at the top few(say, 20) cases that are retrieved by these systems. Hence, our approach to both these tasks is based on increasing the precision of retrieval. Our method for the case law retrieval task shows the effectiveness of query reformulation along with using a retrieval model like BM25, followed by post processing of the retrieved results using year filtering and answer selection method. In the case of the Case Law Entailment task, we have explored sparse retrieval models like BM25, as well as dense retrieval models like zero-shot T5 and GPT3.5 based reranker.

The rest of the paper is organised as follows: Section 2 contains the description of the two tasks explored in this paper. Section 3 focuses on the related work that has been done in this field. Section 4 presents our methods and results for the Case Law Retrieval task, which is then followed by Section 5 which focuses on our methods and results for the Case Law Entailment task. Section 6 concludes our work and comments on the future work that could be explored in this field.

2 TASK DESCRIPTION

2.1 Task 1:- The Case Law Retrieval Task

The Case Law Retrieval task consists of identifying the supporting cases of a given query case from a corpus of previous cases which can then be used to strengthen the decision for the given query case. Formally, we can say that given a query case Q and a set of cases s_1, s_2, s_3.... the task is to identify the previous cases S_1, S_2, S_3... which are relevant to the to the given query case Q.

2.2 Task 2:- The Case Law Entailment Task

In the Case Law Entailment Task, given a query paragraph from a base case along with another case, the task is to identify the paragraph from the second case which entails the query paragraph. Formally, given a paragraph q from the base case, and another case c which contains a set of paragraphs p_1, p_2, p_3...., the task is to identify the paragraphs P_1, P_2, P_3... which entails the decision of the paragraph q.

*Both authors contributed equally to this research.

COLIEE 2023, June 19, 2023, Braga, Portugal

Table 1: COLIEE 2023 Dataset Statistics

Task 1	Train	Test
# Query Case	959	319
# Candidate Case	4400	1335
Avg. # Noticed Case / Query	4.68	2.69
Task 2	**Train**	**Test**
# Query Case	625	319
# Candidate Para	734	120
Avg. # Entailed Para / Query	1.17	1.20

2.3 Dataset Description

The corpus for both Task 1 and Task 2 is drawn from judgements from Federal Court of Canada. Task 1 contains 959 queries in the training set and 319 queries in the test set. Task 2 contains 625 queries in training set and 100 queries in the test set. In the case of Task 1, all the query cases are part of the total collection of 4400 cases from which relevant cases are also to be retrieved

2.3.1 ***Dataset Statistics.*** Table 1 presents the details about the training and test dataset.

2.3.2 ***Evaluation Metrics.*** For Task 1 and Task 2, Micro-average of Precision, Recall and F1 score is used as an evaluation metric. The formula to calculate this metrics are as follows:

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

$$\text{Recall} = \frac{T.P.}{T.P. + F.N.}$$

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

where TP (True Positives) refers to the number of correctly retrieved cases for all queries, FP (False Positives) refers to the number of cases which are retrieved but are non-relevant and FN (False Negatives) refers to the cases which have not been retrieved but are part of the relevant set.

3 RELATED WORK

A lot of the previous work done in the field of legal case retrieval and entailment is based on the classical IR methods as well as some machine learning and deep learning based approaches. Some of the methods used in the previous iterations of COLIEE for Task 1 and Task 2 are described below:

UA [15] makes use of a transformer based model in order to generate paragraph level embeddings, which are then used to create feature vectors (using a 10-bin histogram) using similarity between the embeddings. It is then passed through a gradient-boosting classifier in order to classify the case as relevant or non-relevant.

LeiBi [2] uses query reformulation to shorten the query cases using methods like KLI, PLM and IDF-r. This reformulated query is then used to obtain an initial set of possible relevant cases. These cases are then reranked using different lexical and semantic models. In order to improve the effectiveness of the retrieval, the team aggregates the relevance scores obtained in the first stage and the reranking models.

nigam [11] combined the classical IR and transformer-based models. They first select a set of possible relevant cases using the BM25 model. This is then followed by creating sentence embeddings using sentence-BERT and sent2Vec. A score is then calculated using the cosine similarity between the query and the candidate case.

NeuralMind [18] uses the monoT5 model for Task2. They use monot5-base and monoT5-3B models and fine tune them for 10k steps. They then explore merging the results of the two models using their own answer selection method to select the final result from the two models.

Schilder et al. [22] generates an initial candidate set, trying to include most of the relevant cases. This is then followed by applying a classifier which classifies a particular case as being relevant to the given query case.

Rosa et al. [20] splits the query and the candidate cases into segments of 10 sentences. This is followed by the use of BM25 to retrieve candidate segments for each query segment. The relevance score of a candidate case is taken to be maximum of all the scores of the query case segment and candidate case segment pairs.

Althammer et al. [1] made use of neural IR models like BERT. In order to handle the 512 token limitation of BERT, the authors first applied the classical and dense retrieval methods at the paragraph level. This was then followed by summarizing the cases and applying a fine-tuned BERT re-ranker to these summaries.

4 TASK 1

4.1 Methodology

For the purpose of Task 1 (Case Law Retrieval), where the query case and the cases in the candidate corpus are of similar length and can be broadly categorised as long document retrieval since the average case document length is around 5000, our method mainly uses the sparse, traditional method of retrieval using bag of word features. In the previous editions of the COLIEE competition, various submissions [2, 11, 20] have shown the effectiveness of methods using similar kinds of methods using BM25 , DFR, etc in terms of good quality retrieval for this task.

Our retrieval pipeline mainly comprises of the following main steps:

(1) Extraction of unigram terms from the query using standard query reformulation techniques effectively shortening the query representations in the form of keywords.
(2) Retrieval using BM25 as a ranking model to retrieve the top-n results from the corpus
(3) Filtering of the results obtained based on a year filtering method
(4) Answer Selection Method based on [18] to dynamically retrieve different cited cases per query case to improve the overall micro-F1 score.

4.1.1 ***Text Pre-processing.*** For the purpose of the experiments, the training collection consisting of 989 queries was divided into a training and validation set on a 70-30 split, which resulted in 671 query cases in the training set and 288 query cases in the validation set. For the purposes of tuning the models in our pipeline, the validation set consisting of 288 queries was used. The main

steps performed in data pre-processing steps are mentioned in the following headings.

Removal of French words Since Canadian Case documents are often bilingual, containing both French and English, the French portion in the documents were identified and removed. It has previously shown to improve the performance of the traditional bag of words retrieval models as well as neural models in [1, 15]

The removal of French from the corpus was done with a pycld2 library which detects the probability of words belonging to a particular language. [13]

Year Extraction It is clear from the definition of a cited case that it must have been judged prior to the query case. It implies the most recent year mentioned in the cited case must be less than or equal to the most recent year mentioned in the query case. With this assumption or claim, the retrieved candidate cases were filtered out.

The years were extracted from the case documents using a regex pattern that detects all years between 1800 and 2023. The most recent year for each case in the corpus was initially assumed to be the year in which the case was judged and published. However, it was found during experiments on the validation set that instead of keeping the most recent year found through the regex pattern as the case year, it gave slightly better results if we kept (the most recent year found + 1) th year as the case year.

Feature Extraction For this task, only unigram/word features were used. The case documents were tokenized using the Word Tokenizer from nltk library. The <FRAGMENT_SUPPRESSED> tags present in the case which supposedly refer to various hidden citations were removed from the document. Various text normalization techniques such as stemming, lemmatization as well as stopword removal were experimented with for text cleanup purposes.

It was observed that only stopword removal had a positive impact on retrieval quality. Lemmatization and stemming had a slightly negative impact on the results. Hence, during tokenization, only stopword and punctuation removal were performed.

4.1.2 ***Term Extraction.*** In [2], the authors show the effectiveness of lexical based term extraction methods for this task. This is also consistent with the observation in [10] that keyword queries that are shorter in length as compared to the candidate corpus result in better quality retrieval.

For the purpose of query reformulation, the following two standard term scoring methods were experimented with: Kullback-Leibler Divergence for Informativeness (KLI) [2, 23] and Term Frequency and Inverse Document Frequency (TF-IDF).

KLI The KLI score for each term in a query case document was calculated using the formula used in [5].

$$KLI(t) = P(t|Q) \times \log \frac{P(t|Q)}{P(t|C)}$$

In this equation, P(t|Q) stands for the probability of a term t in the query document and P(t|C) stands for the probability of a term t in the whole candidate corpus. The probabilities of each individual term are calculated using their term frequencies in a case document.

TF-IDF The TF-IDF [21] score of each term is calculated using the above formula.

$$\text{tf-idf}(t, d) = \text{tf}(t, d) \times \text{idf}(t)$$

where

(1) t is a term in a document.
(2) d is a document.
(3) $\text{tf}(t, d)$ is the term frequency of term t in document d, which is the number of times t appears in d
(4) idf(t) is computed by the following formula:

$$\text{idf}(t) = \log_e \left[\frac{(1+N)}{(1+\text{df}(t))} + 1 \right]$$

where
(a) N is the total number of documents in the corpus.
(b) $df(t)$ is the number of documents in the corpus that contain term t.

4.1.3 ***Retrieval using BM25.*** The BM25 algorithm developed in the 1990s [17], based on a probabilistic term scoring model for bag of words style ad-hoc retrieval, has produced competitive results in various previous editions of COLIEE [2, 20]. The total BM25 score of a document is the sum of the contributions of each query term, which is also present in the candidate document. The equation to score each document using BM25 is as follows:

$$\text{BM25}(q, d) = \sum_{t \in q \cap d} \left[\text{idf}(t) \cdot \frac{tf(t,d) \cdot (k_1 + 1)}{tf(t,d) + k_1 \cdot \left(1 - b + b \cdot \frac{L_d}{L_{avg}}\right)} \right]$$

where

(1) q is the query.
(2) d is a candidate document.
(3) t is a term in both the query and the candidate document.
(4) idf(t) is computed using the following equation

$$\text{idf}(t) = \log \frac{N - df(t) + 0.5}{df(t) + 0.5}$$

where
(a) N is the total number of documents in the corpus.
(b) $df(t)$ is the number of documents in the corpus that contain term t.
(5) $tf(t, d)$ is the frequency of term t in document d.
(6) k_1 and b are tuning parameters.
(7) L_d is the length of document d.
(8) L_{avg} is the average document length in the corpus.

In this retrieval model, k1 and b are parameters that can be optimized according to the corpus. [2, 11] has shown that BM25 optimized for k1 and b produces significantly better results as compared to default parameters. Accordingly, the above parameters are tuned using the validation set to obtain the optimized k1 and b values.

For the BM25 implementation, we have used rank_bm25 library [4] which provides an implementation of the above formula. All the documents in the corpus are first indexed and then ranked for each query.

4.1.4 ***Year Filter.*** The cases obtained after initial retrieval from BM25 are then passed through a year filter such that all candidate cases with a higher value for the most recent year than the query case are removed.

Since on average there are 4.68 candidate cases per query in the training collection, it is assumed that similar statistics will hold

true for the test collection. So, for each query case, 5 candidate cases were extracted for two of the runs (run 1 and run 2). In the case of run3, an answer selection method based on score based thresholding was used to improve the overall F1 score.

4.1.5 ***Other Filters experimented with.*** In addition to the year filter, some other filters such as an Act based Filter as well as a Topical Model Identification based Filter were also experimented with. However, they produce slightly inferior results as compared to the runs submitted and are not a part of the submission made. They could be a direction to look into for future works with slight modifications.

Act based Filter Since the judgements are extracted from those of the Federal Court of Canada, all the Canadian Federal Acts were extracted. The corpus was then scanned using a regex pattern to extract those acts case-wise. The retrieved cases were then filtered out based on the presence of these acts assuming that both query and candidate cases must have at least one common act, provided the query case mentions at least one act.

Topic based Filter For this filter, a Latent Dirichlet Allocation (LDA) [3] based topical model was created for the training corpus, and the dominant topic(s) were identified for each case and noted down as metadata. Similar to the previous filter, the retrieved cases were filtered out on the assumption that the dominant topic would be the same in both the query and relevant case. The topic model implementation was done through the gensim library's LDA model [16].

For both of these filters, they were experimented with as both pre-processing (before retrieval using BM25) and post-processing (after fixing the top-5 cases per query) methods.

4.1.6 ***Post-processing of retrieved cases.*** Since 5 cases were being retrieved for each query case, it resulted in a high recall but low precision. To improve the precision and thereby the micro-F1, it was necessary to implement a thresholding scheme that produces different number of retrieved cases per query. This method was submitted as the run 3 for this task.

An answer selection method based on [18] was used to select the final set of candidate relevant cases for each query. The method involved the following three steps.

(1) Firstly, pick all cases having a BM25 score greater than x
(2) Secondly, pick top-y cases among them.
(3) Thirdly, pick all cases whose score is top- z percent of the score of the highest scoring case.

This answer selection method used by [18] had shown promising results for Task 2 in COLIEE 2022. In the context of this task (Task 1), we found that this method improved the F1 score on the validation set through a precision-recall tradeoff. It resulted in a higher precision but a lower recall value as compared to run 1 and 2 , but measured overall, it contributed to a higher F1 score.

4.2 Experiments

For the purpose of Task 1, the validation set consisting of 288 queries was used to tune the model hyper-parameters. For any retrieval task, the model parameters often play a crucial role in improving the overall results, as shown in previous COLIEE edition submissions such as [1, 2, 15],. The following main classes of hyper-parameters were optimized.

4.2.1 ***Term Portion.*** For the term extraction method, the percentage of top-n unigrams selected as a query representation or the term portion is a crucial parameter that affects retrieval quality. Combined with the lexical ranker (BM25), Figures 1 and 2 represent the variation in F1 score with term portion.

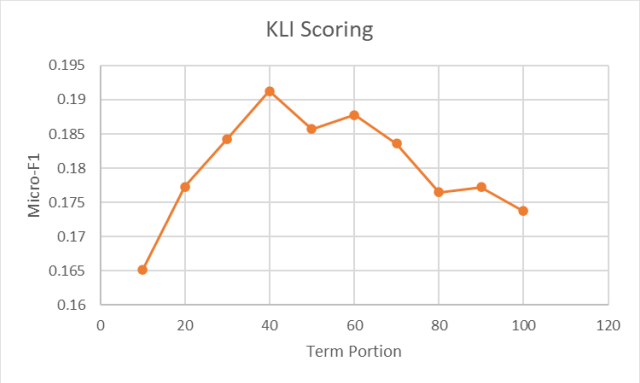


Figure 1: Results with BM25 opt for the validation set varying term portion in KLI scoring of terms.

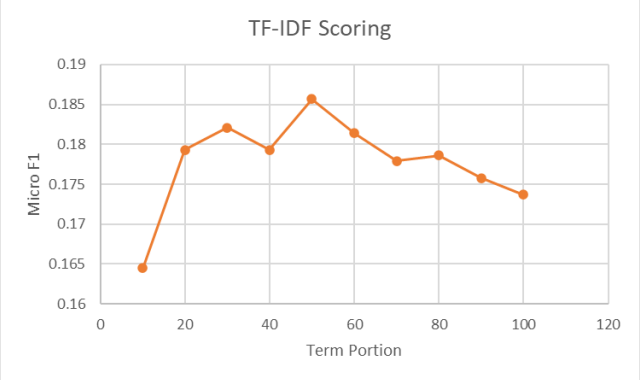


Figure 2: Results with BM25 opt for the validation set varying term portion in TF-IDF scoring of terms.

As we can see from Figures 1 and 2, BM25 is sensitive to the term portion values, and the peak value of F1 score occurs at 40 % term portion for KLI term scorer and 50 % term portion for TF-IDF term scorer. It points to the fact that moderately large query representations are perhaps the most effective queries for sparse models with bag of words corpora and query representations.

4.2.2 ***Parameters in BM25.*** By default, the parameters in BM25 are k1 = 1.5, b = 0.75. Grid Search was performed over the validation set with k1 = { 1, 1.5, 2, 2.5, 3, 3.5, 4, 4.5, 5 } and b = {0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1} A quick grid search on the k1 and b parameters revealed that the optimized value of b was 1 for this task. Figure 3 represents the variation of k1 keeping b as 1.

Table 2: Results of Task 1 Ablation Study on validation set

Method	F1	Precision	Recall
Best Performing Method	0.1975	0.2174	0.1809
Without FR	0.1974	0.2173	0.1809
Without YF	0.1607	0.1711	0.1514
Without TE	0.1652	0.1625	0.1678
Without PP	0.1913	0.1881	0.1945

It is clear from Figure 3 that for a corpus such as this one with a high average document length and more variation in term frequencies, higher values of k1 are preferred, which is consistent from the findings and observations in [6, 24]

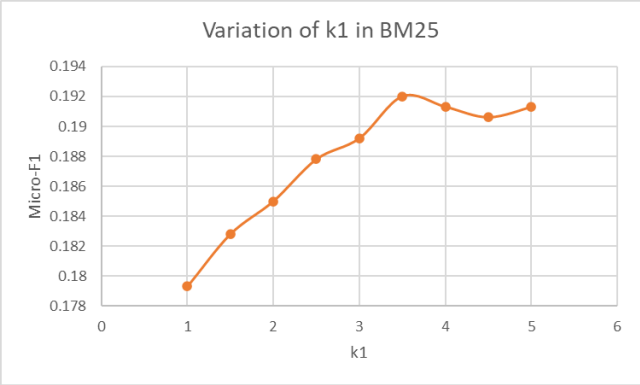


Figure 3: Results with BM25 for the validation set varying parameter k1 keeping b = 1.

4.3 Parameters in the Answer Selection Method

The paramters x, y, z in the answer selection method were tuned with the following values x = { 0, 100, 200, 300, ... , 1000 }, y = {1,2,3,... 25}, z = {0, 10, 20, 30, ...90, 91, 92, ... 99, 100}

Based on the grid search, the best parameters obtained were : x = 250, y = 5, z = 90

4.3.1 ***Ablation Study.*** In this section, we analyze the incremental impact of the various steps in the pipeline. For this analysis, we remove the components mentioned in the headers and keep all other components intact. The differences in results will signify to some extent, the partial contribution of that component to the overall pipeline. The following different components are studied in the ablation study:

The results of the ablation study are compiled in Table 2.

French Removal (FR) We can clearly see that French removal had an almost negligible but very small positive impact on the best performing method.

Year Filter (YF) It is clear that the year filter is a crucial component of the pipeline since it filters out many cases which may be similar in content to the query case but published after the query case. The Micro-F1 falls by 3 .7% as a result of removing the year filter, indicating its importance.

Table 3: Task 1 Test Results

Team	F1	Precision	Recall
THUIR	0.3001	0.2379	0.4063
THUIR	0.2907	0.2173	0.4389
IITDLI	0.2874	0.2447	0.3481
THUIR	0.2771	0.2186	0.3783
NOWJ	0.2757	0.2263	0.3527
NOWJ	0.2756	0.2272	0.3504
IITDLI	0.2738	0.2107	0.3912
IITDLI	0.2681	0.2063	0.3830
JNLP	0.2604	0.2044	0.3586
NOWJ	0.2573	0.2032	0.3504
UA	0.2555	0.2847	0.2317
UFAM	0.2545	0.2975	0.2224
JNLP	0.2511	0.1971	0.3458
JNLP	0.2493	0.1931	0.3516
UA	0.2390	0.3045	0.1967
UA	0.2345	0.2400	0.2293
UFAM	0.2345	0.3199	0.1851
UFAM	0.2156	0.3182	0.1630
YR	0.1377	0.1060	0.1967
YR	0.1051	0.0809	0.1502
LLNTU	0.0000	0.0000	0.0000
LLNTU	0.0000	0.0000	0.0000

Table 4: Important Results obtained on the validation set in various experiments

Method	F1	Precision	Recall
KLI opt + BM25 default	0.1737	0.1708	0.1766
KLI opt + BM25 opt	0.1913	0.1881	0.1945
TF-IDF opt + BM25 default	0.1652	0.1625	0.1680
TF-IDF opt + BM25 opt	0.1857	0.1826	0.1889
KLI opt + BM25 opt + Post-processing	0.1975	0.2174	0.1809

Term Extraction (TE) Removing the Term Extraction method is equivalent to only retrieving results using BM25, which uses all the tokens in the query case as query. It reduced the Micro-F1 by 3.2 % pointing to the fact that having term extraction component which shortens the queries helps in situations where the queries are extremely long.

Post-Processing of results (PP) The answer selection method or the post-processing improves the Micro-F1 significantly from 0.1913 to 0.1974.

4.4 Results

This section outlines the results obtained in different experiments on the validation set as well as the overall results of all submissions made in COLIEE'23 Task 1. Table 3 presents the results of all submissions made in COLIEE'23 Task 2. Table 4 presents some of the important results obtained on the validation set in our experiments.

4.4.1 ***Discussion of Results.*** IITDLI ranked 2nd in COLIEE'23 Task 1 with a Micro-F1 score of 0.2874. Our best performing method was a traditional lexical retrieval model with additional components such as year filter, term extraction and post-processing. This clearly shows the effectiveness of properly tuned lexical models such as BM25 in producing results that are close to state-of-the-art for Legal Case Retrieval which are characterized by long case queries. This has also been observed in previous versions of COLIEE for Task 1.

Among the term extraction methods, it was observed that KLI scoring was more robust to term portion variation and produced the best results when tuned. From the ablation study, it is quite clear that year extraction is also a crucial part of the retrieval pipeline since it improved the Micro-F1 score by 3 % for the validation set.

5 TASK 2

5.1 Methodology

For the purpose of Task 2 (Case Law Entailment), where both the query paragraphs or the entailed fragment for each case and the candidate case paragraphs are significantly shorter in length as compared to Task 1 queries, our method for Task 1 which involved term extraction. didn't produce good results since the queries are already very short. Therefore, the following three distinct methods were explored and experimented with in this task:

(1) BM25 based retrieval method where a separate corpus is created for each query case with distinct paragraphs serving as individual documents to index
(2) Zero Shot T5 model which produces para-para relevance scores. These zero shot T5 models have shown effectiveness in producing excellent results for this task [18].
(3) GPT3.5 based reranker which reranks the top-10 retrieved paragraphs from BM25

5.1.1 ***Text Preprocessing.*** The text pre-processing and cleanup steps were the same for both Task 1 and Task 2 except for the year extraction part. Since the query and candidate paragraphs belong to the same case, the year extraction component to filter by recent year had no relevance.

5.2 Retrieval using BM25

The details of the BM25 algorithm have already been discussed in detail in previous sections.

For this task, separate BM25 models were created for each query. In other words, the background collection for each of the models is : the paragraphs in that query case only. The parameters k1 and b were tuned using the validation queries.

5.2.1 ***Zero Shot MonoT5 retrieval.*** MonoT5 which is an adaptation of the T5 model [12] and fine-tuned on the MS Marco passage dataset to generate "true" or "false" tokens based on the relevance of a passage pair is used to estimate the relevance of the query and candidate paragraphs in this task. This has been explored previously in COLIEE competitions in [18] and has shown promising results. The authors in [19] had shown that with an increasing number of model parameters, the results improved, which is consistent with our observations.

During inference, this model uses the following template :

query : q doc : d relevant:

Here, q is entailed paragraph and d is one of the candidate paragraphs. The model predicts a score, which is the probability of the token "true" being assigned to this template. All the candidate paragraphs are then ranked according to the score in decreasing order of relevance score.

As one of the runs, this method was explored and the variation of F1 score with different model sizes was also experimented with. To implement the zero shot mono-T5 model with different parameters, the pygaggle library built upon pyserini [9] was used.

5.2.2 ***GPT3.5 based reranker.*** Recently, large language models such as GPT have shown great promise in many zero-shot and few-shot retrieval cases [7, 14]. As a part of the 3rd run, a reranker based on the GPT3.5 turbo API from OpenAI is used to rerank the top-10 results retrieved by BM25. Since the Recall@10 from BM25 method for the validation set was more than 90 %, it made sense to develop a 2 stage retrieval pipeline : BM25 retrieval, which optimizes for recall, followed by a GPT3.5 based reranker based on prompt engineering.

As a part of the implementation, a prompt was fed to the GPT3.5-turbo model which consisted of the text from all the top-10 paragraphs along with a instruction to return a ranked list of paragraphs with a confidence score signifying the relevance of the entailed query fragment to the candidate paragraph.

However, one of the challenges of this method based on GPT3.5 is reproducibility. Since the output is in the form of text tokens, it is also necessary to develop a parser that can handle slight variations in the output formats from the reranker.

5.3 Experiments

In the case of Task 2, where three distinct methods were used in separate runs, the hyper-parameters varied in each method were different. Some of those experiments are described here.

In this task, since the average number of relevant paragraphs per query is 1.17, only the top-ranked paragraph among all the paragraphs was retrieved as the result set for each query.

5.3.1 ***Parameters in BM25.*** The parameters k1 and b in BM25 are tuned with the following values. k1 = { 1, 1.1, 1.2, 1.3, ... 4.7, 4.8, 4.9, 5.0 }, b = {0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1}

Based on this grid search, the best parameters obtained were for k1 = 1.5 and b = 0.7.

5.3.2 ***Number of Model Parameters in Mono-T5.*** In this section, the Mono-T5 models described in [12] with different number of model parameters are evaluated on Task 2. Table 5 compiles all the results regarding the T5 models with different parameters.

From the results, we can clearly observe that having a higher number of parameters improves the zero shot retrieval quality of T5 consistent with the observations in [19].

5.4 Results

This section outlines the results obtained in different runs on the validation set as well as the overall results. Table 6 presents the results of all submissions made in COLIEE'23 Task 2. Table 7 presents the results obtained for different runs on the validation set in our experiments.

Table 5: Variation in evaluation metrics with the number of model parameters in zero shot Mono-T5

Model	# Params	F1	Precision	Recall
MonoT5-base	220M	0.6816	0.5804	0.6269
MonoT5-large	770M	0.6896	0.5872	0.6343
MonoT5-3b	3000M	0.7088	0.6035	0.6519

Table 6: Task 2 Test Results

Team	F1	Precision	Recall
CAPTAIN	0.7456	0.7870	0.7083
CAPTAIN	0.7265	0.7864	0.6750
THUIR	0.7182	0.7900	0.6583
CAPTAIN	0.7054	0.7596	0.6583
THUIR	0.6930	0.7315	0.6583
JNLP	0.6818	0.7500	0.6250
IITDLI	0.6727	0.7400	0.6167
JNLP	0.6545	0.7200	0.6000
UONLP	0.6387	0.6441	0.6333
THUIR	0.6091	0.6700	0.5583
NOWJ	0.6079	0.6449	0.5750
NOWJ	0.6036	0.6569	0.5583
NOWJ	0.5982	0.6442	0.5583
IITDLI	0.5304	0.5545	0.5083
JNLP	0.5182	0.5700	0.4750
IITDLI	0.5091	0.5600	0.4667
LLNTU	0.1818	0.2000	0.1667
LLNTU	0.1000	0.1100	0.0917

Table 7: Important Results obtained on the validation set in various experiments for Task 2

Method	F1	Precision	Recall
BM25 opt	0.6528	0.5558	0.6004
MonoT5-3b	0.7088	0.6035	0.6519
GPT3.5 Reranker	0.5529	0.5157	0.5336

5.4.1 ***Discussion of Results.*** For Task 2, our team ranked 4th among all the teams with a best Micro-F1 score of 0.6727. Our best performing run used zero shot Mono-T5 which was also the best performing method in COLIEE'22 Task 2 [8]. Our 2nd best performing method was a GPT3.5 based reranker that showed promise in terms of retrieving some correct relevant cases in which T5 fails. However, compared overall, it performs significantly worse than that of T5 based method. For entailment tasks such as Task 2 where queries are paragraphs (shorter in length than Task 1 queries by quite a few orders of magnitude), it can be opined that large language models with billions of parameters that has been trained out-of-domain, such as Mono-T5 with 3b parameters have performed significantly well as compared to traditional lexical models like BM25.

6 CONCLUSION

In this work, we have described a retrieval pipeline combining components such as year filter, term extraction, and post-processing along with a lexical retrieval model (BM25) for Legal Case Retrieval. We have also discussed in detail the incremental impact of each of these components and how they combine to produce close to state-of-the-art results for Task 1. The result also shows that BM25 is a good baseline for this task. At the same time, this task also presents a novel challenge of tackling extremely long queries and candidate cases, which negatively affects the effectiveness of both neural and lexical models.

For the Legal Entailment Task (Task 2), we compared the results of 3 different methods. Consistent with COLIEE'22 [8] edition, our method that produced the highest F1 score was a zero shot Mono-T5 model trained on billions of parameters. Another run of ours, GPT3.5 based reranker showed some promise in retrieving a few relevant paragraphs in which MonoT5 failed. However, compared overall, the results for the reranker are significantly worse.

In the future, we would like to explore whether second stage retrieval using neural rankers and post-processing would be useful for Tasks 1 and 2.

ACKNOWLEDGMENTS

We thank the organizers of COLIEE 2023 for providing us access to the data for Task 1 and Task 2. A. Chakraborty gratefully acknowledges the support of the DAKSH Centre of Excellence (CoE) for Law and Technology at Indian Institute of Technology Delhi.

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] Arian Askari, Georgios Peikos, Gabriella Pasi, and Suzan Verberne. 2022. LeiBi@COLIEE 2022: Aggregating Tuned Lexical Models with a Cluster-driven BERT-based Model for Case Law Retrieval. In *Sixteenth International Workshop on Juris-informatics(JURISIN).*

[3] David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *Journal of machine Learning research* 3, Jan (2003), 993–1022.

[4] Dorian Brown. 2020. *Rank-BM25: A Collection of BM25 Algorithms in Python.* https://doi.org/10.5281/zenodo.4520057

[5] Claudio Carpineto, Renato De Mori, Giovanni Romano, and Brigitte Bigi. 2001. An information-theoretic approach to automatic query expansion. *ACM Transactions on Information Systems (TOIS)* 19, 1 (2001), 1–27.

[6] Shane Connelly. 2019. Practical BM25 - part 3: Considerations for picking B and K1 in Elasticsearch. https://www.elastic.co/blog/practical-bm25-part-3-considerations-for-picking-b-and-k1-in-elasticsearch

[7] Zhengbao Jiang, Jun Araki, Haibo Ding, and Graham Neubig. 2021. How can we know when language models know? on the calibration of language models for question answering. *Transactions of the Association for Computational Linguistics* 9 (2021), 962–977.

[8] 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.* Springer, 51–67.

[9] 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.* 2356–2362.

[10] Daniel Locke, Guido Zuccon, and Harrisen Scells. 2017. Automatic Query Generation from Legal Texts for Case Law Retrieval. 181–193. https://doi.org/10.1007/978-3-319-70145-5_14

[11] Shubham Kumar Nigam and Navansh Goel. 2022. nigam@COLIEE-22: Legal Case Retrieval and Entailment using Cascading of Lexical and Semantic-based

models. In *Sixteenth International Workshop on Juris-informatics(JURISIN)*.

[12] Rodrigo Nogueira, Zhiying Jiang, Ronak Pradeep, and Jimmy Lin. 2020. Document Ranking with a Pretrained Sequence-to-Sequence Model. In *Findings of the Association for Computational Linguistics: EMNLP 2020*. Association for Computational Linguistics, Online, 708–718. https://doi.org/10.18653/v1/2020.findings-emnlp.63

[13] Jeroen Ooms. 2023. *cld2: Google's Compact Language Detector 2*. https://github.com/cld2owners/cld2

[14] Raul Puri, Ryan Spring, Mostofa Patwary, Mohammad Shoeybi, and Bryan Catanzaro. 2020. Training question answering models from synthetic data. *arXiv preprint arXiv:2002.09599* (2020).

[15] Juliano Rabelo, Mi-Young Kim, and Randy Goebel. 2022. Semantic-based Classification of Relevant Case Law. In *Sixteenth International Workshop on Juris-informatics(JURISIN)*.

[16] Radim Rehurek and Petr Sojka. 2011. Gensim–python framework for vector space modelling. *NLP Centre, Faculty of Informatics, Masaryk University, Brno, Czech Republic* 3, 2 (2011).

[17] Stephen E Robertson, Steve Walker, Susan Jones, Micheline M Hancock-Beaulieu, Mike Gatford, et al. 1995. Okapi at TREC-3. *Nist Special Publication Sp* 109 (1995), 109.

[18] Guilherme Rosa, Luiz Henrique Bonifacio, Vitor Jeronymo, Hugo Queiroz 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. https://doi.org/10.48550/arXiv.2205.15172

[19] 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.

[20] Guilherme Moraes Rosa, Ruan Chaves Rodrigues, Roberto de Alencar Lotufo, and Rodrigo Nogueira. 2021. Yes, BM25 is a Strong Baseline for Legal Case Retrieval. In *: Proceedings of the COLIEE Workshop in ICAIL*.

[21] Gerard Salton, Anita Wong, and Chung-Shu Yang. 1975. A vector space model for automatic indexing. *Commun. ACM* 18, 11 (1975), 613–620.

[22] 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*.

[23] Suzan Verberne, Maya Sappelli, Djoerd Hiemstra, and Wessel Kraaij. 2016. Evaluation and analysis of term scoring methods for term extraction. *Information Retrieval Journal* 19 (2016), 510–545.

[24] Chengxiang Zhai and John Lafferty. 2004. A study of smoothing methods for language models applied to information retrieval. *ACM Transactions on Information Systems (TOIS)* 22, 2 (2004), 179–214.

Transformer-based Legal Information Extraction

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

Juliano Rabelo
Alberta Machine Intelligence Institute, University of Alberta
Edmonton, AB, Canada
rabelo@ualberta.ca

Housam Babiker
Dept. of Computing Science, University of Alberta
Edmonton, AB, Canada
khalifab@ualberta.ca

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

ABSTRACT

The challenge of information overload in the legal domain increases every day. The COLIEE competition has created four challenge tasks which are intended to encourage the development of systems and methods to alleviate some of that pressure: a case law retrieval (Task 1) and entailment (Task 2), and a statute law retrieval (Task 3) and entailment (Task 4). In this paper, we describe our methods for Task 1 and Task 4. In Task 1 we used a sentence-transformer model to create a numeric representation for each case paragraph, then create a histogram of the similarities between a query case and a candidate case. The histogram is then submitted to a binary classifier which decides whether that candidate case should be noticed or not. Some postprocessing heuristics are also applied. Our method for Task 4 was ranked third among eight participating teams in the COLIEE 2023 competition. Our approach relies on fine-tuning a pre-trained DeBERTa large language model trained on SNLI and MultiNLI datasets.

CCS CONCEPTS

• **Information systems** → **Content analysis and feature selection**; **Similarity measures**; **Clustering and classification**; *Document topic models*; *Information extraction*; Specialized information retrieval.

KEYWORDS

legal textual entailment, document similarity, binary classification, imbalanced datasets

ACM Reference Format:
Mi-Young Kim, Juliano Rabelo, Housam Babiker, and Randy Goebel. 2023. Transformer-based Legal Information Extraction. In *Proceedings of COLIEE 2023 workshop, June 19, 2023, Braga, Portugal.* ACM, New York, NY, USA, 5 pages.

ICAIL '23, June 19–23, 2023, Braga, Portugal

1 INTRODUCTION

Every day, large volumes of legal data are produced by law firms, law courts, independent attorneys, legislators, regulators and many others. Within that context, the disciplined management of legal information becomes manually intractable, and requires the development of tools which automatically or semi-automatically aid legal professionals to address the information overload. The COLIEE competition[1] addresses four facets of that challenge: case law retrieval, case law entailment, statute law retrieval and statute law entailment. Here we summarize the details of our approach to the case law retrieval and statute law entailment, evaluate the results achieved and comment on future work to further improve our models.

The case law retrieval task (Task 1) involves identifying legal cases that should be "noticed" with respect to a given query case from amongst a given set of candidate cases. Our approach to this challenging task relies on a transformer-based model that creates a multidimensional numeric representation of each paragraph within an individual case. We calculate cosine distances between each paragraph of a query case and a candidate case, create a histogram from the results, and use those distances to train a binary classifier to determine whether an input case should be noticed. Additionally, in the context of the COLIEE datasets, we perform some simple pre-processing and post-processing steps, such as removing French fragments and applying a minimum confidence score for the classifier outputs, to generate the final results.

The goal of the statute law entailment (Task 4) is to construct yes/no question answering systems for legal queries, by confirming entailment of a query from relevant articles. The answer to a question is typically determined by measuring some kind of semantic similarity between question and answer. Because the legal bar exam query and relevant articles are complex and varied, we need to carefully determine what kind of information is needed for confirming textual entailment. Here we exploit the idea of natural language inference and fine-tune a DeBERTa-large language model to construct a yes/no question answering system for legal queries.

Our approach for Task 4 relies on a transformer (DeBERTa)-based model to construct a classifier for yes/no questions. The DeBERTa model was initially trained for NLI using two dataset namely SNLI [2] and MultiNLI [16]. In addition, to standardize all the inputs

[1] https://sites.ualberta.ca/ rabelo/COLIEE2023/

to the model, we provide all lowercase sentences to the model to generate the final results. This approach achieved an accuracy of 0.6634 in the official test dataset, which was ranked third amongst eight competitors in Task 4 of the COLIEE 2023 competition.

Our paper is organized as follows: Section 2 presents a brief state-of-the-art analysis; Sections 3 and 4 describe our method in more detail; Section 5 analyzes the results; and Section 6 provides some final remarks and proposes some future work.

2 LITERATURE REVIEW

Most current approaches to legal information retrieval rely on traditional information retrieval (IR) methods, and more recently, transformer-based techniques. We will briefly summarize below some of the most successful approaches proposed in recent editions of COLIEE.

TR. [15] uses a two-phase approach to legal case document information retrieval. In the first phase, they generate a candidate set optimized for recall, aiming to include all true noticed cases while removing some false candidates. In the second phase, they train a binary classifier to predict whether a given (*query case*, *candidate case*) pair represents a true noticed relationship. In this step, these authors experimented with logistic regression, naive Bayes, and tree-based classifiers.

NeuralMind. [14] applied "vanilla" BM25 to the case law retrieval problem. The authors first indexed all base and candidate cases in the dataset. Prior to indexing, each document was split into segments of text using a context window of 10 sentences with overlapping strides of 5 sentences (the "candidate case segments"). BM25 was then used to retrieve candidate case segments for each base case segment. The relevance score for a (*base case*, *candidate case*) pair was the maximum score among all the base case and candidate case segment pairs. The candidates were then ranked using empirically determined threshold heuristics.

TUWBR. This team [5] starts from two assumptions: first, that there is a topical overlap between query and notice cases, but that *not all parts of a query case are equally important.* Secondly, they assume that traditional IR methods, such as BM25, provide competitive results in Task 1. They perform both document level and text passage level retrieval, and also augment the system by adding external domain knowledge by extracting statute fragments and explicitly adding those fragments to the documents.

JNLP. [3] applies an approach that first splits the documents into paragraphs, then calculates the similarities between cases by combining term-level matching and semantic relationships at the paragraph level. An attention model is applied to encode the whole query in the context of candidate paragraphs, which is then used to infer the relationship between cases.

DoSSIER. [1] combined traditional and neural network-based techniques in Task 1. The authors investigate lexical and dense first stage retrieval methods aiming for a high recall in the initial retrieval and then compare shallow neural-network re-ranking between the MTFT-BERT model and the BERT-PLI model. They then investigate which part of the text of a legal case should be taken into account for re-ranking. Their results show that BM25 shows a consistently high effectiveness across different test collections in comparison to the neural network re-ranking models.

Task 1 has been recently adjusted in COLIEE. The new configuration increased the difficulty so that the heretofore typical Information Retrieval methods, even augmented with transformers-based approaches, did not show great results [10]. Given that most of the current approaches work at the document level, we intended to experiment with the documents at the sentence level to try and capture more localized information. More details of the approach are presented in section 3.

Textual entailment, which is also called Natural Language Inference (NLI), is a logic task in which the goal is to determine whether one sentence can be inferred from another (more generally, whether one text segment can be inferred from another).

In the sentential case, the task consists of classifying an ordered pair of sentences into one of three categories: "positive entailment" occurs when one can use the first sentence to prove that a second sentence is true. Conversely, "negative entailment" occurs when the first sentence can be used to disprove the second sentence. Finally, if the two sentences have no correlation, as determined by failure of the first two tests, they are considered to have a "neutral entailment."

The statute law entailment task (Task 4) in COLIEE is similarly designed: the participants are required to decide if a query is entailed from the relevant civil law statutes.

This NLI task of identifying whether a hypothesis can be inferred from, contradicted by, or not related to a premise, has become one of the standard benchmark tasks for natural language understanding. NLI datasets are typically built by asking annotators to compose sentences based on premises extracted from corpora, so that the composed sentences stand in entailment/contradiction/neutral relationship to the premise [9]. In COLIEE 2023, we have two relationships that need to be verified: entailment and non-entailment.

For this problem, we rely on the base DeBERTa model [7] which is an extension of the original BERT model. The DeBERTa-based model was trained on large volumes of raw text corpora using the idea of self-supervised learning. As compared to the original BERT model, DeBERTa captures more fine-grained contextual information and relationships between tokens, resulting in a significant performance gain on a wide range of Natural Language Understanding (NLU) tasks. Deep learning methods have enabled the construction of complex and accurate models for the NLI task [13]. Most current approaches are formulated as a 3-way classification (entailment, contradiction, and neutral) of the entailment relation between a pair of sentences. When approached with logical principles, this task requires a sophisticated semantic framework to understand the context for the two sentences (premise, hypothesis).

3 OUR METHOD-TASK 1

3.1 Dataset Analysis

In Task 1, the training dataset consists of 4,400 files, with 959 of those identified as query cases. There are a total of 4,488 noticed cases, an average of 4.67 noticed cases per query case. In the provided test dataset there were 319 query cases and a total of 1,335 files.

3.2 Details of our Approach

Our approach to the case law retrieval task relied on the use of a sentence-transformer model to generate a multidimensional numeric representation of text. This model is applied to each paragraph from both the query case and every candidate case. We then use a cosine measure to determine the distances between the 768-dimension vectors from the query paragraphs and the candidate paragraphs. A 10-bin histogram of those distances is generated and a Gradient Boosting [6] binary classification model is trained on those inputs.

Given the formulation of the problem, we had to make some choices to produce a manageable training dataset: since the test set contains a total of approximately 1,300 of which 319 are query cases, we assumed we should generate a training dataset with around 1,000 negative samples per query case. So we needed to down-sample the negative samples in the training dataset to 1,000. At the same time, the positive class is significantly underrepresented (less than 5 samples per query case in average), so we over-sampled those examples by simple replication.

We also implemented some simple pre-processing steps:

Removal of French contents. through a language identification model based on a naive Bayesian filter [11];

Splitting of input text into paragraphs. based on simple pattern matching which relies on the common format used in cases. This method relies on finding a sequence of numbered paragraphs (specified as digits between brackets) as the first characters in the line starting at "[1]" and looking for the next natural number;

Extraction of dates mentioned in the cases. is done with by the application of a named entity recognition model [8]. These dates are later used to remove candidate cases which mention dates more recent than the most recent date mentioned in a query case, under the assumption those candidates cannot be a true noticed case because they are more recent than the query case. So, we basically extract all date entities in both the query and the candidate case and if the query case contains a date which is more recent than the most recent date in the candidate case, that candidate case will be removed from the list.

At inference time we used the following steps:

Date filtering: we apply the same date pre-processing steps mentioned above;

Histograms. : we generate histograms for every pair of query document and each candidate which does not contain dates more recent than the query document dates;

Apply model: we use those histograms as inputs to our trained classification model.

Based on our analysis of the training dataset, we also apply some simple post-processing steps:

Number of noticed cases per query case: the average number of noticed cases per query case in the training dataset is 4.67, so we establish a range of 3 to 10 maximum noticed cases per query case;

Confidence score: we establish a minimum confidence score for the classifier, disregarding outputs which are below a given threshold;

Repeating noticed cases: if the same case is noticed across many different query cases, we also remove that noticed case from our final answer as it is observed in the training dataset that this is an uncommon situation.

We experimented with a range of parameters for each one of those post-processing criteria and selected the 3 combinations which produced the best output in a validation set containing 50 query cases[2].

3.2.1 Sentence-Transformer Model. The model used to produce the 768-dimensional representations for the case paragraphs was the HuggingFace sentence-transformers/all-mpnet-base-v2 model[3]. That model was trained on very large sentence level datasets using a self-supervised contrastive learning objective, which used the pretrained Microsoft/mpnet-base model[4] as the base model and fine-tuning it on a 1B sentence pairs dataset. The authors use a contrastive learning objective: given a sentence from the pair, the model should predict which of a set of randomly sampled other sentences was actually paired with it in the dataset.

3.2.2 Binary Classification Model. The method used for training was a Gradient Boosting model [6] which was trained on the calculated similarity histograms as described above. Since the training dataset is significantly unbalanced, we oversample the positive class by simple duplication, and undersample the negative class by establishing a target maximum number (which was chosen as 1,000 samples). The only hyper-parameter we varied in the classifier itself was the number of estimators, which was set to 1,000, 3,000 and 5,000.

3.2.3 Hyper-parameter Setting. We performed a grid search for 3 hyper-parameters:

- Maximum number of noticed cases per query case: based on the dataset analysis performed, given the average number of noticed cases per query case in the training set is around 5, we experimented with establishing a limit which varied from 3 to 10 (step 1) in an attempt to reduce the false positives;
- Minimum confidence score: we trained a binary classifier to determine if a given case should be noticed with respect to a given query case. With this hyper-parameter we can filter candidate cases for which the classifier confidence score is below a given threshold. We experimented with values from 0.55 to 0.80 (step 0.05);
- Maximum duplicate noticed cases: we noticed in our validation results that the same case was classified as noticed with respect to more than one query case, which is not common in the training dataset, so we establish the maximum number of times the same case can be present in the output. This parameter was varied from 1 to 5 (step 1).

[2]The validation set was randomly drawn from the provided dataset and has no overlap with the cases used for training.
[3]https://huggingface.co/sentence-transformers/all-mpnet-base-v2
[4]https://huggingface.co/microsoft/mpnet-base?text=The+goal+of+life+is+%3Cmask%3E.

Team	F1	Precision	Recall
THUIR	0.3001	0.2379	0.4063
THUIR	0.2907	0.2173	0.4389
IITDLI	0.2874	0.2447	0.3481
THUIR	0.2771	0.2186	0.3783
NOWJ	0.2757	0.2263	0.3527
NOWJ	0.2756	0.2272	0.3504
IITDLI	0.2738	0.2107	0.3912
IITDLI	0.2681	0.2063	0.3830
JNLP	0.2604	0.2044	0.3586
NOWJ	0.2573	0.2032	0.3504
UA	*0.2555*	*0.2847*	*0.2317*
UFAM	0.2545	0.2975	0.2224
JNLP	0.2511	0.1971	0.3458
JNLP	0.2493	0.1931	0.3516
UA	*0.2390*	*0.3045*	*0.1967*
UA	*0.2345*	*0.2400*	*0.2293*
UFAM	0.2345	0.3199	0.1851
UFAM	0.2156	0.3182	0.1630
YR	0.1377	0.1060	0.1967
YR	0.1051	0.0809	0.1502
LLNTU	0.0000	0.0000	0.0000
LLNTU	0.0000	0.0000	0.0000

Table 1: Official results for the Case Law Retrieval task.

Table 2: NLI (Task 4) results on test data.

Team	sid	Correct	Accuracy
	BaseLine	No 52/All 101	0.5149
JNLP	JNLP3	79	0.7822
JNLP	JNLP1	76	0.7525
JNLP	JNLP2	76	0.7525
KIS	KIS2	70	0.6931
KIS	KIS1	68	0.6733
UA	UA_V2	67	0.6634
AMHR	AMHR01	66	0.6535
KIS	KIS3	66	0.6535
AMHR	AMHR03	65	0.6436
LLNTU	LLNTUdulcsL	63	0.6238
UA	UA	63	0.6238
HUKB	HUKB2	60	0.5941
CAPTAIN	CAPTAIN.gen	59	0.5842
CAPTAIN	CAPTAIN.run1	58	0.5743
LLNTU	LLNTUdulcsS	57	0.5644
HUKB	HUKB1	56	0.5545
HUKB	HUKB3	56	0.5545
LLNTU	LLNTUdulcsO	56	0.5545
NOWJ	NOWJ.multi-v1-jp	55	0.5446
CAPTAIN	CAPTAIN.run2	53	0.5248
NOWJ	NOWJ.multijp	53	0.5248
NOWJ	NOWJ.multi-v1-en	49	0.4851

The 3 best performing hyper-parameter combinations were used in our submission. You can see that the best (max noticed cases = 10, min score = 0.8, max dups = 3) achieved good precision but poor recall. The second best combination (9, 0.7, 2) had an even higher precision, but a very poor recall. We attribute this to the effect of the minimum confidence score, which was higher in this case, whereas the other parameters were pretty much the same. Even though the difference in the final f1-score wasn't material, having the ability to tweak parameters and influence precision and recall would be a good feature of the method in real-world applications, where users could adopt parameters according to their requirements with respect to precision and recall.

4 OUR METHOD-TASK 4

In Task 4, the problem of answering a legal yes/no question can be viewed as a binary classification problem. We assume that a set of questions Q, where each question $q_i \in Q$ is associated with a list of corresponding article sentences $a_{i1}, a_{i2}, \ldots, a_{im}$, where $y_i = 1$ if the answer is 'yes' and $y_i = 0$ otherwise. We choose the most relevant sentence a_{ij}. Therefore, our task is to learn a classifier over these triples so that it can predict the answers of any additional question-article pairs. BERT [4] has shown good historical performance in both COLIEE and in general on the natural language inference tasks. However, Jiang and Marnaffe [9] insisted that despite high F1 scores, BERT models have systematic error patterns, suggesting that they still do not capture the full complexity of human pragmatic reasoning.

We reformulate the problem as a natural language inference task, where the objective of the model is to determine the logical relationship between a premise and a hypothesis (e.g., whether the hypothesis entails, contradicts or is neutral with respect to the given premise). In the case of answering a legal yes/no question, if the NLI model predicts the relationship as entailment, we then consider the prediction as 'yes' otherwise 'no.'

To construct the training data from the NLI model fine-tuning, we modify the ground-truth labels. For questions with a ground-truth label of 'yes,' we change them to 'entailment,' and for questions with a label of 'no,' we change them to 'contradiction.' Because we have two inputs, before making a prediction we follow the procedure proposed by [12], i.e., we concatenate the sentence embedding u and v from input 1 and 2 respectively, and then use the element-wise difference $|u - v|$ and multiply it with the trainable weight W.

We then fine-tune the model by minimizing the cross-entropy loss over the labeled training data to penalize incorrect classification.

5 RESULTS

5.1 Task 1 Results

The results on the official COLIEE evaluation set are shown in Table 1:

Our best result was achieved with the following post-processing parameters: minimum confidence score = 0.80, maximum noticed cases = 10, maximum number of repeated noticed cases = 3[5]. Our second best score had similar parameters (0.7, 9 and 2 respectively). In the third submission we used 0.65, 10 and 3 respectively). This

[5]We simply remove noticed cases which appear in more than the maximum allowed query cases. An obvious improvement is to keep just the highest scoring noticed cases.

Table 3: NLI (Task 4) results on test data considering only the best system in each team

Team	sid	Correct	Accuracy
	BaseLine	No 52/All 101	0.5149
JNLP	JNLP3	79	0.7822
KIS	KIS2	70	0.6931
UA	UA_V2	67	0.6634
AMHR	AMHR01	66	0.6535
LLNTU	LLNTUdulcsL	63	0.6238
HUK	HUKB2	60	0.5941
CAPTAIN	CAPTAIN.gen	59	0.5842
NOWJ	NOWJ.multi-v1-jp	55	0.5446

provided a more balanced trade-off between precision and recall, as opposed to the first two which had a higher precision but a lower recall. This is an interesting characteristic for real world applications, as one could make an informed decision on how to tweak parameters depending on which metric is more important for their particular scenario.

5.2 Task 4 Results

Table 2 shows the Task 4 results on test data in COLIEE 2023. We submitted two results, UA_V1 fine-tuned on DeBERTa-small [7] and UA_V2 fine-tuned on DeBERTa-large model[12]. In the table, we report the performance of the best model i.e., UA_V2.

The test results considering only one best system in each team are in Table 3:

We found that the current model struggles in predicting the correct class "yes" which requires a deep understanding of the semantics in the input.

6 CONCLUSION

We explained our models for legal entailment and question answering in COLIEE 2023. For the case law retrieval task (Task 1), we used a sentence-transformer model to generate a multidimensional numeric representation of text, with some heuristic pre-processing and post-processing methods. For the statute law tasks, our transformer-based NLI system was ranked 3rd in Task 4. As future research, we will investigate the potential improvements to obtain deeper semantic representation from paragraphs and perform natural language inference between paragraphs. We also need to further investigate how to improve the discrimination power of the learned representations.

ACKNOWLEDGMENTS

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), [including funding reference numbers DGECR-2022-00369 and RGPIN-2022-0346], and Alberta Innovates.

REFERENCES

[1] Amin Abolghasemi, Sophia Althammer, Allan Hanbury, and Suzan Verberne. 2022. DoSSIER@COLIEE2022: Dense retrieval and neural re-ranking for legal case retrieval. In *Sixteenth International Workshop on Juris-informatics (JURISIN).*

[2] Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. 2015. A large annotated corpus for learning natural language inference. *arXiv preprint arXiv:1508.05326* (2015).

[3] Minh Quan Bui, Chau Nguyen, Dinh-Truong Do, Nguyen-Khang Le, Dieu-Hien Nguyen, and Thi-Thu-Trang Nguyen. 2022. Using Deep Learning Approaches for Tackling Legal's Challenges (COLIEE 2022). In *Sixteenth International Workshop on Juris-informatics (JURISIN).*

[4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *CoRR* abs/1810.04805 (2018). arXiv:1810.04805 http://arxiv.org/abs/1810.04805

[5] Tobias Fink, Gabor Recski, Wojciech Kusa, and Allan Hanbury. 2022. Statute-enhanced lexical retrieval of court cases for COLIEE 2022. In *Sixteenth International Workshop on Juris-informatics (JURISIN).*

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

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

[8] Matthew Honnibal and Mark Johnson. 2015. An Improved Non-monotonic Transition System for Dependency Parsing. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Lisbon, Portugal, 1373–1378. https://aclweb.org/anthology/D/D15/D15-1162

[9] N. Jiang and M.C. de Marneffe. 2019. Evaluating BERT for natural language inference: A case study on the CommitmentBank. 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)*. 6088––6093.

[10] 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).

[11] Shuyo Nakatani. 2010. Language Detection Library for Java. https://github.com/shuyo/language-detection

[12] 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.

[13] Tim Rocktäschel, Edward Grefenstette, Karl Moritz Hermann, Tomáš Kočiskỳ, and Phil Blunsom. 2015. Reasoning about entailment with neural attention. *arXiv preprint arXiv:1509.06664* (2015).

[14] Guilherme Moraes Rosa, Ruan Chaves Rodrigues, Roberto Lotufo, and Rodrigo Nogueira. 2021. Yes, BM25 is a Strong Baseline for Legal Case Retrieval. In *Proceedings of the 18th International conference on Artificial Intelligence and Law (ICAIL).*

[15] 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.*

[16] Adina Williams, Nikita Nangia, and Samuel R Bowman. 2017. A broad-coverage challenge corpus for sentence understanding through inference. *arXiv preprint arXiv:1704.05426* (2017).

THUIR@COLIEE 2023: More Parameters and Legal Knowledge for Legal Case Entailment

Haitao Li
DCST, Tsinghua University
Quan Cheng Laboratory
Beijing 100084, China
liht22@mails.tsinghua.edu.cn

Changyue Wang
DCST, Tsinghua University
Quan Cheng Laboratory
Beijing 100084, China
changyue20@mails.tsinghua.edu.cn

Weihang Su
DCST, Tsinghua University
Quan Cheng Laboratory
Beijing 100084, China
swh22@mails.tsinghua.edu.cn

Yueyue Wu
DCST, Tsinghua University
Quan Cheng Laboratory
Beijing 100084, China
wuyueyue@mail.tsinghua.edu.cn

Qingyao Ai
DCST, Tsinghua University
Quan Cheng Laboratory
Beijing 100084, China
aiqy@tsinghua.edu.cn

Yiqun Liu*
DCST, Tsinghua University
Quan Cheng Laboratory
Beijing 100084, China
yiqunliu@tsinghua.edu.cn

ABSTRACT

This paper describes the approach of the THUIR team at the COLIEE 2023 Legal Case Entailment task. This task requires the participant to identify a specific paragraph from a given supporting case that entails the decision for the query case. We try traditional lexical matching methods and pre-trained language models with different sizes. Furthermore, learning-to-rank methods are employed to further improve performance. However, learning-to-rank is not very robust on this task. which suggests that answer passages cannot simply be determined with information retrieval techniques. Experimental results show that more parameters and legal knowledge contribute to the legal case entailment task. Finally, we get the third place in COLIEE 2023. The implementation of our method can be found at https://github.com/CSHaitao/THUIR-COLIEE2023.

CCS CONCEPTS

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

KEYWORDS

legal case entailment, language model, legal NLP

ACM Reference Format:
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. In *Proceedings of COLIEE 2023 workshop, June 19, 2023, Braga, Portugal.* ACM, New York, NY, USA, 5 pages.

1 INTRODUCTION

In countries with Case Law system, like the United States, Canada, etc, past precedent is an essential reference for making judicial judgments [15, 26]. However, with the rapid growth of digital legal cases, legal practitioners need to expend significant effort to retrieve relevant documents and identify entailment parts. Recently, more and more researchers focus on intelligent legal systems to ease the heavy manual work [1, 2, 11, 17, 25, 29].

As a well-known competition in the legal field, the Competition on Legal Information Extraction/Entailment (COLIEE) aims to achieve state-of-the-art methods to help the realization of intelligent legal systems [20]. COLIEE contains two types of tasks: retrieval and entailment. The retrieval task is to identify the cases that support the query case from the large corpus [12, 16, 18]. The entailment task identifies a specific paragraph from a given supporting case that entails the decision for the query case [22, 23].

In this paper, we introduce the solution of the THUIR team for Legal Case Entailment task, which achieves third place and fifth place in the competition. To be specific, we formalize the entailment task as the paragraph ranking task. Then, we implemented several lexical matching models, such as BM25, QLD. Furthermore, contrastive learning loss is employed to fine-tune pre-trained models of different sizes. Finally, we utilize the above features to ensemble the final score. However, due to the sparse training data, the leaning to rank method does not achieve satisfactory performance, which may also indicate that the answer paragraphs cannot be simply confirmed by information retrieval techniques. As a result, THUIR teams placed third and fifth in 18 submissions from seven teams. Extensive experimental results show that more parameters and more legal knowledge contribute to better legal text understanding. The implementation of our method can be found at https://github.com/CSHaitao/THUIR-COLIEE2023.

This paper is organized as follows: Section 2 introduces the related work. In Section 3, details of the legal case entailment task are elaborated. Section 4 shows our detailed methodology. Then, the experimental setting and results are introduced in Section 5. Finally, we conclude our work and discuss the future direction in Section 6.

2 RELATED WORK

Legal Case Entailment is an essential task in the legal field which aims to determine whether some specific paragraphs entail the

*Corresponding author

COLIEE 2023, June 19, 2023, Braga, Portugal

Entailed fragment: It is well established that ministerial guidelines are permissible so long as they are not meant to bind administrative officers or to fetter their discretion……

Candidate 1: [1] This case is about a visa officer's refusal to issue student authorizations to young students wishing to attend primary and secondary schooling in Canada.
Candidate 2: [2] Citizenship and Immigration Canada (CIC) has published Processing Student Authorizations. Foreign students have the burden of proving to the satisfaction of officers that they are bona fide visitors……
Candidate 3: [3] The applicants, brothers who are now 14 and 12, are citizens of India and live in New Delhi with their parents.....

Target paragraph: [2] Citizenship and Immigration Canada (CIC) has published Processing Student Authorizations. Foreign students have the burden of proving to the satisfaction of officers that they are bona fide visitors……

Figure 1: An example of legal case entailment task.

decision of the query case. In past COLIEE competitions, various teams have achieved excellent results with traditional methods and deep learning methods, or a combination of both. For instance, the JNLP team [3] employs LEGAL-BERT and BM25 and try to capture keywords with "Abstract Meaning Representation". NM team [18] explores the zero-sample learning potential of language models using monoT5, which wins the championship in COLIEE 2022. Furthermore, the TR group [24] employs hand-crafted similarity features and trains with random forest classifiers. The UA team [10], on the other hand, utilizes methods such as fine-tuned language models and text summaries to accomplish the task. In this paper, we focus on the impact of more parameters and legal knowledge on legal case entailment task.

3 TASK OVERVIEW

3.1 Task Description

The Legal case entailment task refers to identifying specific paragraphs from existing cases that entail the decision for the query case. Formally, given a query case Q and a supporting case R consisting of paragraphs $P = P_1, P_2, ..., P_n$, this task is to determine the paragraph $P_i \in P$ that entails the decision for query case Q.

3.2 Data Corpus

The legal case entailment task is based on the existing collection of predominantly Federal Court of Canada case law. The training data includes the query case Q, a set of candidate paragraphs P and the corresponding labels. Test data includes only query case Q, and candidate paragraphs P. Figure 1 illustrates an example of legal case entailment task.

The COLIEE 2023 dataset contains 625 query cases for training and 100 query cases for testing. Table 1 shows the statistics of datasets for the last three years. The dataset has an average of 35 candidate paragraphs for each query case, of which only one is relevant on average. We randomly selected 100 query cases as the validation set and the rest for training. Details of the validation set can be found on GitHub https://github.com/lihaitao18375278/THUIR-COLIEE2023.

3.3 Metrics

The evaluation metrics of legal case entailment task are precision, recall and F1 score. Definition of these measures is as follows:

$$\text{Precision} = \frac{\#TP}{\#TP + \#FP} \tag{1}$$

$$\text{Recall} = \frac{\#TP}{\#TP + \#FN} \tag{2}$$

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

where $\#TP$ is the number of correctly retrieved candidate paragraphs for all query cases, $\#FP$ is the number of falsely retrieved candidate paragraphs for all query cases, and $\#FN$ is the number of missing noticed candidate paragraphs for all query cases.

4 METHOD

4.1 Traditional Lexical Matching Model

Traditional lexical matching models, i.e. BM25 and QLD, rank the candidate documents by a statistical probability model based on bag-of-words representation. In COLIEE 2023 task 2, we implement the following two lexical matching methods as baselines:

- **BM25** [21] is a classical lexical matching model with robust performance. The calculation formula of BM25 is shown in Eq 4.

$$BM25(d,q) = \sum_{i=1}^{M} \frac{IDF(t_i) \cdot TF(t_i,d) \cdot (k_1+1)}{TF(t_i,d) + k_1 \cdot \left(1 - b + b \cdot \frac{len(d)}{avgdl}\right)} \tag{4}$$

  where k_1, b are hyperparameters, TF represents term frequency and IDF represents inverse document frequency.
- **QLD** [30] is another representative traditional retrieval model based on Dirichlet smoothing. The equation for QLD is shown in Eq 5.

Table 1: Statistics of COLIEE Task 2.

	2021		2022		2023	
	Train	Test	Train	Test	Train	Test
# of query cases	425	100	525	100	625	100
Avg. # of candidates per query	35.80	35.24	35.69	32.78	35.22	37.65
Avg. # positive candidates per query	1.17	1.17	1.14	1.18	1.17	1.2
Avg. query length	37.51	32.97	36.64	32.21	35.36	36.57
Avg. candidate length	103.14	100.83	102.71	104.61	102.32	104.71

Table 2: Features that we used for learning to rank.

Feature ID	Feature Name	Description
1	query_length	Length of the query
2	candidate_length	Length of the candidate paragraph
3	BM25	Query-candidate scores with BM25
4	QLD	Query-candidate scores with QLD
5	BERT-large	Query-candidate scores with BERT-large
6	RoBERTa-large	Query-candidate scores with RoBERTa-large
7	LEGAL-BERT-base	Query-candidate scores with LEGAL-BERT-base
8	DeBERTa-v3-large	Query-candidate scores with DeBERTa-v3-large
9	monoT5-3B	Query-candidate scores with monoT5-3B

$$\log p(q|d) = \sum_{i:c(q_i;d)>0} \log \frac{p_s(q_i|d)}{\alpha_d p(q_i|C)} + n \log \alpha_d + \sum_i \log p(q_i|C) \tag{5}$$

For better performance, we remove all placeholders in the paragraph, e.g. "FRAGMANT_SUPPRESSED", "REFERENCE_SUPPRESSED" etc. Pyserini toolkit [1] is employed to implement BM25 and QLD with default parameters.

4.2 Pre-trained Language Model

4.2.1 Cross Encoder. As pre-trained language models have shown great potential in legal case entailment task [8, 27], we experimented with several pre-trained models of different sizes. More specifically, we employ the cross-encoder architecture. The scores of queries and candidates are defined as follows:

$$Input = [CLS]query[SEP]candidate[SEP] \tag{6}$$

$$score(query, candidate) = MLP(CLS[PLM(Input)]) \tag{7}$$

where CLS is the [CLS] token vector and MLP is a Multilayer Perceptron that projects the CLS vector to a score. PLM represents pre-trained language models. The purpose of training is to make the query case closer to related paragraphs in the vector space compared to the irrelevant ones. Thus, given a query case q, let d^+ and d^- be relevant and negative paragraphs, the loss function L is formulated as follows:

$$L(q, d^+, d_1^-, ..., d_n^-) = -\log \frac{exp(s(q, d^+))}{exp(s(q, d^+)) + \sum_{j=1}^{n} exp(s(q, d_j^-))} \tag{8}$$

We use the following model as the backbone for training:

- **BERT** [7] is a multi-layer bidirectional Transformer encoder architecture that utilizes the Masked Language Model(MLM) and Next Sentence Prediction(NSP) as pre-training tasks.
- **RoBERTa** [14] is an enhanced version of BERT with a more extensive dataset, which is pre-trained only with the Masked Language Model(MLM) task.
- **LEGAL-BERT** [4] is pre-trained with extensive English legal database and has achieved state-of-the-art performance on multiple legal tasks
- **DeBERTa** [9] proposes a disentangled attention mechanism and an enhanced mask decoder to improve the original BERT architecture, achieving state-of-the-art performance in COLIEE task2 in previous years.

4.2.2 Sequence-to-Sequence Model. Applying cross encoder to the legal case entailment task can be seen as a classification-based approach. Correspondingly, sequence-to-sequence models have been widely explored in this task [22, 23]. In general, sequence-to-sequence models perform better than cross encoders in data-poor settings due to capturing the underlying semantic relationships. In this section, we implement the following sequence-to-sequence models:

- **monoT5** [19] is an encoder-decoder architecture. It generates "true" or "false" token based on the relevance of queries and candidates, and regards the probability of generating "true" as the final relevance score. To be specific, the input to monoT5 has the following form:

$$Input = Query : [Q]Document : [P]Relevant : \tag{9}$$

[1] https://github.com/castorini/pyserini

Table 3: Overall experimental results of different methods on COLIEE2023 validation set.

Method	Params	P@1	R@1	F1 score
Lexical Matching Model				
BM25	-	0.670	0.563	0.612
QLD	-	0.632	0.529	0.575
Cross Encoder				
BERT-base	110M	0.750	0.630	0.685
BERT-large	340M	0.780	0.655	0.712
RoBERTa-base	110M	0.770	0.648	0.704
RoBERTa-large	340M	0.810	0.681	0.739
LEGAL-BERT-base	110M	0.830	0.697	0.758
DeBERTa-v3-base	110M	0.790	0.666	0.721
DeBERTa-v3-large	340M	0.830	0.697	0.758
Sequence-to-Sequence Model				
monoT5-base	250M	0.800	0.672	0.731
monoT5-3B	3000M	0.850	0.714	0.776
FLAN-T5-base	250M	0.700	0.588	0.639
FLAN-T5-3B	3000M	0.730	0.613	0.666
Ensemble	-	0.930	0.782	0.849

where [Q] and [P] are replaced with the query case and candidate paragraph texts, respectively. During the fine-tune process, "true" and "false" are the generated target token. At inference time, we calculate the probability of generating "true" token to determine the final paragraphs ranking.

- **FLAN-T5** [6] significantly improves zero-shot and few-shot abilities with instrution tuning and chain-of-thougt technology. FLAN-T5 is fine-tuned on 1836 tasks from 473 datasets with different instruction and improves zero-shot reasoning capability by a stepwise thinking approach.

4.3 Learning to Rank

Learning-to-rank, a popular machine learning method, is widely applied in various information retrieval competitions [5, 13, 28]. To further improve the performance, we employ learning-to-rank techniques in legal case entailment task, trying to mine the intrinsic qualities of different features with the gradient boosting framework.

More specifically, we extract all the feature scores listed in Table 2 and apply LightGBM to estimate the final scores of all query-candidate pairs. By fitting on the training set, we can learn the weights of different features. Finally, we choose the model that achieves the best NDCG@1 on the validation set for testing.

5 EXPERIMENT RESULT

The performance comparisons of different methods on the validation set are shown in Table 3. Since the average number of relevant paragraphs is about one for each query, we evaluate the precision, recall, and f1-score at the cut-off value of 1. From the experimental results, we have the following observations:

- As simple lexical matching cannot accurately capture the implication relationship between paragraphs, traditional methods such as BM25 and QLD have a more average performance.
- Benefit from supervised data, pre-trained language model further improves performance. The model with more parameters usually has a better performance. The monoT5-3B achieves the best results among all single models. This indicates that more parameters facilitate the understanding of the legal case entailment.
- Surprisingly, LEGAL-BERT-base with 110m parameters outperforms BERT-large and RoBERTa-large, which indicates that legal-oriented pre-training tasks allow the language model to have more legal knowledge and thus achieve better performance.
- Unexpectedly, the performance of FLAN-T5 drops dramatically. We assume that this is due to the inconsistency between the instruction fine-tuning process and the downstream tasks. In the future, we will explore more prompt formats to exploit the potential of large language models for legal case entailment task.
- Learning to rank techniques significantly improves the performance on the validation set. However, in the final leaderboard, the performance decreases after learning to rank on the contrary. We think this is due to overfitting due to a few training data. Also, it may indicate that the answer paragraphs cannot be simply confirmed by information retrieval techniques.

Table 4: Final top-5 of COLIEE 2023 Task 2 on the test set.

Team	Submission	Precision	Recall	F1
CAPTAIN	mt5l-ed	0.7870	0.7083	0.7456
CAPTAIN	mt5l-ed4	0.7864	0.6750	0.7265
THUIR	thuir-monot5	0.7900	0.6583	0.7182
CAPTAIN	mt5l-e2	0.7596	0.6583	0.7054
THUIR	thuir-ensemble_2	0.7315	0.6583	0.6930
THUIR	new ensemble	0.8000	0.6667	0.7273

Overall, more parameters and more legal knowledge can help language models perform better on legal case entailment tasks. In the future, we will explore the application of large legal language models to legal case entailment task.

The final top5 results of COLIEE2023 task 2 are shown in Table 4. The run with monoT5 has the third placement and the run with ensemble placed fifth. When we get the test set labels, we retrain the learning-to-rank model and choose a smaller early stop step. The results on the test set are reported in the last row of Table 4. We can find that learning to rank techniques can slightly improve performance by avoiding overfitting.

6 CONCLUSION

This paper shows our solution for the COLIEE2023 legal entailment task. We experiments with lexical matching model, cross encoder and sequence to sequence model. Learning to sort techniques is employed to get the final score. We finally achieve third place in this competition. Results show that more parameters and legal knowledge contribute to legal case entailment. In the future, we will design larger models with legal-oriented pre-training tasks.

REFERENCES

[1] Sophia Althammer, Arian Askari, Suzan Verberne, and Allan Hanbury. 2021. DoSSIER@ COLIEE 2021: leveraging dense retrieval and summarization-based re-ranking for case law retrieval. *arXiv preprint arXiv:2108.03937* (2021).

[2] Trevor Bench-Capon, Michał Araszkiewicz, Kevin Ashley, Katie Atkinson, Floris Bex, Filipe Borges, Daniele Bourcier, Paul Bourgine, Jack G Conrad, Enrico Francesconi, et al. 2012. A history of AI and Law in 50 papers: 25 years of the international conference on AI and Law. *Artificial Intelligence and Law* 20, 3 (2012), 215–319.

[3] 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)*.

[4] 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).

[5] Jia Chen, Haitao Li, Weihang Su, Qingyao Ai, and Yiqun Liu. 2023. THUIR at WSDM Cup 2023 Task 1: Unbiased Learning to Rank. arXiv:2304.12650 [cs.IR]

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

[7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

[8] Qian Dong, Yiding Liu, Suqi Cheng, Shuaiqiang Wang, Zhicong Cheng, Shuzi Niu, and Dawei Yin. 2022. Incorporating Explicit Knowledge in Pre-trained Language Models for Passage Re-ranking. *arXiv preprint arXiv:2204.11673* (2022).

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

[10] 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*.

[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]

[12] Haitao Li, Qingyao Ai, Jingtao Zhan, Jiaxin Mao, Yiqun Liu, Zheng Liu, and Zhao Cao. 2023. Constructing Tree-based Index for Efficient and Effective Dense Retrieval. arXiv:2304.11943 [cs.IR]

[13] Haitao Li, Jia Chen, Weihang Su, Qingyao Ai, and Yiqun Liu. 2023. Towards Better Web Search Performance: Pre-training, Fine-tuning and Learning to Rank. *arXiv preprint arXiv:2303.04710* (2023).

[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 preprint arXiv:1907.11692* (2019).

[15] Daniel Locke and Guido Zuccon. 2022. Case law retrieval: problems, methods, challenges and evaluations in the last 20 years. *arXiv preprint arXiv:2202.07209* (2022).

[16] Yixiao Ma, Yunqiu Shao, Bulou Liu, Yiqun Liu, Min Zhang, and Shaoping Ma. 2021. Retrieving legal cases from a large-scale candidate corpus. *Proceedings of the Eighth International Competition on Legal Information Extraction/Entailment, COLIEE2021* (2021).

[17] 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*. 2342–2348.

[18] Shubham Kumar Nigam, Navansh Goel, and Arnab Bhattacharya. 2023. nigam@ COLIEE-22: Legal Case Retrieval and Entailment using Cascading of Lexical and Semantic-based models. 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, 96–108.

[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] 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.

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

[22] 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).

[23] 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.

[24] 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*.

[25] 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*. 3501–3507.

[26] Yunqiu Shao, Yueyue Wu, Yiqun Liu, Jiaxin Mao, and Shaoping Ma. 2023. Understanding Relevance Judgments in Legal Case Retrieval. *ACM Transactions on Information Systems* 41, 3 (2023), 1–32.

[27] Xiaohui Xie, Qian Dong, Bingning Wang, Feiyang Lv, Ting Yao, Weinan Gan, Zhijing Wu, Xiangsheng Li, Haitao Li, Yiqun Liu, et al. 2023. T2Ranking: A large-scale Chinese Benchmark for Passage Ranking. *arXiv preprint arXiv:2304.03679* (2023).

[28] Shenghao Yang, Haitao Li, Zhumin Chu, Jingtao Zhan, Yiqun Liu, Min Zhang, and Shaoping Ma. 2022. THUIR at the NTCIR-16 WWW-4 Task. *Proceedings of NTCIR-16. to appear* (2022).

[29] Weijie Yu, Zhongxiang Sun, Jun Xu, Zhenhua Dong, Xu Chen, Hongteng Xu, and Ji-Rong Wen. 2022. Explainable legal case matching via inverse optimal transport-based rationale extraction. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 657–668.

[30] ChengXiang Zhai. 2008. Statistical language models for information retrieval. *Synthesis lectures on human language technologies* 1, 1 (2008), 1–141.

Performance of Individual Models vs. Agreement-Based Ensembles for Case Entailment

Michel Custeau
University of Ottawa
Ottawa, Canada
mcust094@uottawa.ca

Diana Inkpen
University of Ottawa
Ottawa, Canada
diana.inkpen@uottawa.ca

Abstract

This paper investigates the performance of an agreement-based ensemble approach in Task 2 of the COLIEE 2023 competition, which aims to assess entailment relationships between queries and candidate paragraphs in legal language. The experiments utilized agreement-based RoBERTa ensembles that combined differently pretrained RoBERTa models, which had the goal of improving overall performance by evaluating their agreement on entailment decisions. This method was designed based on its success on the previous year's competition dataset. The findings show that while the RoBERTa agreement-based ensembles achieved a higher score compared to individual models on the COLIEE 2022 dataset, it failed to outperform the individual models on the 2023 competition data, indicating that the ensemble approach in its current state may not be the most effective for this task. These results emphasize the necessity for further investigation of the suggested ensemble methodology, while concurrently identifying specific components within the ensembles that warrant enhancement.

Keywords: Legal NLP, Case Entailment, RoBERTa, SNLI, Agreement-based Ensemble

ACM Reference Format:
Michel Custeau and Diana Inkpen. 2023. Performance of Individual Models vs. Agreement-Based Ensembles for Case Entailment. In *Proceedings of COLIEE 2023 workshop, June 19, 2023, Braga, Portugal*. ACM, New York, NY, USA, 5 pages.

1 Introduction

Natural language processing (NLP) has made remarkable strides in recent years, with applications extending to complex domains such as legal text analysis. The ability to measure entailment relationships between queries and candidate paragraphs in legal documents can significantly enhance the automation of legal reasoning and decision-making processes. The COLIEE [4] competition serves as an ideal platform for assessing and advancing NLP techniques in the context of legal information extraction and entailment.

In this paper, we concentrate on Task 2 of the COLIEE 2023 competition, which was about determining the entailment relationships between paragraphs from court cases. We examine the potential of an agreement-based ensemble approach that incorporates differently pretrained RoBERTa [5] models by assessing their agreement on entailment decisions in order to improve overall performance. The first RoBERTa model was pretrained on a large corpus of Canadian court cases, the second model was pre-finetuned on a corpus of annotated entailment text pairs, and the third model combined the pretraining and pre-finetuning data from the previous two models into one model. Since all models had a different focus in their training data, the goal of the ensemble approach was to leverage both the strengths of the different models by prioritizing candidate cases that all models would agree upon. This strategy showed itself to be extremely effective on data from the previous year's competition. It also showed itself to be convenient from the point of view of the resources used, given that the base RoBERTa model has only 123 million parameters. However, this year proved that this strategy does not necessarily guarantee its effectiveness on the current data, landing it in 9th place in the competition and with a lower score than the individual models from the ensemble trained and tested on the same data.

2 Task Overview

Task 2 at COLIEE 2023 competition consists of finding textual entailment, also known as natural language inference, between a query and candidate paragraphs predominantly extracted from court case documents from the Federal Court of Canada. Given a query Q, we are given n number of candidate paragraphs $P = \{P_1, P_2, ..., P_n\}$ where we must classify the ones that entail or contradict the query Q.

2.1 Data Corpus

For the COLIEE 2023 task 2, the training set consists of 625 queries and the test set consists of 100 queries. As for the validation set, 20% of the training queries were used in our work.

COLIEE 2023, June 19, 2023, Braga, Portugal

Type	Train	Test
Number of Queries	625	100
Total Number of Paragraphs	22,018	3,765
Average Number of Entailment Paragraphs per Query	1.174	1.2
Average Number of Contradicting Paragraphs per Query	34.054	36.45
Average Number of Candidate Paragraphs per Query	35.229	37.65

Table 1. Analysis of the COLIEE 2023 data

Analysis of the dataset shown in Table 1 reveals that, on average, there are 1.174 entailment paragraphs per query in the training set and 1.2 in the test set. For the number of contradicting paragraphs per query, there is an average of 34.054 in the training set and 36.45 in the test set. Furthermore, the number of candidate paragraphs per query, which encompasses both entailment and contradicting paragraphs, amounts to an average of 35.229 in the training set and 37.65 in the test set. This shows that there are significantly more contradicting paragraphs than there are entailment paragraphs in the candidate set for each query.

There is also a much larger number of total paragraphs for training in comparison to last year's dataset, with 22,018 in total, amounting to 3,278 more training examples.

Type	Train	Test
Number of Queries	525	100
Total Number of Paragraphs	18,740	3,278
Average Number of Entailment Paragraphs per Query	1.116	1.18
Average Number of Contradicting Paragraphs per Query	34.579	31.6
Average Number of Candidate Paragraphs per Query	35.695	32.78

Table 2. Analysis of the COLIEE 2022 data

As indicated in Table 2, the averages derived from the 2022 dataset are not significantly different from those of the 2023 dataset. However, we can observe in the 2023 dataset that there is a small increase in the number of paragraphs per query within the test set, and a small increase in the occurrence of contradictory paragraphs per query for the test set also. It is also worth noting that the data from the training and test set of the 2022 competition is present in training set of the 2023 dataset.

2.2 Evaluation Metrics

The evaluation metrics used were recall, precision and F1 score.

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives + False Negatives}} \tag{1}$$

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives + False Positives}} \tag{2}$$

$$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision + Recall}} \tag{3}$$

3 Methods

This section will discuss the baseline model BM25 [8] and will discuss the RoBERTa agreement-based ensembles alongside the individual models themselves.

3.1 BM25

BM25 is an information retrieval function that ranks documents based on their relevance to a given query. It is built upon the classic Term Frequency-Inverse Document Frequency (TF-IDF) [10] function and calculates the relevance score for a document by taking into account term frequency, document length, and the inverse document frequency of query terms.

While BM25 is primarily used for ranking documents based on their relevance to a query, it can also be adapted for text entailment tasks where the highest-ranking candidate documents for the query are classified as entailment. While it may not fully capture the nuances and complexities of natural language inference due to its bag-of-words nature and focus on term frequency statistics, it can still serve as a good baseline for the overall performance of neural network models for legal text retrieval [9].

For this task, all text was turned into unigram tokens, with the stopwords and punctuation marks removed. The tokens were also lemmatized using the NLTK [6] library and lowercased. During classification, the first top-ranked candidate paragraph for each query was classified as entailment and the rest of the candidate paragraphs were classified as a contradiction.

3.2 Court Case RoBERTa

In recent years, self-supervised learning techniques employed by language models based on the transformer architecture [11], such as BERT [2] and GPT [7], have demonstrated remarkable effectiveness in addressing various NLP tasks. Given the success of these models, the RoBERTa model was utilized for the task, as it has been shown to match or even surpass BERT's performance while maintaining a lightweight architecture suitable for training on most local GPUs.

Research has shown that a second phase of pretraining with domain-specific data can benefit the performance of language models [3], which is why the RoBERTa model was chosen to be further pretrained on Canadian court case text. The dataset used to further pretrained the model consisted

	COLIEE 2022			COLIEE 2023		
Model	Recall	Precision	F1 Score	Recall	Precision	F1 Score
BM25	0.483	0.570	0.523	0.467	0.560	0.509
Court Case RoBERTa	0.619	**0.730**	0.670	0.592	0.710	0.645
SNLI RoBERTa	0.568	0.670	0.615	0.608	**0.730**	**0.664**
General and Legal Entailment RoBERTa	0.551	0.650	0.596	0.608	**0.730**	**0.664**
Agreement-Based Ensemble v1	0.720	0.669	**0.694**	**0.633**	0.644	0.639
Agreement-Based Ensemble v2	**0.729**	0.623	0.672	**0.633**	0.613	0.623

Table 3. Recall, Precision, and F1 Scores for Different Models

of 34,459 Canadian court cases. This corpus was provided by the Department of Justice Canada and the individual court case documents can be found open source on the CanLII website. The pretraining process spanned 20 epochs and employed dynamic masking to optimize the learning of relevant patterns in the legal domain. It was then subsequently fine-tuned on the task 2 entailment data and included in the ensemble.

In order to fine-tune the pretrained model to the dataset of Task 2, each query was concatenated with its corresponding candidate paragraphs. For every query-paragraph pair, the [SEP] token was employed as a separator for the model to distinguish the query from the paragraph. Subsequently, these concatenated query-paragraph pairs were passed into the model, which underwent training to classify the pairs as either entailment or contradiction.

3.3 SNLI RoBERTa

The Stanford Natural Language Inference (SNLI) [1] dataset is a widely used and publicly available resource consisting of pairs of text that are annotated for entailment, contradiction, or neutrality. The dataset comprises a substantial number of examples, with 550,000 training, 10,000 validation, and 10,000 test examples. It is worth noting that the neutral class is absent from the COLIEE task 2 annotations, which resulted in the exclusion of those examples from the SNLI dataset for this task specifically. Although the SNLI dataset is not tailored to a specific legal domain, its size and diversity make it a valuable resource for training models to perform textual inference, especially when taking into account that the task 2 legal entailment training corpus is of much smaller size than the SNLI dataset. To this end, the RoBERTa model was pre-finetuned on the SNLI dataset and further fine-tuned on the task 2 data, with the goal of leveraging the strengths of both datasets to enhance the model's performance.

3.4 General and Legal Entailment Roberta

Building on the idea of the models presented previously, we decided to test whether a model could benefit from exposure to all three datasets. Initially, the model was further pretrained on the corpus of Canadian court cases, then pre-finetuned on the SNLI dataset and finally, further finetuned on the Task 2 entailment dataset. This approach was predicated on the hypothesis that combining these diverse sources of knowledge could potentially lead to a more versatile and effective model.

3.5 Agreement-Based Ensemble

In order to create an ensemble approach, the outputs of the different RoBERTa models were combined. For each query, the models generated individual predictions concerning which candidate paragraphs could be classified as entailment. An agreement-based criterion was established for the ensemble's final prediction: if the models concurred on a set of the predictions of which candidates entailed the query, those candidates were selected as the final prediction.

In cases where the models did not reach an agreement on any candidate paragraph, a confidence-based criterion was resorted to. In that case, the paragraph exhibiting the highest confidence level for entailment, as determined by taking the softmax of the model's output, was then passed as the final prediction. This ensemble strategy aimed to capitalize on the strengths of the different RoBERTa models while mitigating the potential weaknesses of each individual model.

Two kinds of ensembles were created from the models. The first version of the ensemble used the outputs from the SNLI RoBERTa and the Court Case RoBERTa, while the second version also included the outputs from General and Legal Entailment RoBERTa alongside the two other models.

4 Results

The following presents the results of the ensembles and individual models on the dataset from this year's competition and the 2022 dataset.

Based on the results from Table 3, when looking at the results from the COLIEE 2022 data, the first version of the ensemble gave extremely good results with an F1 score of 0.69, which is even higher than the winning team of the competition that year who used a 3 Billion parameter model. These results are extremely favourable when also considering that we are using the base model of RoBERTa, which in relation is much less computationally expensive with only 123 million parameters. The models were also evaluated individually by extracting for each query the top paragraph with the highest

	COLIEE 2022	COLIEE 2023
Number of Agreements	67	62
Recall for Agreement Set	0.80	0.797
Precision for Agreement Set	0.680	0.688
F1-score for Agreement Set	0.736	0.738
Recall for Disagreement Set	0.553	0.412
Precision for Disagreement Set	0.636	0.553
F1-score for Disagreement Set	0.591	0.471

Table 4. Analysis of the Agreement and Disagreement Set of the Ensemble

confidence for entailment as the final prediction. We can see that for the 2022 data, both ensembles performed better than the individual models, with the Court Case RoBERTa performing better individually than the SNLI RoBERTa and the General and Legal Entailment RoBERTa. We can also see that all models were able to perform better than the BM25 baseline.

But contrary to the findings from the data of the previous year, this year's competition results indicate that the ensemble approach does not consistently outperform the individual models. A comparison of F1 scores reveals that the individual models exhibit superior performance when assessed independently rather than as part of the ensembles.

Interestingly, the SNLI RoBERTa and the General and Legal Entailment RoBERTa model, which were the least effective of the individual RoBERTa models on the 2022 dataset, emerged as the better performing model on this year's data, both achieving an F1 score of 0.664 which surpasses both the Court Case RoBERTa and the ensembles. In contrast, the performance of the ensembles were inferior to that of the individual models, as the first version only achieved an F1 score of 0.639 and the second version a lower f1 score of 0.623. Despite this, it is important to note that both the language models and the ensembles still managed to outperform the BM25 baseline, demonstrating an overall level of effectiveness.

When looking at the performance of the General and Legal Entailment RoBERTa model, it doesn't present significant benefits over the SNLI RoBERTa model. At best, Its performance matched that of SNLI RoBERTa on the 2023, while on the 2022 data, it got the lowest f1 score of all RoBERTa models. When considering its impact on the ensemble, it resulted in a decrease in the F1 score for both years relative to the ensemble excluding it. In terms of recall, a slight increase of 0.009 was noted on the 2022 data compared to the initial version of the ensemble, while it equaled the recall on the 2023 data. This observation suggests that pre-finetuning on the SNLI dataset and additional finetuning on the Task 2 dataset potentially negates most of the gains obtained during the pre-training on Court Case dataset step, thereby questioning the necessity of this step when the model is also finetuned on the SNLI and Task 2 dataset.

It is worth noting that when looking at the results of the ensembles for both years, one consistent outcome is that in both cases the recall of the ensembles increased above all of the individually evaluated models, and the precision of the ensembles dropped under those of the individual models comprising the ensemble. However, on the 2022 data, the gain in recall from using the ensembles was greater than 0.1 in comparison to the individual models, hence while the ensembles were penalised in terms of precision, this significant leap in recall were able to make up for it when calculating the F1 score. In contrast, when looking at the results on the 2023 test data, while the recall improved from using the ensembles, this gain was much smaller, with only a 0.041 gain from the Court Case RoBERTa model, and only a 0.025 gain from the SNLI RoBERTa model and General and Legal Entailment RoBERTa model. As a result, the gain in recall was not high enough to make up for the drop in precision that each ensemble brought. Overall, these metrics signify that, while the ensembles were adept at correctly identifying a large proportion of text pairs as entailment, they also inaccurately marked a substantial number of contradictory text pairs as entailment.

Several factors could potentially explain the difference in performance between the two datasets. One possible factor could be a variation in the distribution for the test set this year, which as we saw in the data corpus analysis, is not outside the realm of possibilities, as there is in fact an small increase in overall paragraphs and contradicting candidate paragraphs for each query of the test set in comparison to the data of last year. Another factor is that, when looking at Agreement-Based Ensemble v1, it had fewer agreements on this year's dataset than on the previous year's dataset, with 67 agreements on the 2022 data and 62 agreements on the 2023 data. This point is further supported by running the F1 score evaluation on only the queries where the models agreed on at least one candidate document. As we can see in Table 4, there is no significant difference between both years, hence the drop in performance on the 2023 data is most likely stemming from its predictions on queries where no agreement was reached. We can confirm this by evaluating the performance on only the queries where the models had no agreements, which as we can see from Table 4, is where

a significant drop in performance is seen on the 2023 data. Thus, the key to enhancing the performance of the ensembles lies in finding an alternative to the current confidence-based criterion when no agreement can be reached.

5 Conclusion

In conclusion, the findings of the experiments indicate that the agreement-based ensemble method holds potential for harnessing the strengths of multiple models, but it is not a foolproof approach that consistently outperforms individual models. The current limitations of this method underscore the necessity for further exploration and refinement of the ensemble method's criteria to optimize the benefits derived from each individual model, while minimizing any detrimental effects on overall performance. This calls for further research that seeks to enhance the ensemble approach and uncover innovative strategies to ensure its robustness and reliability in diverse settings.

References

[1] Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. 2015. A large annotated corpus for learning natural language inference. *arXiv preprint arXiv:1508.05326* (2015).

[2] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

[3] Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. *arXiv preprint arXiv:2004.10964* (2020).

[4] 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*. Springer, 51–67.

[5] 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 preprint arXiv:1907.11692* (2019).

[6] Edward Loper and Steven Bird. 2002. Nltk: The natural language toolkit. *arXiv preprint cs/0205028* (2002).

[7] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. 2018. Improving language understanding by generative pre-training. (2018).

[8] Stephen E Robertson, Steve Walker, Susan Jones, Micheline M Hancock-Beaulieu, Mike Gatford, et al. 1995. Okapi at TREC-3. *Nist Special Publication Sp* 109 (1995), 109.

[9] Guilherme Moraes Rosa, Ruan Chaves Rodrigues, Roberto Lotufo, and Rodrigo Nogueira. 2021. Yes, bm25 is a strong baseline for legal case retrieval. *arXiv preprint arXiv:2105.05686* (2021).

[10] Gerard Salton, Anita Wong, and Chung-Shu Yang. 1975. A vector space model for automatic indexing. *Commun. ACM* 18, 11 (1975), 613–620.

[11] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30 (2017).

Japanese Legal Bar Problem Solver Focusing on Person Names

Takaaki Onaga
Shizuoka University
Hamamatsu, Shizuoka, Japan
tonaga@kanolab.net

Masaki Fujita
Shizuoka University
Hamamatsu, Shizuoka, Japan
mfujita@kanolab.net

Yoshinobu Kano
Shizuoka Univerity
Hamamatsu, Shizuoka, Japan
kano@kanolab.net

ABSTRACT

This paper describes our system for COLIEE 2023 Task 4, which automatically answers Japanese legal bar exam problems. We propose an extension to our previous system in COLIEE 2022, which achieved the highest accuracy among all submissions by using data augmentation. In this paper, we present three main contributions. First, we incorporate LUKE as our deep learning component, a named entity recognition model trained on RoBERTa. Second, we ensemble the given training datasets in a manner similar to cross-validation, to utilize the training data to the fullest extent possible. Third, we fine-tune the pretrained LUKE model in multiple ways, comparing fine-tuning on training datasets that include alphabetical person names and ensembling different fine-tuning models. Our formal run results show that LUKE and our fine-tuning approach using alphabetical person names were effective, achieving an accuracy of 0.69 in the COLIEE 2023 Task 4 formal run.

CCS CONCEPTS

• **Information systems** → **Content analysis and feature selection**; **Similarity measures**; **Clustering and classification**; *Document topic models*; *Information extraction*; Specialized information retrieval.

KEYWORDS

COLIEE, legal textual entailment, Legal Ber Exam, LUKE, Predicate Argument Structure Analysis

ACM Reference Format:

Takaaki Onaga, Masaki Fujita, and Yoshinobu Kano. 2023. Japanese Legal Bar Problem Solver Focusing on Person Names. In *Proceedings of COLIEE 2023 workshop, June 19, 2023,Braga, Portugal*. ACM, New York, NY, USA, 9 pages.

1 INTRODUCTION

COLIEE (Competition for Legal Information Extraction) is an annual international competition held in conjunction with the ICAIL (International Conference on Artificial Intelligence and Law) and JURISIN (Juris-informatics) conferences [1] [8] [7] [6] [11] [14] [12] [13] [5]. COLIEE 2023 consists of four tasks: Tasks 1 and 2 are case law tasks that use datasets from the Canadian Federal Court, while Tasks 3 and 4 are statute law tasks that use the Japanese Legal Bar exam. In Task 3, a participant system is given a problem text and asked to retrieve relevant articles from Japanese Civil Law to solve the problem. In Task 4, a participant system is given a problem text and its relevant articles, and asked to determine whether the articles entail the problem text or not by answering *Yes* or *No*. We participated in Task 4.

The analysis of problem types in previous COLIEE tasks [12] showed that the COLIEE dataset includes diverse types of problems. Some are relatively easy to solve because the texts in the pairs are very similar, while others are complex and difficult, requiring parsing, semantics, anaphora, logic, etc. Previous Task 4 participant systems have included rule-based and deep learning-based systems such as BERT [19], ELECTRA [10], and GNN [17]. However, previous systems have not performed well on problems that require inferences about person roles.

In this paper, we focus on person name resolution, where person names/roles are represented using alphabetical letters. We propose a system that extends our previous system in COLIEE 2022, which achieved the highest accuracy among all submissions by using data augmentations. Our proposed system provides three main contributions. First, while we use an ensemble of a rule-based component and a deep learning-based component, we adopt LUKE as our deep learning-based component, which is a named entity trained model based on RoBERTa, instead of BERT. Second, we perform an ensemble of the given training datasets using a cross-validation method to make the most of the training dataset. Third, we fine-tune the pretrained LUKE model in multiple ways, comparing fine-tuned training datasets that include alphabetical person names and an ensemble of different fine-tuned models. Our formal run results show that LUKE and our fine-tuning approach for alphabetical person names are effective.

2 RELATED WORKS

LUKE [18] is a language model based on RoBERTa [9], which is a derivative of BERT [2]. BERT is a deep learning model that is commonly used in various NLP tasks, and it utilizes the encoder part of the Transformer [16] architecture. LUKE, on the other hand, uses a unique mechanism called Entity-aware Self-attention. At the time of its development, LUKE achieved the highest accuracy in several NLP tasks. In this paper, we fine-tune the pretrained LUKE model.

Hoshino et al. [4] is our previous work presented in COLIEE 2019. They proposed a rule-based system that parses sentences into clauses based on their original definition. The parsing results were then used to extract the set of clauses, including subject, predicate, and object for each clause, and compared these sets. They developed several modules, such as the Precise Match module, which compared the relevant civil law clauses with the clause set of the problem text and answered *Yes* if all the elements in the clause sets matched.

COLIEE 2023, June 19, 2023, Braga, Portugal

Fujita et al. [3] is another recent work of ours in COLIEE 2022, which proposed an ensemble of the rule-based system developed by Hoshino et al.'s rule-based system and a BERT-based system. This system achieved the highest accuracy in the formal run of COLIEE 2022 Task 4. In order to address the issue of limited training data, we performed data augmentation such as logical inversion, replacement of person terms, and replacement of article numbers. In this paper, we extended our previous system by replacing BERT with LUKE and modifying the ensemble method to build different fine-tuned models depending on the type of problem.

3 SYSTEM

3.1 System Overview

Our system comprises a rule-based component and a LUKE-based component. The LUKE-based component utilizes a LUKE model, which is fine-tuned on three different datasets: all training datasets provided by COLIEE, and two types of training datasets extracted from different problem types. The rule-based and LUKE-based components are integrated through ensemble, which performs binary classification, predicting either *Yes* or *No* based on the higher probability value.

In the COLIEE Task 4 dataset, alphabetical characters are used to represent persons in the problem text, as illustrated in Figure 1, which shows an example of a problem involving alphabetical person characters. It is necessary to determine the relationship between each person indicated by an alphabetical character and the person role described in the civil law text. In the example, A in the problem text represents a person who contracted as an agent of another person, B represents a different person, and C corresponds to a counterparty, as defined in the civil code text. Such problems are considered to be among the most challenging to solve automatically.

We focus on problems that involve alphabetical person names, and create separate LUKE models trained on such problems and trained on other problems. For the LUKE-based part, we prepare three LUKE models for comparison: a LUKE model trained on all data (**LUKE-all**), a LUKE model trained on problems with alphabetical person names (**LUKE-person**), and a LUKE model trained on problems without alphabetical person names (**LUKE-nonperson**). While our previous system [4] had different modules with different matching methods for the clause sets, our previous study [3] showed that the Precise Match module was the most effective, answering *Yes* only when all pairs of subjects, objects, and predicates match. Therefore, we adopt the Precise Match module as our rule-based part.

We fine-tuned a publicly available LUKE model (studio-ousia/luke-japanese-base-lite[1]) which was pre-trained on Wikipedia articles, to output binary probabilities of *Yes* or *No*, given a problem text and a relevant civil law article as input.

In this section, we describe the design of our system as follows. First, we create additional training data using civil law articles (3.2). Second, after preprocessing the data, we select the most relevant civil law article for solving a given problem statement, based on the similarity of their texts (3.3). Third, we expand the training data by performing logical inversion and replacing person terms (3.4). Fourth, we fine-tune the LUKE model using these datasets. We split the datasets by year and create multiple models for all possible combinations of the training and validation datasets (3.5). Based on the methods above, we created three different submission models for our formal run results: **KIS1**, **KIS2**, and **KIS3**, which were designed for different types of problems (3.6). Among the three formal run submissions, **KIS2** was our proposed system. **KIS1** was an ensemble of a LUKE-based model using all of the training data and the rule-based system. **KIS2** was an ensemble of **KIS1** and a model trained specifically for problems in which alphabetical person names appear. **KIS3** was an ensemble of a model trained specifically for problems in which alphabetical person names appear and a model trained specifically for problems in which they do not appear. Figure 2 illustrates these relationships. We applied our article selection preprocess (3.3) to the formal run test dataset.

<id= ” R02-4-I ” , label= ” Y">
<article>
A person who has contracted as an agent of another person shall be liable to the other party for performance or damages at the other party's option, unless he has proved his own agency or has obtained his own additional authorization.
<problem>
A, purporting to be B's agent, entered into a purchase agreement with C to sell land owned by B to C, but did not actually have the agency to enter into the agreement; if B ratified the purchase agreement, A is not liable to C as an unauthorized agent.

<関連条文>
他人の代理人として契約をした者は、自己の代理権を証明したとき、又は本人の追認を得たときを除き、相手方の選択に従い、相手方に対して履行又は損害賠償の責任を負う。
<問題文>
Aは， B の代理人と称して， C との間で B の所有する土地を C に売却する旨の売買契約を締結したが，実際にはその契約を締結する代理権を有していなかった。 B が売買契約を追認した場合， A は C に対する無権代理人の責任を負わない。

Figure 1: An example of a problem where alphabetical person characters appears

3.2 Create Training Data from Article(s)

In order to increase the size of the official training dataset, we created an additional training dataset using the civil code articles without problem texts. In this subsection, we will refer to the relevant articles in COLIEE as premise (t1) and the problem text in COLIEE as hypothesis (t2) to avoid confusion since both are taken from the articles.

First, we divided the distributed civil law articles into sections and created pairs of identical civil code sections, setting their correct answer labels to *Yes*. For example, "A minor must obtain the consent of his/her legal representative to perform a legal act. However, this shall not apply to acts merely to obtain rights or to be

[1]https://huggingface.co/studio-ousia/luke-japanese-base-lite

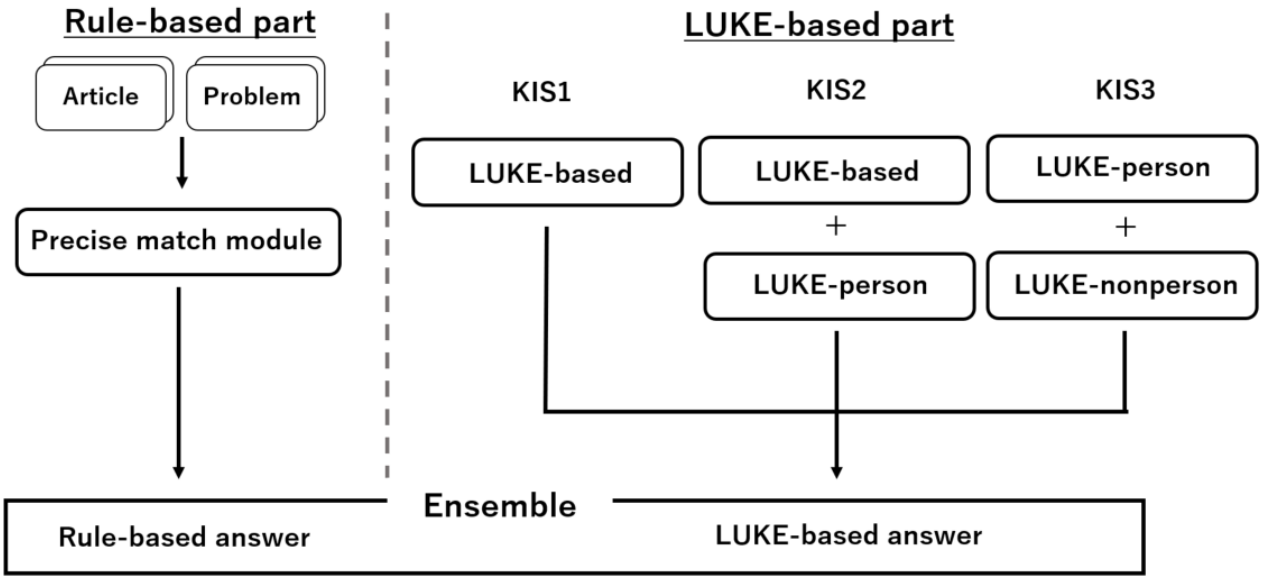


Figure 2: System Overview

relieved of obligations. (Civil Code Article 5)" and the same paragraph are paired with the label *Yes*. If the text of the article contains an exception sentence or proviso, such as "Provided, however, , this shall not apply.", we divided the original article texts into a text before the sentence (a principle part) and after the sentence (a proviso part). If "However, , this shall not apply." describes an act, person, or right, we manually replace that act, person, or right in the principle part with an act, person, or right in the proviso part. Then, we invert the logic of the predicate as described in 3.4. In the example in Figure 3, Article 5 of the Civil Code "However, this shall not apply to acts by which a minor merely acquires a right or is relieved of a duty." was rewritten as "A minor need not obtain the consent of his or her legal representative to commit an act merely to obtain a right or to be relieved of a duty." The subject normally appears in the principle part, but sometimes it appears in the proviso part. When the subject appears in the proviso part, we revert the affirmative/negation of the principle part using the method described later (3.4) and add it to the training dataset, sharing the same original premise (t1). Figure 4 shows an example.

3.3 Preprocess and Article Selection

First, we apply the following preprocessing steps to the articles and then select the relevant ones. A problem statement may have multiple related articles. If we concatenate the texts of all these articles as input, the input to the model may become too long, exceeding the upper limit (in our case, 512 tokens), and important parts may be lost when we truncate the input. To address this issue, we split the relevant articles into sections (each article consists of one or more sections). Then, we create all possible combinations of the divided sections (Figure 5). We discard any combination in which

<Article 5 of the Civil Code (Original)>
A minor must obtain the consent of the minor's legal representative to perform a juridical act. provided, however, that this does not apply to a juridical act for merely acquiring a right or being released from an obligation.
・Split by "however."
<principle part>
A minor must obtain in the consent of his/her legal representative to perform a legal act.
<exception part>
A minor need not obtain the consent of his or her legal representative for a juridical act for merely acquiring a right or being released from an obligation.

民法第五条
<民法第 5 条 (原文)>
未成年者が法律行為をするには、その法定代理人の同意を得なければならない。 ただし、単に権利を得、又は義務を免れる法律行為については、この限りでない。
・「ただし、」で分割する。
<原則部分>
未成年者が法律行為をするには、その法定代理人の同意を得なければならない。
<例外部分>
未成年者が、単に権利を得、又は義務を免れる法律行為をするには、その法定代理人の同意を得なくてもよい。

Figure 3: Divide into principle and exception

Pair 1: <t1> and <t2> are identical, the original text of Article 5 of the Civil Code.
<t1>
A minor must obtain the consent of the minor's legal representative to perform a juridical act. provided, however, that this does not apply to a juridical act for merely acquiring a right or being released from an obligation.
未成年者が法律行為をするには、その法定代理人の同意を得なければならない。ただし、単に権利を得、又は義務を免れる法律行為については、この限りでない。
<t2>
A minor must obtain the consent of the minor's legal representative to perform a juridical act. provided, however, that this does not apply to a juridical act for merely acquiring a right or being released from an obligation.
未成年者が法律行為をするには、その法定代理人の同意を得なければならない。ただし、単に権利を得、又は義務を免れる法律行為については、この限りでない。

Pair 2: Proviso part is not needed to solve the problem.
<t1>
Same as pair 1 ' s <t1>
<t2>
A minor must obtain the consent of his/her legal representative to perform a legal act.
未成年者が法律行為をするには、その法定代理人の同意を得なければならない。

Pair 3: Proviso part is needed to solve the problem.
<t1>
Same as pair 1 ' s <t1>
<t2>
A minor need not obtain the consent of his or her legal representative for a juridical act for merely acquiring a right or being released from an obligation.
未成年者が、 単に権利を得、又は義務を免れる法律行為をするには、その法定代理人の同意を得なくてもよい。

Figure 4: <t1><t2> pairs created using exceptions

the total number of tokens of the combined sections and the given problem text exceeds the upper limit.

If the generated text contains reference notations such as "preceding paragraph" or "Article XX", we search the given relevant articles for the referred article and replace the reference notations with the text from the referred article (as shown in Figure 6). The replaced version is then added to the training dataset. Notations such as "listed below" are substituted with the specified items in the article. Figure 7 provides an example of this process.

As shown in the figure 3, the proviso part of an article describes an exceptional situation where the principle part does not apply. To understand the meaning of the proviso part, we need to include the principle part as well. Therefore, we concatenate the proviso part with its principle part, inverting the affirmation/negation of the latter. If the proviso part includes an act, person, or right, we

<id= " R01-3-E", label= " Y " >
<article>
1, Article 106 A sub agent shall represent the principal with respect to acts within his/her authority.
2,(2) A sub agent shall have the same rights and assume the same obligations as an agent with respect to the principal and third parties within the scope o f his/her authority.
・Generate combinations for each divided term
1, A sub agent shall represent the principal with respect to acts within his/her authority.
2, A sub agent shall have the same rights and assume the same obligations as an agent with respect to the principal and third parties within the scope of his/her authority.
1 + 2, A sub agent shall represent the principal with respect to acts within his/her authority. A sub agent shall have the same rights and assume the same obligations as an agent with respect to the principal and third parties within the scope of his/her authority.

<関連条文>
1, 第百六条 復代理人は、その権限内の行為について、本人を代表する。
2, 2 復代理人は、本人及び第三者に対して、その権限の範囲内において、代理人と同一の権利を有し、義務を負う。
・項ごとの組合せを生成する
1, 復代理人は、その権限内の行為について、本人を代表する。
2, 復代理人は、本人及び第三者に対して、その権限の範囲内において、代理人と同一の権利を有し、義務を負う。
1+2, 復代理人は、その権限内の行為について、本人を代表する。復代理人は、本人及び第三者に対して、その権限の範囲内において、代理人と同一の権利を有し、義務を負う。

Figure 5: An example of combinations reconstruction

replace the corresponding item in the principle part with the one in the proviso part.

Among these preprocessed articles, we select most relevant article to solve the given problem by the similarity scores of the vectors obtained by Sentence Luke (sonoisa/sentence-luke-japanese-base-lite[2])). Sentence LUKE is a tool for creating advanced sentence vectors using the LUKE model (LUKE version of the Sentence BERT [15] in other words), which was pretrained by the Japanese Wikipedia and the Siamese network.We remove the suffixes of predicates, which could contain negation expressions. This is because we search for the most similar content regardless of affirmative/negative. Figure 8 shows an example.

3.4 Data Augmentation

Our previous COLIEE 2022 system [3] consisted of two expansions: negation expansion and person term replacement, which we describe below. In this year's formal run, we have added more negative words and person terms to our manual dictionary.

[2]https://huggingface.co/sonoisa/sentence-luke-japanese-base-lite

<id= ”R03-5-A”, label= ”N”>
<article>
Article 150
When a demand is made, the prescription shall not be completed until six months have elapsed from that time.
(2) Another demand made while the completion of prescription is postponed by a demand shall not have the effect of postponing the completion of prescription under the preceding paragraph.
・Substitute the first paragraph for **"preceding paragraph"** in the second paragraph.
Another demand made while the completion of prescription is postponed by a demand shall not have the effect of postponing the completion of prescription under the paragraph that state: "When a demand is made, the prescription shall not be completed until six months have elapsed from that time."

<関連条文>
第百五十条
催告があったときは、その時から六箇月を経過するまでの間は、時効は、完成しない。
２ 催告によって時効の完成が猶予されている間にされた再度の催告は、 前項 の規定による時効の完成猶予の効力を有しない。
・**"前項"**を置き換える。
催告によって時効の完成が猶予されている間にされた再度の催告は、「催告があったときは、その時から六箇月を経過するまでの間は、時効は、完成しない。」の規定による時効の完成猶予の効力を有しない。

Figure 6: An example of article reference

<pair id="H30 4 I", label="N">
<article>
Article 103 An agent without prescribed authority shall have the authority to perform only the following acts
(i) acts of preservation
(ii) acts for the purpose of utilizing or improving the object or right for which the agent is acting, to the extent that the nature of such object or right is not changed
・Substitute each item for "the following acts".
(i) An agent with no defined authority is authorized only to perform acts of preservation.
(ii) An agent without prescribed authority is authorized only to perform acts for the purpose of using or improving the thing or right that is the object of the representation, to the extent that the nature of the thing or right is not changed.

<関連条文>
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
・次に掲げる に各号を代入する。
(一) 権限の定めのない代理人は、 保存行為のみをする権限を有する。
(二) 権限の定めのない代理人は、 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 のみをする権限を有する。

Figure 7: An example of substituting each item for "lited below”

For negation expansion, we create a new sample by reversing the logic at the end of a sentence, along with its *Yes* or *No* answers, using a predefined list of affirmative and negation expression pairs. We apply this expansion to both pairs created from the Civil Code articles as described in the previous sections and the given problem text. However, we do not apply this expansion to problems with a gold standard answer of *No* since the negative form at the end of a sentence does not always result in a *Yes* when the original answer is *No*.

The COLIEE problems sometimes use alphabetical characters, such as A or B, to represent person names. Our person term replacement expansion addresses this issue by creating a dataset from the training data that replaces person names with alphabetical characters. We assign the alphabetical letters in the order of appearance, holding identical person names to be identical characters.

3.5 Combinatorial Split of Training and Validation Dataset

In order to fully utilize the COLIEE official training dataset, we created multiple models trained with different parts of the official dataset. We split the official dataset using various patterns, such as the cross-validation method, where we selected each two-year period as a validation dataset and used the rest of the official dataset as its training dataset. After fine-tuning for each pattern, we applied an ensemble of these multiple models. We chose two years as our splitting unit because it would be too many combinations if we split by year. Figure 9 illustrates this split method.

3.6 Fine Tune for Alphabetical Person Names

When alphabetical letters are used as person names in the given problem text, a different approach is required to solve the problem, as it becomes necessary to determine which person the alphabetical character corresponds to in the relevant civil law article. Therefore, we fine-tune a model specifically for such problems.

Additionally, we fine-tune a model for problems in which alphabetical person names do not appear. Each model internally performs an ensemble of the combinatorial split fine-tunes described in section 3.5, and thus, the preprocessing steps described in sections 3.1 to 3.4 are applied before the fine-tuning.

We regard a problem as an alphabetical person name type problem if it contains any single alphabetical character (as the original

<id= ” R02-5-U ”, label= ” Y ” >
<article>
Article 724
The right to claim damages for a tort shall be extinguished by prescription in the following cases
(i) When the victim or his/her legal representative has not exercised it for three years from the time when he/she became aware of the damage and the perpetrator
(ii) When the right is not exercised for 20 years from the time of the tortious act.
<problem>
The right to claim damages based on a tort shall be extinguished by prescription if not exercised for 20 years from the time of the tortious act.

・Find the similarity between the problem statement and the article paragraph by paragraph.
1, **The right to claim damages in tort shall be extinguished by prescription if not exercised for twenty years from the time of the tortious act. (The similarity of this statement was the highest.)**
2, The right to claim damages for a tort shall be extinguished by prescription if the victim or his/her legal representative does not exercise the right for three years from the time when he/she learned of the damage and the perpetrator.

<関連条文>
第七百二十四条
不法行為による損害賠償の請求権は、次に掲げる場合には、時効によって消滅する。
一 被害者又はその法定代理人が損害及び加害者を知った時から三年間行使しないとき。
二 不法行為の時から二十年間行使しないとき。
<問題>
不法行為に基づく損害賠償請求権は，不法行為の時から２０年間行使しない場合，時効によって消滅する。
・問題文と条文の組合せごとの類似度を求める。
1, 不法行為による損害賠償の請求権は、不法行為の時から二十年間行使しないときには、時効によって消滅する。(この文の類似度が最も高くなった。)
2, 不法行為による損害賠償の請求権は、被害者又はその法定代理人が損害及び加害者を知った時から三年間行使しないときには、時効によって消滅する。

Figure 8: An example of article selection

text is in Japanese except for these characters). As mentioned earlier, **KIS2** and **KIS3** use the model fine-tuned with problems containing alphabetical characters, while **KIS1** uses the model fine-tuned without them.

During binary classification, a fully connected linear transformation is performed on the output of the last layer's node corresponding to the "<s>" token (or the "[CLS]" token in the case of BERT) for both *Yes* and *No* answers. Then, the classification scores are compared to determine whether the answer is "*Yes* or *No*. For

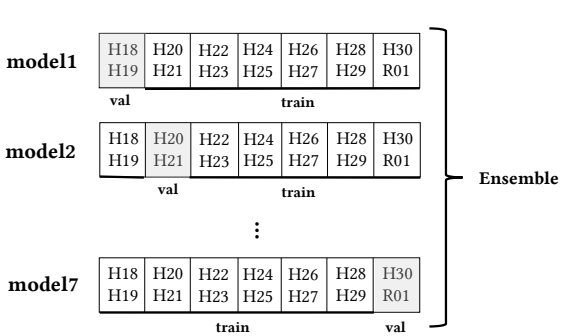


Figure 9: A conceptual figure of training data split

fine-tuning, the classification scores are converted into probabilities for each label using the Softmax function, and the loss is calculated using cross-entropy.

3.7 Ensemble Prediction

Finally, we perform an ensemble of our rule-based part and our LUKE-based part. The rule-based (precise match module) is the same as in our previous work, which has high precision but a low number of answerable problems. Therefore, we first apply the rule-based part when applicable, and then apply the LUKE-based part when the rule-based part is not applicable.

For the LUKE-based part, we have prepared three models: **LUKE-all** (fine-tuned on all of our datasets), **LUKE-person** (fine-tuned on problems with alphabetical person names), and **LUKE-nonperson** (fine-tuned on problems without alphabetical person names). **KIS3** applies **LUKE-person** when the problem includes alphabetical person names and applies **LUKE-nonperson** when the problem does not include any alphabetical person names. Similarly, **KIS2** applies **LUKE-person** in the same way but uses **LUKE-all** when the problem does not include any alphabetical person names. If the rule-based part is not applicable, KIS always applies **LUKE-all**.

4 EXPERIMENTS AND RESULTS

4.1 Fine-tune Parameters

We performed our fine-tuning with the following parameters: maximum string length of 512, batch size of 32, learning rate of 1e-5, and a maximum number of epochs of 10 but terminates early due to Early Stopping.

4.2 COLIEE 2023 Formal Run Results

Table 1 shows the results of all teams in the COLIEE 2023 Task 4 ' s formal run, where KIS is our team name.

4.3 Previous COLIEEs ' formal run results

Table 2 shows the results of our experiments using previous formal runs of COLIEE 2019, 2020, and 2021 (test datasets are H30, R01, and R02, respectively) as required by the organizers.

Table 1: COLIEE 2023 Task 4 ' s formal run results for each participant ' s submission

Submission	Correct	Accuracy
BaseLine	52 / 101	0.5149
AMHR02	82	0.8119
JNLP3	79	0.7822
TRLABS_D	79	0.7822
TRLABS_I	79	0.7822
JNLP1	76	0.7525
JNLP2	76	0.7525
TRLABS_T	76	0.7525
KIS2	**70**	**0.6931**
KIS1	**68**	**0.6733**
UA_V2	67	0.6634
AMHR01	66	0.6535
KIS3	**66**	**0.6535**
AMHR03	65	0.6436
LLNTUdulcsL	63	0.6238
UA	63	0.6238
HUKB2	60	0.5941
CAPTAIN.gen	59	0.5842
CAPTAIN.run1	58	0.5743
LLNTUdulcsS	57	0.5644
HUKB1	56	0.5545
HUKB3	56	0.5545
LLNTUdulcsO	56	0.5545
NOWJ.multi-v1-jp	55	0.5446
CAPTAIN.run2	53	0.5248
NOWJ.multijp	53	0.5248
NOWJ.multi-v1-en	49	0.4851

Table 2: Results of previous formal run datasets

Model name	H30	R01	R02
(Number of problems)	70	111	81
KIS1	44 (0.62)	75 (0.67)	56 (0.69)
KIS2	44 (0.62)	77 (0.69)	58 (0.71)
KIS3	43 (0.61)	73 (0.65)	52 (0.62)

4.4 Comparison of BERT and LUKE

Table 3table:Results of BERT and LUKE shows the results of the experiments on the formal run and the past formal runs using BERT and LUKE. Each cell shows numbers of correct answers with total numbers of problems from H30 to R04; the **all** column shows the total numbers, the **person** column shows the numbers for problems containing characters of the alphabetical person names, and the **nonperson** column shows the numbers for problems without the alphabetical person names. The results of this table shows that the correct numbers of the LUKE model is larger than the BERT model in H30 and R04. Especially in R04, LUKE improved the performance of the alphabetical person names problems. On the other hand, BERT had higher performance in R01 and similar performance in R02.

Table 3: Results of BERT and LUKE

Model	all	person	nonperson
H30 (BERT)	42/70	7/13	35/57
H30 (LUKE)	44/70	7/13	37/57
R01 (BERT)	73/111	20/34	53/77
R01 (LUKE)	72/111	20/34	52/77
R02 (BERT)	56/81	23/35	33/46
R02 (LUKE)	56/81	22/35	34/46
R04 (BERT)	61/101	27/41	34/60
R04 (LUKE)	66/101	29/41	37/60

Table 4: Results of problem types

	LUKE-all	LUKE-person	LUKE-nonperson
H30 nonperson	37 / 57	35 / 57	35 / 57
H30 person	7 / 13	8 / 13	5 / 13
R01 nonperson	52 / 77	51 / 77	51 / 77
R01 person	20 / 34	23 / 34	18 / 34
R02 nonperson	34 / 46	32 / 46	31 / 46
R02 person	22 / 35	25 / 35	22 / 35
R04 nonperson	37 / 60	36 / 60	34 / 60
R04 person	29 / 41	31 / 41	27 / 47

4.5 Evaluation of Fine Tune Models without Ensemble Using Previous Formal Runs

Table 4 shows the evaluation results of the individual fine-tuned models on the formal run of COLIEE 2023 and the formal runs of the past three years. Each fine-tuned model was evaluated independently without any ensemble. We evaluated the models separately for the problems with alphabetical person names (person) and others (nonperson).

5 DISCUSSION

The individual results of the fine tuned models (Table 4) demonstrate that the fine-tuning was effective for the corresponding type of problems but not for the other types.

Our team's formal run results (Table 1) and the results of our experiments using past formal runs (Table 2) also showed that **KIS2**, which is an ensemble using the fine tuned model for alphabetical person names, achieved the highest score.

Table 3 shows that LUKE and BERT have different percentages of correct answers. We analyzed the patterns in which either LUKE or BERT answered problems correctly. As shown in Figure 10, R04-08-A is an example of a person name problem where LUKE was correct and BERT was incorrect. In this problem, the gold label is "*No*" because "B consented to this" in the problem text is different

<pair id="R04-08-A" label="N">
<article>
Article 178, An assignment of a real right over movables may not be asserted against a third party without delivery of the movables.
Article 184, In cases of possession by an agent, if the principal has ordered the agent to take possession of the thing for a third party thereafter and the third party has consented thereto, the third party shall acquire the right of possession.
第百七十八条 動産に関する物権の譲渡は、その動産の引渡しがなければ、第三者に対抗することができない。
第百八十四条 代理人によって占有をする場合において、本人がその代理人に対して以後第三者のためにその物を占有することを命じ、その第三者がこれを承諾したときは、その第三者は、占有権を取得する。
<problem>
If A sells to C a painting A owned by A while leaving it with B, and A orders B to take possession of A for C thereafter, and B agrees to this, C may assert against the third party the acquisition of the ownership of A.
Ａがその所有する絵画甲をＢに預けたままＣに売却した場合において、ＡがＢに対して以後Ｃのために甲を占有すべきことを命じ、Ｂがこれを承諾したときは、Ｃは、甲の所有権の取得を第三者に対抗することができる。

Figure 10: An example of a problem where LUKE provided the correct answer

from "a third party consented to this" in the article, since B is an agent and C is a third party. LUKE was able to predict that the label for this problem would be "*No*". This example suggests that LUKE might be more proficient in understanding personal relationships compared to BERT.

We analyzed the results of our article selection by Sentence LUKE and found an unsuccessful example shown in Figure 11. In this example, our system selected Article 5, "A minor shall obtain the consent of his/her legal representative in order to perform a legal act. Any legal act contrary to the provisions of the preceding paragraph may be revoked," while Article 124-2, item 2 was required to solve the problem. The non-relevant article our system selected shares similar tokens with the problem text, such as "minor" and "consent," but the relevant article also shares these tokens. This may be because abstract paraphrases like "Any legal act contrary to the provisions of the preceding paragraph may be revoked" make the cosine similarities larger. Pretraining and fine tuning on legal documents and paraphrase preprocessing into everyday language may help improve this issue.

6 CONCLUSION AND FUTURE WORKS

We extended our previous system from COLIEE 2022 by performing an ensemble of the rule-based part and the LUKE-based part for COLIEE 2023 Task 4. We discriminated problems into two types based on whether they included alphabetical person names or not, and fine-tuned three different datasets on these two types of problems and all problems. We confirmed that our fine-tuned model for alphabetical person names improved the overall accuracy for

<pair id="R04-01-E" label="Y">
<article>
Article 5, A minor shall obtain the consent of his/her statutory representative before performing a legal act. However, this shall not apply to legal acts merely to obtain rights or to be relieved of obligations.
(2) Any legal act in violation of the provisions of the preceding paragraph may be rescinded.
Article 124, A supplementary acknowledgment of a revocable act shall not be effective unless it is made after the circumstances causing the revocation have ceased to exist and the rescuer becomes aware of his/her right to rescind.
(2) In any of the following cases, the ratification set forth in the preceding paragraph shall not be required to be made after the circumstances that were the cause of the rescission have ceased to exist.
(ii) Where a person with limited capacity to act (excluding an adult ward) (iii) When a person with limited capacity to act (excluding an adult ward) makes a supplementary acknowledgment with the consent of his/her statutory representative, conservator or assistant.
<problem>
A minor who has entered into a contract without the consent of a person with parental authority may not, until he or she reaches the age of majority, follow up on the contract on his or her own without the consent of the person with parental authority.

・The bold text is the important part to solve the problem, our article selection system by Sentence-luke selected below.
A minor shall obtain the consent of his/her statutory representative before performing a legal act. Any legal act in violation of the provisions of the preceding paragraph may be rescinded.

<関連条文>
第五条 未成年者が法律行為をするには、その法定代理人の同意を得なければならない。ただし、単に権利を得、又は義務を免れる法律行為については、この限りでない。
２ 前項の規定に反する法律行為は、取り消すことができる。
第百二十四条 取り消すことができる行為の追認は、取消しの原因となっていた状況が消滅し、かつ、取消権を有することを知った後にしなければ、その効力を生じない。
２ 次に掲げる場合には、前項の追認は、取消しの原因となっていた状況が消滅した後にすることを要しない。
二 制限行為能力者（成年被後見人を除く。）が法定代理人、保佐人又は補助人の同意を得て追認をするとき。
<問題文>
親権者の同意を得ずに契約を締結した未成年者は、成年に達するまでは、親権者の同意を得なければ、自らその契約の追認をすることができない。

・問題を解くためには、関連条文の太字部分が重要であるが、私達の条文選択システムは以下の条文を選択した。
未成年者が法律行為をするには、その法定代理人の同意を得なければならない。前項の規定に反する法律行為は、取り消すことができる。

Figure 11: Examples of article selection failures

those types of problems, achieving 0.69 accuracy in the formal run for COLIEE 2023 Task 4.

Our future work includes improving the data split method and processing other types of problems, as well as working on improving the accuracy of article selection.

7 ACKNOWLEDGMENTS

This research was partially supported by MEXT Kakenhi 00271635, JP22H00804, Japan, and SECOM Science and Technology Foundation.

REFERENCES

[1] 2014. Competition on Legal Information Extraction/Entailment (COLIEE-14) Workshop on Ju-ris-informatics (JURISIN) 2014. (2014). http://webdocs.cs.ualberta.ca/~miyoung2/jurisin_task/index.html

[2] 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

[3] Masaki Fujita, Takaaki Onaga, Ayaka Ueyama, and Yoshinobu Kano. 2022. Legal Textual Entailment using Ensemble of Rule based and BERT based method with Data Augmentaion by Rekated Articke Generation. In *Proceedings of the Sixteenth International Workshop on Jurisinformatics (JURISIN 2022)*. 84–97.

[4] Reina Hoshino, Naoki Kiyota, and Yoshinobu Kano. 2019. Question Answering System for Legal Bar Examination using Predicate Argument Structures focusing on Exceptions. In *Proceedings of the Sixth International Competition on Legal Information Extraction/Entailment (COLIEE 2019)*. 38–42.

[5] Rabelo Juliano, Goebel Randy, Mi-Young Kim, kano Yoshinobu, Yoshioka Masaharu, and Satoh Ken. 2022. COLIEE 2022 Summary: Methods for Legal Document Retrieval and Entailment. In *Proceedings of the Sixteenth International Workshop on Jurisinformatics (JURISIN 2022)*. 1–14.

[6] Yoshinobu Kano, Mi-Young Kim, , Randy Goebel, and Satoh Ken. 2021. *Proceedings of the Competition on Leal Information Retrieval and Entailment Workshop (COLIEE2017) in association with the 16th International Conference on Artificial Intelligence and Law*. 196–210.

[7] Mi-Young Kim, Randy Goebel, Yoshinobu Kano, and Satoh Ken. 2016. *Overview of COLIEE 2017*. 1–13.

[8] Mi-Young Kim, Randy Goebel, and Satoh Ken. 2015. *COLIEE-2015: Evaluation of Legal Question Answering. In Proceedings of the Ninth International Workshop on Juris-informatics (JURISIN2015)*. 1–11.

[9] 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. In *arXiv preprint arXiv:1907.11692*.

[10] Bui Minh-Quan, Nguyen Minh Chau, Dinh-Truong Do, Nguyen-Khang Le, Dieu-Hien Nguyen, and Thi-Thu-Trang Nguyen. 2022. JNLP team: Using deep learning approaches for tackling legal ' s challenges in COLIEE 2022. In *Proceedings of the Sixteenth International Workshop on Jurisinformatics (JURISIN 2022)*. 70–83.

[11] Yoshioka Msaharu, Kano Yoshinobu, Kiyota Naoki, and Satoh Ken. 2018. *Overview of Japanese Statute Law Retrieval and Entailment Task at COLIEE-2018*. 1–12.

[12] Juliano Rabelo, Mi-Young Kim, Randy Goebel, Masaharu Yoshioka, Yoshinobu Kano, and Ken Satoh. 2021. *COLIEE 2020: Methods for Legal Document Retrieval and Entailment*. 196–210. https://doi.org/10.1007/978-3-030-79942-7_13

[13] Goebel Randy, Mi-Young Kim, Yoshinobu Kano, Yoshioka Masaharu, and Satoh Ken. 2021. COLIEE 2019 Overview. In *Proceedings of the Competition on Legal Information Retrieval and Entailment Workshop (COLIEE 2019) in association with the 17th International Conference on Artificial Intelligence and Law*. 1–9.

[14] Goebel Randy, Kano Yoshinobu, Mi-Young Kim, Rabelo Juliano, and Masaharu Yoshioka Ken, Satoh. 2019. *COLIEE 2019 Overview*. 1–9.

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

[16] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention Is All You Need. *CoRR* abs/1706.03762 (2017). arXiv:1706.03762 http://arxiv.org/abs/1706.03762

[17] Sabine Wehnert, Libin Kutty, and Ernesto William De Luca. 2022. Using textbook knowledge for statute retrieval and entailment classification. In *Proceedings of the Sixteenth International Workshop on Jurisinformatics (JURISIN 2022)*. 137–146.

[18] Ikuya Yamada, Akari Asai, Hiroyuki Shindo, Hideaki Takeda, and Yuji Matsumoto. 2020. LUKE: Deep Contextualized Entity Representations with Entity-aware Self-attention. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Online, 6442–6454. https://doi.org/10.18653/v1/2020.emnlp-main.523

[19] Masaharu Yoshioka, Youta Suzuki, and Yasuhiro Aoki. 2022. HUKB at the COLIEE 2022 Statute Law Task. In *Proceedings of the Sixteenth International Workshop on Jurisinformatics (JURISIN 2022)*. 33–46.

HUKB at COLIEE 2023 Statute Law Task

Masaharu Yoshioka
yoshioka@ist.hokudai.ac.jp
Faculty of Information Science and Technology, Hokkaido University
Graduate School of Information Science and Technology, Hokkaido University
Sapporo-shi, Hokkaido, Japan

Yasuhiro Aoki*
yasu-a_01@eis.hokudai.ac.jp
Graduate School of Information Science and Technology, Hokkaido University
Sapporo-shi, Hokkaido, Japan

ABSTRACT

We participated in the statute law task (task 3: information retrieval and task 4: legal textual entailment) of the Competition on Legal Information Extraction/Entailment (COLIEE). For task 3, we modify the system used for the COLIEE 2022, which uses three different IR systems; a BERT-based IR system, an ordinal keyword-based IR system, and an ordinal keyword-based IR system that uses the similarity of judicial decision descriptions between questions and articles. For task 4, we try to include a module that selects the most relevant part of the article for the entailment to make the description of the article concise. We discuss the characteristics of the system using evaluation results for COLIEE 2023 submissions.

CCS CONCEPTS

• **Computing methodologies** → **Information extraction**; *Ensemble methods*; • **Information systems** → *Structured text search*.

KEYWORDS

Information Retrieval, Textual entailment, BERT, Ensemble method

ACM Reference Format:
Masaharu Yoshioka and Yasuhiro Aoki. 2023. HUKB at COLIEE 2023 Statute Law Task. In *Proceedings of COLIEE 2023 workshop: Competition on Legal Information Extraction/Entailment (COLIEE 2023).* ACM, New York, NY, USA, 5 pages.

1 INTRODUCTION

The Competition on Legal Information Extraction/Entailment (COLIEE) [3, 4] serves as a forum to discuss issues related to legal information retrieval (IR) and entailment. There are two types of tasks in COLIEE. One is a task using case law (tasks 1 and 2), and the other is a task using Japanese statute law using Japanese bar exam questions (tasks 3 and 4).

COLIEE 2023, June 19, 2023, Online

HUKB participated in the Japanese statute law task, task 3: information retrieval task, and task 4: textual entailment task. For task 3, we use an ensemble of three different IR systems proposed in the previous COLIEE[8]. This system uses a combination of keyword-based IR system and BERT[1]-based IR system to retrieve the relevant articles with different characteristics. This year, we modify the setting of the BERT training process to achieve better retrieval results. For task 4, we use the IR system for task 3 to select a most relevant part of each article as a pre-process and use it for the BERT-based entailment system proposed in the previous COLIEE[8].

In this paper, we introduce our methods for tasks 3 and 4 in detail and discuss the characteristics of the system using the evaluation results of the submitted runs.

2 TOOLS AND SETTING

Our system for this year's submission is an extension of the system used for COLIEE 2022[8]. For task 3, we use an ordinal keyword-based IR system with different settings. In addition, we tried to include a BERT-based IR system, but it is not effective in COLIEE 2022. For Task 4, our system uses a BERT-based system that classifies the given relevant article pair including the question with data augmentation. We describe the details of the system as follows.

2.1 System for Information Retrieval

There are two types of questions in the statute retrieval task (task 3). One type of question is designed to test the candidate's understanding of a particular article. These questions have in common the terms used in the relevant articles. For such questions, the keyword-based system using such terms works well.

The other type discusses the appropriateness of applying the article to the particular cases. In these questions, the number of common terms is smaller than in the previous type of questions. However, since the relevant article can be used for entailment, the questions and relevant articles may share the similar description about judicial decision. Therefore, we extract the judicial decision part of articles and questions to calculate the similarity between them.

In addition to the comparison of the decision part, it is necessary to identify the correspondence between the ordinal terms for explaining the cases and the technical terms of the legal domain used in the article. Therefore, we use the BERT-based system [7] to identify such relationships.

Based on this understanding, we used three IR systems.

- Keyword-based IR system using BM25 scoring. (Elasticsearch[1])
- Keyword-based IR system that aggregates the BM25 scores of articles and judicial decisions.
- BERT-based IR system.

We use the following three article databases introduced in [8].

- Original article database

  Basic database that uses the original text of the article.
- Expanded article database

  In order to calculate the similarity of judicial decisions, we extract the judicial decision part of the article as metadata of each article. In addition, there are several articles that refer to the other articles and are difficult to understand the meaning by themselves. For example, the articles that explain mutatis mutandis. These articles refer to the multiple articles and describe a new similar concept that is used to replace the target concept in the original articles. For example, article 350 explains mutatis mutandis for 質件 ("right of pledges") and refers to 296 to 300 which discuss 留置権 ("right of retention"). In such a case, we generate combined articles that replace 留置 ("retention") to 質 ("pledges") for 296 to 300 as combined articles. These items are represented as "350+296".
- sub-article database

  In many cases, the articles have two or more sentences and discuss the multiple court decisions. In order to compare the exact case law, we split the text of the expanded articles into sub-articles by considering the item, the sentence, and the existence of the case law.

At the time of making a query from the question, the system extracts judicial decision parts based on the results of dependency structure analysis using CaboCha [2] as we described in [8].

We use Elasticsearch[2] with basic BM25 [6] settings with kuromoji_tokenizer[3] as IR engine. Elasticsearch can use a structured query that can use multiple indexes to compute similarity. We use this feature to use two indexes; one is for all text and the other is for judicial decision part.

For the IR task, BERT model is fine-tuned for a binary classification task that classifies the pair of query and article are relevant or not [7]. In this paper, we use BERT Japanese [4] as a pre-trained language model. For the training process, in addition to pairs of question and relevant articles, we use pairs of same articles (one is used for question part and the other is used for relevant article part) as positive examples. For each positive example, the system randomly selects 10 articles from non-relevant articles as negative examples. After the fine tuning process, the confidence value to classify the relevant articles or not is used for the score for each article.

Finally, the following three IR systems are used to generate the results and we merged the results to generate the final answers.

- Keyword-based IR system with original article database (Original)
  Elasticsearch system that uses questions as queries against the original article database.
- Keyword-based IR system with expanded article database (Expand)
  Elasticsearch system that uses structured queries against the expanded article database. Structured queries consist of queries for the text part and queries for the decision part. If the retrieved article is a combined article (e.g. "350+296"), the system splits the article into two articles (e.g. "350" and "296") and we use the score of the combined article for the one for the splitted article. If there are two or more scores for an item, we use the highest score among them.
- BERT-based IR system with expanded article database and sub articles. (BERT)
  For the BERT-based IR system, pairs of the question with one of the expanded articles and sub-articles are used to calculate the score. Scores for the sub-articles are treated as scores for the original article. Scores for the combined articles are also treated as scores for the split articles. If there are two or more scores for an article, we use the highest score among them.

Each system returns the article(s) with the highest score as candidates for the respective article(s). In most cases, the system returns one article, but if the combined article is the highest ranked one, the system returns two or more split articles.

The final submitted system simply merges the results as the final output. If all systems return the same item, the system returns one item. In other cases, the system returns two or more items as combined results.

2.2 System for Entailment

For task 4, we use a BERT-based entailment system with data augmentation [8]. In this system, the BERT model is refined as a binary classification task that classifies the question-article pair to check whether the articles entail the question or not. In this system, we use the sub-article database of information retrieval as a set of statement of data augmentation. The pair of the same statement is used as a positive example, and the pair of the original statement with the inverted statement is used as a negative example.

The original training data is divided into two types: training data, validation data, and validation data for ensemble. BERT used training data and extended data for fine-tuning

[1] https://www.elastic.co/
[2] https://www.elastic.co/
[3] https://www.elastic.co/guide/en/elasticsearch/plugins/current/analysis-kuromoji-tokenizer.html
[4] https://github.com/cl-tohoku/bert-japanese-whole-word-ma with model https://huggingface.co/cl-tohoku/bert-base-japanese-whole-word-masking

the BERT model by using validation data to find out the best model in the training process. To construct the data set, we first select validation data for ensemble and randomly divide the rest of the original training data into training data (90%) and validation data (10%). We construct 10 models with different random seeds.

After constructing 10 models, we use the validation data for ensemble to check the best set of models for ensemble. Our ensemble models use the output of the entailment system, which is a probability value for the "yes" case. The original BERT system returns "yes" if the value is greater than 0.5. We use the average of this probability value to generate ensemble results instead of a majority vote, because we want to emphasize the result with higher confidence, a larger value for "yes" or a smaller value for "no". We generate results using all possible combinations of these 10 models (two or more) and select the best performing set.

If there are two or more sets with the same highest score, we select the smallest set for the candidate set.

3 SUBMITTED SYSTEMS

3.1 Task3

Our submitted system is based on the system used for COLIEE 2022 explained in section 2.1 with the following modification.

- Construction of training data by selecting similar articles using SentenceTransformer[5] [5].
- The result of the BERT-based IR system is used when the keyword-based IR cannot find appropriate terms to classify whether the articles are related or not.

In the previous system, negative examples are randomly selected from the whole articles. This means that most of these negative examples are about other civil law topics. Therefore, there are smaller examples that are useful to train the corresponding articles from the similar topics. Therefore, we propose a new method to select similar topics for training. In this method, we use a framework of SentenceTransformer to compute the similarity between the query and the articles.

In the pilot study with distributed training data, we try to use the top 10 ranked irrelevant articles for the negative examples. However, the retrieval performance is not stable (good results for retrieving R02 data, but not good for R03 data). Based on these experimental results, we remove the top 10 documents from the negative example candidates and select the next top 10 documents (rank 11 or higher in the original rank) for the negative examples.

Another problem of the previous BERT-based IR system is the inclusion of the unnecessary retrieved results when the keyword-based system can find appropriate articles. We assume that if the keyword-based system can identify the appropriate terms that can distinguish the relevant article from others, the BM 25 scores of the top-ranked documents will be significantly larger than those of the lower-ranked documents.

[5]https://www.sbert.net/index.html

In order to identify such cases, we calculate the ratio of the scores of the top-ranked documents and the 30th-ranked documents ($ratio = score_{1st}/score_{30th}$), which is used to judge whether the keyword-based IR system can identify the appropriate keywords. If the score is less than *threshold* (3 is used for the submission system), the system will use the result of BERT-based IR. But the system will not use the result of BERT-based IR if *ratio* is greater than *threshold*. We call this merge process "strategic merge", and merge process using all top-ranked articles is called "simple merge".

We submit three runs that combine the output of three different IR systems: keyword-based IR system with reconstructed article database (New), keyword-based IR system with original article text (Original), and BERT-based IR system (BERT). We also use two merge processes to produce the final submission results, as follows.

HUKB1 Simple merge results from Original, Expand and BERT

HUKB2 Simple merge results from Original and Expand

HUKB3 Strategic merge results from Original, Expand and BERT

Table 1 shows the evaluation results of our submitted runs and the best runs for each team. HUKB1 is ranked 6th out of 15 submissions. Comparing the results of HUKB1 and HUKB2, we can discuss the effectiveness of using BERT-based IR system. In total, the BERT IR model finds unique articles for 41 questions, and 7 of them are relevant articles. 5 of them are for the questions where the keyword-based IR system cannot find the relevant articles (R04-02-U, R04-03-A, R04-04-I, R04-21-O, R04-36-A) and 2 of them (R04-01-I, R04-19-O) add relevant articles for the questions with multiple relevant articles. The strategic merge is generally effective, the system identifies 12 questions as easy questions. 9 of them are questions for which the keyword-based IR system can find relevant articles. However, 2 of them (R04-01-I, R04-19-O) for the questions with multiple answers and remove 1 question (R04-02-U) that BERT-based IR system can find unique article.

Detailed analysis of the charactersitics of these three system will be discused in the Section 4.

3.2 Task4

We submit the following three runs, using different methods to select text for inclusion, as follows.

HUKB1 All article text is used to select useful sentences for the BERT model to check for entailment.

HUKB2 Keyword-based IR using the expand model and the sub-article database to select useful sentences for the BERT model to check for entailment.

HUKB3 The BERT-based IR model is used to select useful sentences for the BERT model to check for entailment.

To tune the BERT model, we use Adam as the optimizer, cross entropy as the loss function, a training batch size of

Table 1: Evaluation results of the submitted runs (Task3)

sid	F2	Precision	Recall	MAP
CAPTAIN.allEnssMissq	**0.757**	0.726	0.792	0.692
CAPTAIN.allEnssBoostraping	0.747	0.716	0.782	0.692
JNLP3	0.745	0.645	**0.822**	0.710
CAPTAIN.bjpAll	0.742	0.706	0.777	**0.846**
NOWJ.ensemble	0.727	0.682	0.767	0.790
HUKB1	0.673	0.628	0.708	0.740
JNLP2	0.663	0.642	0.703	0.686
HUKB3	0.662	0.650	0.683	0.741
JNLP1	0.657	0.665	0.678	0.686
LLNTUgigo	0.653	0.733	0.644	0.764
HUKB2	0.648	0.678	0.658	0.741
LLNTUkiwiord	0.633	0.703	0.624	0.762
UA.TFIDF_threshold2	0.564	0.620	0.564	0.655
UA.TFIDF_threshold1	0.554	0.634	0.545	0.655
UA.BM25	0.550	0.634	0.540	0.649

32, a maximum number of epochs of 10, and a learning rate of 1e-5.

Table 2 shows the evaluation results of the submitted runs. IDs starting with * are submissions that do not follow the rules of the COLIEE competition. Our system ranks 12th out of 22 submissions.

For this test case, HUKB2 is better than HUKB1 and HUKB3, but the best performing systems of the R01 and R02 datasets are HUKB1 and HUKB2, respectively. It is difficult to say that the sentence selection method proposed in the paper is effective.

4 DISCUSSION

To understand the characteristics of the difference between the three systems, we classify all relevant articles based on the information which system can retrieve the article as relevant (Table 3).

This analysis shows that BERT has different characteristics compared to other keyword based methods. However, for the relevant articles found by two systems, the third system also has a higher rank (mostly for 2 or 3). These articles are not difficult to find with these three systems.

On the contrary, the relevant articles that can be retrieved by one system. There are some articles that are difficult to find with other systems. For example, in the case of Expand, Article 701, which explains mutatis mutandis for R04-28-E, is retrieved only by Expand. There are several questions that require mutatis mutandis articles, but Expand fails to rank the combined articles in 1st place. For this reason, using Expand Articles is not as effective for this year's data. For the BERT only case, Article 122 of R04-04-I and Article 467 of R04-36-A are the ones where the other two systems cannot retrieve the article in the top 5. R04-04-I is a case that uses anonymized symbols and requires semantic matching. R04-36-A is not a question that uses anonymized symbols, but there are other articles that share more keywords with the question than the corresponding article. BERT can be helpful in identifying the important keyword from the context. For the Original only case, Article 334 from R04-12-U is retrieved. However, Article 334 does not share a judicial

Table 2: Evaluation results of the submitted runs (Task4)

ID	Correct	Accuracy
*AMHR02	82	0.8119
JNLP3	79	0.7822
*TRLABS_D	79	0.7822
*TRLABS_I	79	0.7822
JNLP1	76	0.7525
JNLP2	76	0.7525
*TRLABS_T	76	0.7525
KIS2	70	0.6931
KIS1	68	0.6733
UA_V2	67	0.6634
AMHR01	66	0.6535
KIS3	66	0.6535
AMHR03	65	0.6436
LLNTUdulcsL	63	0.6238
UA	63	0.6238
HUKB2	60	0.5941
CAPTAIN.gen	59	0.5842
CAPTAIN.run1	58	0.5743
LLNTUdulcsS	57	0.5644
HUKB1	56	0.5545
HUKB3	56	0.5545
LLNTUdulcsO	56	0.5545
NOWJ.multi-v1-jp	55	0.5446
CAPTAIN.run2	53	0.5248
NOWJ.multijp	53	0.5248
NOWJ.multi-v1-en	49	0.4851

Table 3: Number of articles classified by the retrieved system

Systems	Number of Articles
No	45
Expand only	2
Original only	1
BERT only	7
Expand + Original	18
Expand + BERT	6
Original + BERT	1
All	50
Total	130

decision with the question (another relevant article, Article 330, shares a judicial decision with the question). This is an example of how a keyword-based search without consideration of decision can be helpful in finding secondary relevant articles that do not share the judicial decision of the question.

Although the number of unique contributions of these three systems is small, this analysis shows that each system has different characteristics that complement each other. Further discussion is needed to utilize the characteristics of these systems.

5 SUMMARY

In this paper, we have introduced our system to participate in tasks 3 and 4 of COLIEE 2023. For task 3, we extend our system by introducing the new method to construct the training data and the new strategy to merge the IR results. For this year's data, we confirm that the new BERT model helps to find the relevant article that cannot be retrieved by the ordinal IR system. However, the performance is not so good compared to the best performing system. For task 4, we proposed a system that uses sentence selection as a pre-process for entailment. However, it is difficult to judge whether this approach is effective or not.

REFERENCES

[1] 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

[2] Taku Kudo and Yuji Matsumoto. 2002. Japanese Dependency Analysis using Cascaded Chunking. In *CoNLL 2002: Proceedings of the 6th Conference on Natural Language Learning 2002 (COLING 2002 Post-Conference Workshops)*. 63–69.

[3] 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* (2022), 111–133. https://doi.org/10.1007/s12626-022-00105-z

[4] Juliano Rabelo, Mi-Young Kim, Randy Goebel, Masaharu Yoshioka, Yoshinobu Kano, and Ken Satoh. 2021. COLIEE 2020: Methods for Legal Document Retrieval and Entailment. In *New Frontiers in Artificial Intelligence. JSAI-isAI 2020. Lecture Notes in Computer Science*. Springer International Publishing, Cham, 196–210.

[5] 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*. Association for Computational Linguistics. http://arxiv.org/abs/1908.10084

[6] S. E. Robertson and S. Walker. 2000. Okapi/Keenbow at TREC-8. In *Proceedings of TREC-8*. 151–162.

[7] Wataru Sakata, Tomohide Shibata, Ribeka Tanaka, and Sadao Kurohashi. 2019. FAQ Retrieval Using Query-Question Similarity and BERT-Based Query-Answer Relevance. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval* (Paris, France) *(SIGIR'19)*. Association for Computing Machinery, New York, NY, USA, 1113–1116. https://doi.org/10.1145/3331184.3331326

[8] 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.

AMHR Lab 2023 COLIEE Competition Approach

Onur Bilgin, Logan Fields, Antonio Laverghetta Jr.*
Zaid Marji, Animesh Nighojkar, Stephen Steinle*
John Licato
onurbilgin,ldfields,alaverghett@usf.edu
zaidm,anighojkar,ssteinle@usf.edu
licato@usf.edu
Advancing Machine and Human Reasoning Lab
Department of Computer Science and Engineering
University of South Florida
Tampa, FL, USA

ABSTRACT

We report on our submissions for Task 4 of the 2023 COLIEE competition. Our approach was to use prompt engineering techniques with large pre-trained language models, that were not fine-tuned for the task. Our most successful strategy used simple text similarity measures to retrieve articles and queries from the training set. We report on our efforts to optimize performance with both OpenAI's GPT-4 and FLaN-T5. We then used an ensemble approach to find the best combination of models and prompts. We also report on our attempts to understand our results and suggest ideas for future improvements.

KEYWORDS

AI, NLP, Reasoning, Law, Legal

ACM Reference Format:
Onur Bilgin, Logan Fields, Antonio Laverghetta Jr., Zaid Marji, Animesh Nighojkar, Stephen Steinle, and John Licato. 2023. AMHR Lab 2023 COLIEE Competition Approach. In *Proceedings of COLIEE 2023 workshop, June 19, 2023, Braga, Portugal.* ACM, New York, NY, USA, 10 pages.

1 INTRODUCTION

If we hope for AI systems to have a robust understanding of the instructions they are given or the rules they must follow, we must advance the science of how human-written rules can be automatically reasoned over and resolved. Any time a governing law, mission order, code of ethical conduct, or other verbal or written instruction is produced and given to a subordinate in a fixed, referable form (a "rule"), there is some expectation that it will be followed in the spirit in which it was created. Often this means there is an assumption (or hope) that the rule's intent is adequately conveyed. However, the complete conveyance of a rule's intent requires a multitude of background knowledge: the history behind the statement, prototypical examples of its proper and improper interpretations, the intended goals of the rule's creator, the proper scope of the rule's open-textured predicates, and so on [16, 17, 27, 28, 41, 43].

Carrying out such reasoning is a challenging task, even for state-of-the-art artificially intelligent language models (LMs). A primary reason for this difficulty is the prevalence of *open-textured terms* (OTTs)—terms whose extensions are not completely and unambiguously fixed at the time of their initial use [17]. For example, consider a traffic regulation stating that vehicles must "keep to the right as far as is *reasonably safe*" [41]. Such a regulation would require interpretation by autonomous driving vehicles or traffic enforcement bots. However, it is implausible to exhaustively list an exception-free accounting of all possible scenarios and conditions that can be considered instances of the open-textured term "reasonably safe"—any such attempt would inevitably limit the scope of the regulation and render it fatally inflexible in the face of unpredictable conditions. Using OTTs is a necessary and unavoidable feature of regulatory and legal language [16, 17, 27, 28, 41, 43, 51]. Thus ways to work with them must be addressed by any sufficiently robust account of compliance detection.

There are multiple approaches for addressing this problem in AI research. The first approach is to reduce the open-texturedness of the rules so that they can be reasoned over using transparent algorithms and formal methods. For example, our lab has recently explored the translation of rules (containing OTTs) in a collectible card game into programming language code [26], which can then allow for reasoning over the code and the game itself [15]. The second approach is to *embrace open-texturedness*; i.e., to accept that no approach will ever entirely remove the open-texturedness of languages in rules (and to acknowledge that rule systems with no open-texuredness is not desirable, either), and to instead focus on how to reason over OTTs in their natural language forms, without forcing translations into unambiguous formal languages.

Under this second approach, there are multiple sub-approaches. One of these takes the position that for artificially intelligent systems to follow human-written rules properly, they need to be able to interpret them, which requires resolving OTTs. Furthermore, the interpretation the AI chooses should be provided in a form that stakeholders can inspect, test, and use as precedent for future interpretations [23–25]. In other words, given text to be interpreted by an AI, human stakeholders need to be able to inspect: (I1) *how* the AI interpreted that text, and (I2) *why* the AI believes that interpretation

*The first six authors contributed equally to this research and are listed in alphabetical order.

COLIEE 2023, June 19, 2023, Braga, Portugal

is best. An emerging body of work is exploring approaches to (I2) under the topic of *interpretive argumentation* [2, 23, 28, 47, 53, 54].

Given the difficulty of generating and evaluating interpretive argumentation, the present COLIEE competition is a helpful stepping-stone towards that ultimate goal. We submit our entries to this competition with that framing in mind.

Task Description. We focused on Task 4 of the COLIEE 2023 competition. This task was formulated as follows: We are given a set of articles A and a query q. Each article is a short text snippet which is a statute from the Japanese Civil Code translated into English, normally consisting of no more than a few sentences. The query is a short textual description of something that may or may not be true given the articles; e.g., "There is a limitation period on pursuance of warranty if there is restriction due to superficies on the subject matter, but there is no restriction on pursuance of warranty if the seller's rights were revoked due to execution of the mortgage." Our algorithms must output either 'Y' or 'N', depending on whether q follows from A.

Overview of Results. We submitted three entries into the 2023 COLIEE competition. One of them, `AMHR02`, deliberately used GPT-4—although this was a disallowed resource, we wanted to see how well it performed compared to other existing tools. Our `AMHR01` submission used Flan-T5, a language model that had been instruction fine-tuned and which did well on the previous years' datasets, which we used as a validation set (in keeping with the rules, we did not consider the test set R04 at all in selecting our models or hyperparameters). Finally, our `AMHR03` submission was an ensemble approach that tried to combine the recommendations made by various models and hyperparameter settings.

2 BACKGROUND

Methods for In-Context Learning. In recent years, advances in NLP research have been dominated by large language models (LLMs), which have tens or even hundreds of billions of parameters [5]. These models are able to solve new tasks *few-shot*, where the model is given only a small number of training examples yet can achieve strong task performance [6]. This has enabled a new paradigm of NLP engineering, where experts interact directly with LLMs and train them to solve tasks via *prompt engineering*,[1] whereby an input context is discovered, either by manual engineering or via a search algorithm, and used to prompt the model [31]. In this work, we explore two broad categories of prompts: those which focus on finding a combination of training examples (shots) to use as context (*prompt retrieval*) and those based on *chain-of-thought* prompting [58], where instructions given to the model are elaborated to induce more reliable and accurate behavior. Note that both these approaches may be used simultaneously to boost performance further [61]. Prompt retrieval aims to find the optimal strategy for selecting the training examples to use as context. Prior work has employed supervised models trained to predict the most informative shots (e.g., [42, 45, 49]). Others have used unsupervised models based on similarity metrics, for example, BM25 [56] or SBERT [44]. On the other hand, chain-of-thought approaches are meant to help a model "think step-by-step" to arrive at a correct answer and thus embed the correct answer to tasks as well as the reasoning process used to arrive at those answers [58]. Other prompting methods falling into this class include faithful chain-of-thought [34], self-taught reasoning [64], and maieutic prompting [19]. These methods have enabled high few-shot performance on challenging benchmarks of linguistic reasoning [50]. Chain-of-thought and self-taught reasoning have been used in a previous COLIEE competition [63].

Open-texturedness in Machine Learning. Machine learning has the potential to drastically improve equity in the application of laws across racial, socioeconomic, and other categorical features. Where human judges and legal scholars may be influenced by biases [4, 30], a sufficiently-trained machine learning model may be able to objectively recognize the features of a case and render an equitable decision. However, models rarely improve equity in practice because of bias preservation in the models' training [35, 52, 62] and open-texturedness in legal terminology.

OTTs are nearly ubiquitous within legal reasoning [3, 13], where laws may have overarching downstream impacts and disagreements about their scopes are often resolved within appellate courts by expert judges. However, because the interpretation of open-texturedness relies on the discretion of "reasonable humans," which is known to suffer from biases, it presents a considerable challenge for AI models to interpret such terms in a human-like manner without perpetrating those same biases.

3 APPROACH 1: GPT-4

LLMs [6, 60] trained for text generation tend to outperform humans on various professional and academic benchmarks. Some of these models have been tuned to behave like "chatbots", preserving conversation history and adhering to instructions. OpenAI developed a product, GPT-4,[2] that can be used as a chatbot through their API.[3] Given a chat conversation, the API returns a chat completion response, allowing the user to set both the human's and the model's previous responses. The API also allows the user to set a "system" prompt, which persists throughout the "conversation" and helps set the model's behavior.[4] Our use of the API is illustrated in Figure 1. According to OpenAI, GPT-4 scores around the top 10% of test takers on a bar exam, though they provide very few details on this exam and how the model was used to make predictions on it. GPT-4 is also instruction-tuned [38] using reinforcement learning from human feedback (RLHF) [10] to follow a variety of written instructions. This also improves zero-shot performance (especially on classification tasks [57]) of the model because examples (shots) are no longer required to *show* the expected format of responses, the user can just *tell* what the format should be (more details on this training strategy are provided in Section 4). We used the OpenAI API to experiment with multiple types of prompts, all of which are illustrated in Figure 1. We tried zero-shot and few-shot variants of each, and the results are shown in Table 1.

[1] Also called *prompt tuning*.

[2] https://openai.com/research/gpt-4

[3] https://platform.openai.com/docs/api-reference/chat

[4] https://platform.openai.com/docs/guides/chat

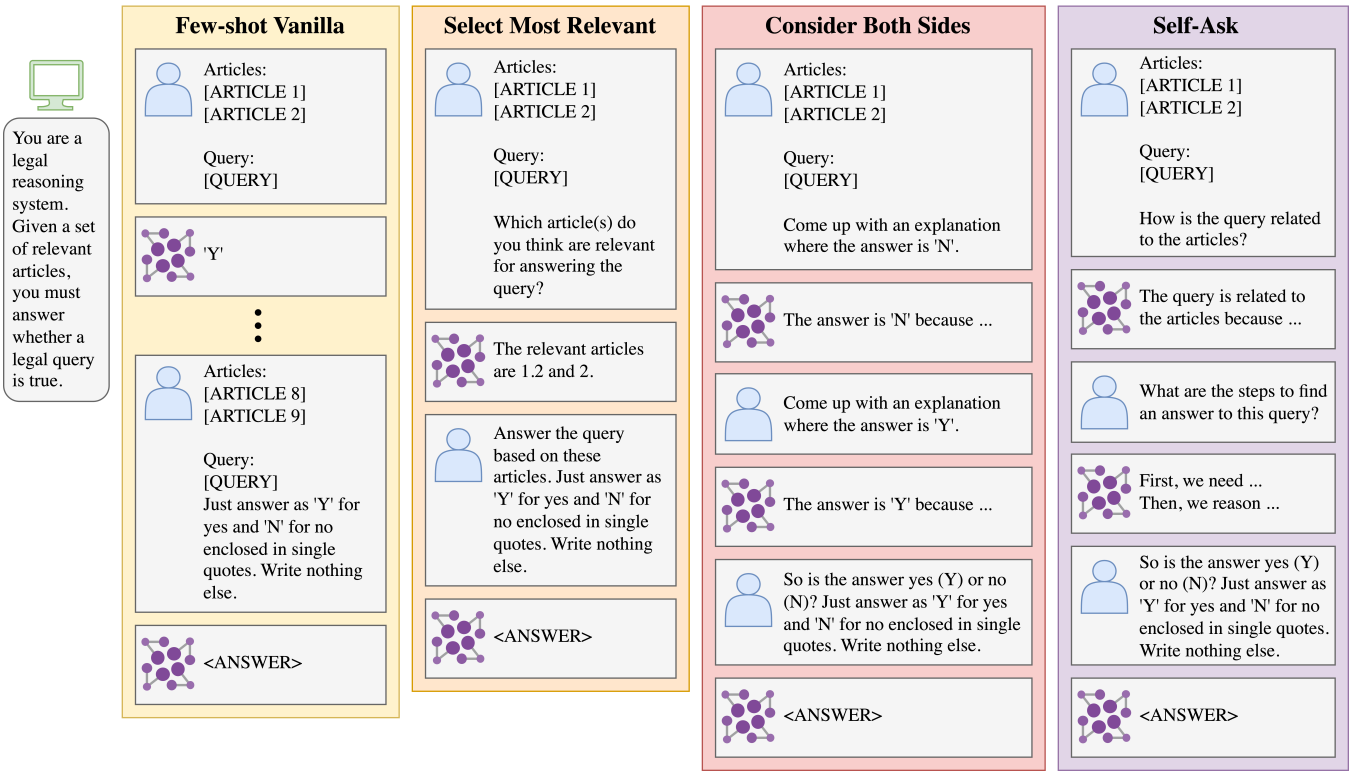


Figure 1: Prompt structures tested for LMs. Suppose this is considered to be a text conversation. In that case, the green computer indicates the system prompt (which was the same for all prompt structures), the blue person indicates the human *messages*, and the purple graph indicates the model's *responses*. Note that not all models support a system prompt. In these cases, the system prompt is appended to the beginning of the human messages.

4 APPROACH 2: INSTRUCTION-TUNED TRANSFORMERS

We use the Flan-T5[5] and T0[6] checkpoints publicly available on HuggingFace and use the transformers library [59] to perform prompt-tuning using the models. Our work used `flan-t5-xxl`, `T0`, `T0p`, `T0pp`, and `T0-3B` checkpoints. We use the text generation pipeline provided by the library and prompt-tune the model to generate the correct label, given the validation example articles and query and an optional number of training shots. We use a simple regex pattern to detect if the model generated a correct label. Given the model's raw outputs, we convert all text to lowercase, strip out leading and trailing spaces and newlines, and check if the output is any of the following strings:

(1) If the correct label is "Y":
  (a) "y", "yes", "the answer is yes"
(2) If the correct label is "N":
  (a) "n", "no", "the answer is no"

We found that `flan-t5-xxl` was quite well-behaved on this task and, in the overwhelming majority of cases, generated only the label string and thus did not require careful pattern matching to avoid false negatives. This was primarily the same with T0; however, the smaller models did not always consistently generate only the label (e.g., "the answer is yes"), and we thus added other valid patterns as possible strings to match. In cases where a valid label was not detected or the model predicted the wrong label, we marked the example as incorrect. We turn off sampling in all experiments to force determinism in all generations so that the prompt only affects the output.[7] We use a maximum sequence length of 512, the longest the models support. We experiment with the same chain-of-thought and similarity-based shot selection strategies as in our GPT-4 system (Figure 1). However, unlike GPT-4, we found that the model's predictions for chain-of-thought were purely extractive. For example, when the prompt indicated the models should generate reasons for the answer to be either 'Y' or 'N,' the model's rationales were extracted verbatim from either the articles or the query. This behavior, coupled with the much smaller maximum sequence length in these models, caused very poor performance in our chain-of-thought prompts, and we did not investigate them in detail for our Huggingface models.

[5]https://huggingface.co/google/flan-t5-xxl
[6]https://huggingface.co/bigscience/T0pp

[7]Because sampling was disabled, the temperature does not affect these models' predictions.

5 APPROACH 3: ENSEMBLE

Ensembles are a combination of multiple models used to achieve a higher prediction accuracy or better generalization because the different classifiers in the ensemble may have sensitivity for a different set of samples and have learned different subsets of features. By combining different classifiers, the goal is to reduce bias and error and to increase prediction performance [14]. Recent work has applied ensembles to the prompting of LMs, for example, by combining multiple prompts into an "ensemble of prompts" using multiple prompts with the same model and combining the predictions from each using a meta-classifier [1, 68]. We reasoned that applying this idea to our Huggingface models might boost performance even more.[8]

We tested two approaches for our ensemble model. In the first approach, we applied brute force search to select the best combination of models with the highest validation accuracy. Thus, we created an ensemble dataset from the validation set, where the features are the model predictions for each sample in the validation set. Then, for each combination of models, we calculated the model's accuracy using majority voting and selected the ensemble with the highest accuracy on the validation set. The models in our best performing brute force ensemble are three `flan-t5-xxl` runs with balanced TF-IDF and two shots, a `flan-t5-xxl` model with TF-IDF unbalanced and three shots, and `T0`, `T0p`, and `T0 3B` each zero-shot. The resulting ensemble consists of 7 models. Further details on the shot selection strategies used by these models are found in Section 6.

In the second approach, we aggregated TF-IDF vectors trained on the validation set as additional features besides model predictions. Here, we reduced the vocabulary size to 10% based on TF-IDF scores to reduce the feature space. Then we applied 5-fold cross-validation for the validation set. We used support vector machines (SVMs) [12] and random forest [18] models for the training, both of which were implemented in scikit-learn [39]. The parameters are chosen based on the validation set results. For the SVM models, the radial basis function kernel is used, and the regularization parameter is 1.0. For our random forest model, the maximum depth of the tree is 5, with a total of 100 estimators. The Gini impurity is used to measure the quality of the split.

6 OVERALL TRAINING PROCEDURE

Per the competition specifications, we use the past four years datasets (H30, R01, R02, R03) as validation datasets and older years' datasets as training datasets. We use only few-shot learning to tune all LMs, no additional pre-training, finetuning, or other forms of gradient updates were applied to any prompted model, and our ensembles were trained only on the outputs of the LMs and content of the articles and query in the training data. No external data was used to train any of the submitted systems beyond the data used to pre-train the LMs.[9] We selected several LMs for initial testing. Specifically, we tested RoBERTa-large [32], which has shown high performance on natural language understanding (NLU) tasks [32], LegalBERT [8], which is pre-trained on legal data, and has been a pivotal contribution in previous COLIEE competitions, Meta's OPT [65], Google's largest Flan-T5 [11], the T0 models from BigScience [46], EleutherAI's GPT-J [55], and OpenAI's ChatGPT [36] and GPT-4 [37]. We found that LegalBERT, RoBERTa, GPT-J, and OPT performed relatively poorly and hence chose not to run extensive ablations on them. Chat-GPT performed reasonably well, matching or exceeding the performance of both Flan-T5 and T0. However, because we believed that any OpenAI submission would likely be disqualified, we chose to use GPT-4 instead as our only submission using their API. However, we use results for Chat-GPT in some of our ablations reported below.

Across all LMs, we experimented with the following strategies for few-shot selection:

(1) **Zero-shot:** The model is given only the validation example without any further context.
(2) **Few-shot no TF-IDF:** The model is randomly given k shots from the training data. The number of shots varies from two to six, depending on the model.
(3) **Few-shot with TF-IDF:** Prior work has demonstrated that the choice of shots sent to an LM significantly impacts model performance [7, 21]. Therefore, choosing the shots based on some metric is important for optimizing model performance. Following prior work [33], we use similarity-based shot selection based on the cosine similarity of TF-IDF vectors. The validation example (articles + query) is embedded using a TF-IDF vector space, and the top k most similar examples are chosen and used as context. This is done for each validation example separately, and the exact order in which shots are presented is random. We use the training data articles and queries to train our TF-IDF vector space.
(4) **Few-shot balanced with TF-IDF:** Same as above, except the shot selection always returns a balanced number of entailment and non-entailment shots to prevent the model from overfitting on one label.
(5) **Few-shot Pruned with TF-IDF:** When the TF-IDF vectors are calculated with the complete vocabulary, they are quite sparse due to the lack of overlap among terms across documents. Therefore, we also explored applying pruning to the vectors before performing cosine similarity. After building the full TF-IDF matrix, we reduced the vocabulary by X% and rebuilt it based on this smaller size, where X is a hyperparameter. Note that these approaches always used the unbalanced form of TF-IDF shot selection. As we found pruning always performed worse than the unpruned unbalanced TF-IDF, we chose not to investigate this strategy further.

Besides intelligent shot selection, prior work has also found that how the prompt is structured can significantly impact downstream performance. For example, advanced prompting strategies, including chain-of-thought [58] and maieutic prompting [19], involve asking a model a series of structured questions in order to help it arrive at a correct answer. We experimented with several such approaches with our models: (1) Select Most Relevant, (2) Consider Both Sides, and (3) Self-Ask. Figure 1 shows examples of each prompt type. Each of these approaches involves asking the model to explain why the answer should be yes (or no), or asking the model to select the most relevant information from the articles to answer the query, both

[8] As we suspected our GPT-4 submission would likely be disqualified, we chose not to use this model in the ensemble.
[9] We refer the reader to the respective papers for details on how each model was trained. Note that, at the time of publication, OpenAI has released no details on what data GPT-4 was trained on.

of which we reasoned could aid the model in choosing the correct answer.

7 RESULTS

Table 1 shows GPT-4 and Flan-T5 systems results. We compare our systems against the results reported by the Kano Laboratory, which achieved first place in the 2022 Task 4 competition [20]. Not surprisingly, our GPT-4 based submission surpasses the prior state-of-the-art by significant margins across all validation splits and achieves an overall accuracy of 83.3%, almost 20 points higher than the prior year's results. Perhaps more surprising, however, is the strong performance achieved using `flan-t5-xxl`. This model has only 11 billion parameters.[10] While this model is still considerably larger than the prior year's submission, we emphasize that it was not directly trained on the train set. The best Flan-T5 model uses only two shots for in-context learning, which is less than 1% of the entire train set, though new shots are sampled each time. Collectively, these results demonstrate the potential of applying prompting with generative LMs to legal reasoning tasks and show that even a relatively simple prompting strategy can outperform carefully tuned systems across multiple validation datasets. Results for each of our submitted systems on the R04 test data set are given in Table 2.

For our ensembles, the best-performing brute force approach achieved 77.0% accuracy on the validation set. With support vector machines, we achieved an ensemble accuracy of 74.9%, and for random forest, an accuracy of 74.4%, as shown in Table 3. According to the results, our brute force approach achieved better accuracy on the validation set by intensively searching for the best model combinations across the ensemble set, outperforming the ensemble's performance with support vector machines and random forest models. In our opinion, the lack of more data prevented achieving higher accuracy for those models. Thus, we selected our brute force approach as our ensemble model based on the overall validation accuracy for our submission and achieved on the test set 64.4% accuracy. We think our model combination is overfitting the validation set and therefore caused the accuracy difference between the validation and test set.

7.1 Ablations

7.1.1 Shot Selection Strategy. For both GPT-4 and Flan-T5 models, we found that TF-IDF selection balanced by label achieved the highest validation accuracy (2% increase for GPT-4, 1.3% increase for Flan-T5 compared to 0-shot prompting). Using TF-IDF with an unbalanced label selection causes a slight decrease in performance for Flan-T5. This is not surprising; prior work has found that in-context learning is highly sensitive to the prompt [9, 40, 67], and not balancing labels likely causes overfitting to the majority class in the prompt. Additionally, we found that few shot prompting with randomly chosen shots caused a substantial decrease in performance. As each validation example contains a significant amount of terminology that may not appear elsewhere, it is unclear how much information an arbitrarily chosen train example will provide for determining the label for a validation example (i.e., knowledge of entailments on contract law likely provides little information for inference on a query related to the rights of the unborn). Our results show that some sort of intelligent shot selection is necessary for few-shot learning to help.

As a selection strategy, TF-IDF is a fairly simple approach that relies on syntactic overlap among documents [22]. However, information retrieval research has developed more sophisticated methods for document similarity that employ modern contextual embeddings. Such approaches include BERTScore [66], SBERT [44], and BLEURT [48], among others. We, therefore, explored using each of these approaches as the similarity method for shot selection to see if an approach based on contextual embeddings could outperform the simpler TF-IDF vectors. We use Chat-GPT for this ablation,[11]. We use standard prompting (no chain-of-thought methods), five shots, and a temperature of one for all models and compute the overall accuracy of each approach across all validation splits. Results are shown in Table 4. Although all similarity-based selection methods perform better than random selection, TF-IDF achieves the best overall accuracy. One possible explanation is that the TF-IDF vectors were the only ones trained on a legal corpus (the train set), and we relied on the pre-trained embeddings for all other methods. It is conceivable that pre-training the contextual similarity metrics on a corpus of legal documents could dramatically improve the quality of the selected shots. However, doing this is impossible with just the train set as these methods require considerably more data than TF-IDF.

Finally, we found that pruning the TF-IDF vectors to select only terms with low document frequency consistently leads to worse validation accuracy. Our goal behind this method was to eliminate terms that appeared across most training examples (likely stopwords) and create better vectors for document ranking. However, it appeared to have the opposite effect. If the pruning was too aggressive, all vectors could have become orthogonal if no terms overlapped across documents. This pattern was observed regardless of the pruning factor, even with minimal pruning.

7.1.2 Advanced Prompting Strategies. We investigated various more sophisticated prompting methods (details of prompt structure discussed in Section 6). We focus on GPT-3.5 for our analysis because, as discussed earlier, the minimal sequence length of Flan-T5 prevented it from using any chain-of-thought approach effectively. Results are shown in Table 2. We find that no prompt outperforms the "vanilla" baseline strategy. Given that legal reasoning often involves highly open-textured phrases, the space of possible explanations may be so extensive that chain-of-thought approaches cannot effectively assist a model in arriving at a correct answer, which confirms with prior research on these prompting strategies in other specialized domains [29].

7.1.3 Choice of Temperature. LMs sample words from their vocabulary and choose which word to predict next—given a sequence—from this sample. Temperature is a hyperparameter (varying between 0 and 1) that controls the randomness of this choice. Lower values decrease randomness, and higher ones increase it. We experimented with multiple temperature values from the set 0, 0.25, 0.5, 0.75, and 1.0. However, we found that different values did not significantly

[10] Although the exact number of parameters in GPT-4 is unknown, it is likely to be on the order of hundreds of billions, given the known size of GPT-3.

[11] `'gpt-3.5-turbo'`

Model	Prompt Type	Shots	Similarity Method	Temperature	Ensemble Size	H30	R01	R02	R03	Train	Overall Acc	Seq Length	Pruning Factor
Kano Lab 2022 [20]	n/a	n/a	n/a	n/a	n/a	**70.000%**	**63.960%**	**54.320%**	**67.890%**	n/a	**64.101%**	n/a	n/a
GPT-4	**Vanilla**	**6**	**TF-IDF**	**1**	**5**	**82.857%**	**80.180%**	**81.481%**	**88.073%**	**n/a**	**83.288%**	**2048**	**n/a**
GPT-4	Vanilla	6	TF-IDF Balanced	1	5	81.429%	77.477%	80.247%	88.073%	n/a	81.941%	2048	n/a
GPT-4	Vanilla	6	TF-IDF Balanced	0, 0.25, 0.5, 0.75, 1.0	5	81.429%	79.279%	80.247%	88.073%	n/a	82.480%	2048	n/a
GPT-4	Vanilla	0	None	0, 0.25, 0.5, 0.75, 1.0	5	80.000%	81.081%	83.951%	86.239%	n/a	83.019%	2048	n/a
GPT-4	Vanilla	5	TF-IDF	1	1	81.429%	83.784%	80.247%	88.991%	n/a	84.097%	2048	n/a
GPT-4	Vanilla	0	None	1	1	80.000%	76.577%	83.951%	86.239%	n/a	81.671%	2048	n/a
GPT-4	Vanilla	0	None	0	1	77.143%	80.180%	82.716%	85.321%	n/a	81.671%	2048	n/a
T0	Vanilla	0	None	n/a	n/a	52.857%	55.856%	64.198%	65.138%	61.120%	59.839%	512	n/a
T0p	Vanilla	0	None	n/a	n/a	52.857%	59.460%	65.432%	66.972%	61.280%	61.725%	512	n/a
T0-3b	Vanilla	0	None	n/a	n/a	52.857%	55.856%	65.432%	58.716%	56.640%	58.221%	512	n/a
Flan-T5	Vanilla	3	TF-IDF	n/a	n/a	68.571%	72.072%	77.778%	74.312%	69.440%	73.315%	512	n/a
Flan-T5	Vanilla	2	TF-IDF Balanced	n/a	n/a	67.143%	71.117%	79.012%	77.817%	69.992%	74.059%	512	n/a
Flan-T5	**Vanilla**	**2**	**TF-IDF Balanced**	**n/a**	**n/a**	**70.000%**	**71.117%**	**80.247%**	**79.817%**	**71.200%**	**75.456%**	**512**	**n/a**
Flan-T5	Vanilla	2	TF-IDF Balanced	n/a	n/a	64.857%	65.765%	65.432%	71.559%	n/a	n/a	512	n/a
Flan-T5	Vanilla	4	TF-IDF	n/a	n/a	60.000%	60.360%	59.260%	67.890%	70.720%	n/a	512	n/a
Flan-T5	Vanilla	2	TF-IDF	n/a	n/a	60.000%	66.667%	65.432%	70.642%	69.920%	n/a	512	n/a
Flan-T5	Vanilla	4	TF-IDF	n/a	n/a	58.571%	62.162%	58.024%	67.890%	69.440%	n/a	512	n/a
Flan-T5	Vanilla	2	TF-IDF	n/a	n/a	62.857%	63.963%	65.432%	72.477%	n/a	n/a	512	n/a
Flan-T5	Vanilla	2	TF-IDF Pruned	n/a	n/a	64.286%	67.568%	72.400%	75.230%	70.080%	70.255%	512	0.15
Flan-T5	Vanilla	2	TF-IDF Pruned	n/a	n/a	65.714%	70.270%	76.543%	76.147%	71.680%	72.507%	512	0.3
Flan-T5	Vanilla	2	TF-IDF Pruned	n/a	n/a	65.714%	71.117%	74.074%	75.230%	70.240%	71.952%	512	0.45
Flan-T5	Vanilla	2	TF-IDF Pruned	n/a	n/a	68.571%	72.973%	77.778%	76.147%	70.720%	74.124%	512	0.6
Flan-T5	Vanilla	2	TF-IDF Pruned	n/a	n/a	68.571%	72.973%	76.543%	77.064%	69.920%	74.124%	512	0.75
Flan-T5	Vanilla	2	TF-IDF Pruned	n/a	n/a	64.286%	67.568%	77.778%	75.229%	70.080%	71.429%	512	0.9
GPT-3.5	Select most relevant	0	None	1	1	61.429%	60.360%	62.963%	67.890%	n/a	63.342%	2048	n/a
GPT-3.5	Select most relevant	5	None	1	1	65.714%	63.964%	72.840%	72.477%	n/a	68.733%	2048	n/a
GPT-3.5	Select most relevant	5	TF-IDF	1	1	70.000%	63.964%	70.370%	66.055%	n/a	67.116%	2048	n/a
GPT-3.5	Consider both	0	None	1	1	54.286%	59.459%	55.556%	60.550%	n/a	57.951%	2048	n/a
GPT-3.5	Consider both	5	None	1	1	67.143%	51.351%	48.148%	55.963%	n/a	54.987%	2048	n/a
GPT-3.5	Consider both	5	TF-IDF	1	1	47.143%	54.955%	49.383%	56.881%	n/a	52.830%	2048	n/a
GPT-3.5	Self-Ask	0	None	1	1	65.714%	65.766%	60.494%	69.725%	n/a	65.768%	2048	n/a
GPT-3.5	Self-Ask	5	None	1	1	68.571%	60.360%	65.432%	69.725%	n/a	65.768%	2048	n/a
GPT-3.5	Self-Ask	5	TF-IDF	1	1	67.143%	62.162%	58.025%	68.807%	n/a	64.151%	2048	n/a
GPT-3.5	Self-Ask	0	None	1	1	70.000%	65.800%	66.700%	75.200%	n/a	69.551%	2048	n/a
GPT-3.5	Self-Ask	3	TF-IDF	1	5	71.000%	61.000%	68.000%	69.000%	n/a	66.765%	2048	n/a
GPT-3.5	Self-Ask	1	TF-IDF	1	5	66.000%	63.000%	63.000%	68.000%	n/a	65.035%	2048	n/a

Table 1: Summary of prompting results. Top most row shows results of the best performing model from the 2022 Task 4 competition. The first section shows GPT-4 ablations, the second section shows results from models included in the ensemble (excluding the submitted Flan-T5-xxl model), the third section shows Flan-T5-xxl ablations, and the final section shows ablations for the advanced prompting strategies with GPT-3.5. Underlined rows indicate the submitted Flan-T5-xxl and GPT-4 models.

System	Test Accuracy
Flan-T5	65.35%
GPT-4	**81.19%**
Ensemble	64.36%

Table 2: Test accuracy using different models.

Ensemble Model	Overall Accuracy
Brute Force	**77.0%**
SVM	74.9%
Random Forest	74.4%

Table 3: Overall accuracy across validation splits using different ensemble models.

Shot Selection Strategy	Overall Accuracy
SBERT	69.542%
BERTScore	66.307%
BLEURT	68.194%
TF-IDF	**72.237%**
Random	66.038%

Table 4: Overall accuracy across validation splits using different contextual similarity metrics. All results use Chat-GPT, with standard prompting, five shots, and a temperature of one.

affect the performance of GPT-4. Building on this, we also experimented with temperature-based ensembles where each prediction (vote) came from a GPT-4 model with a different temperature value. We used the same set of temperatures (five in total), and the majority vote was chosen as the final prediction of the ensemble. However, we found that this approach also did not significantly improve the performance of the GPT-4 models. Results from these trials are shown in the first half of Table 1, with the rows containing multiple values indicating temperature-based ensembles.

7.2 Error Analysis

We perform error analysis on our systems, focusing on GPT-4 and Flan-T5. Table 5 lists the false positive rate (FPR) and false negative rate (FNR) of the best-performing Flan-T5 and GPT-4 models on the validation sets. We find that models tend to overpredict 'Y,' which leads to a much higher FPR than FNR. Interestingly, most of the gains from GPT-4 appear to be from reducing the FNR; this metric drops to almost 0, while the FPR drops by only 10 points and remains relatively high. Scaling up models thus does not appear to eliminate this problem. The exact cause of this behavior is unclear. The validation splits are somewhat unbalanced, but not by a sufficient degree to cause such a significant imbalance in error rates. Our shot selection also accounted for this by ensuring the labels were balanced. We leave further analysis of this behavior to future work.

In Figure 2, we graph a histogram of the proportion of non-ensemble models that get each validation example correct. The goal is to determine if there are examples that are consistently difficult to solve. We find that this is the case and that the distribution of accuracy scores is roughly bi-modal. Most examples appear relatively

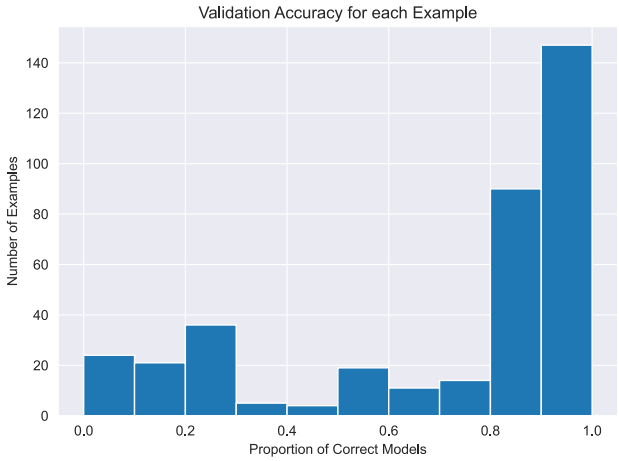


Figure 2: Histogram of model accuracy for all examples in the validation set. For each example, we plot the proportion of models that get that example correct (x-axis). This is done across all validation examples, using only the Huggingface models.

easy for our models; around 60% are correctly predicted by 80% or more of the trials. However, there is a significant fraction (roughly 80 examples) on which fewer than 30% of models get the correct answer. Focusing on improving performance on these consistently challenging examples appears to be a fruitful direction for improving our systems; we leave a more detailed analysis of these examples and how they deviate from the easy ones to future work.

Model	Similarity Method	Shots	FPR	FNR
Flan-T5	TF-IDF Balanced	2	33.889%	15.707%
GPT-4	TF-IDF	5	24.444%	0.079%

Table 5: False positive rate (FPR) and false negative rate (FNR) for the best performing GPT-4 and Flan-T5 runs.

8 CONCLUSION

We have demonstrated that prompting methods using LMs can achieve competitive performance on legal entailment tasks and even outperform carefully engineered systems. Though our system performed quite well overall, several avenues remain for improvement. We attempted to supplement the provided data using similarly-structured rule sets from other domains to capture more robust open-textured terms. For each rule, we generated scenarios that were either entailed or not entailed by the rule. Including this data in training decreased model performance, likely due to substantial syntactical differences between domains, and was not included in any of the submitted systems. However, broader rule sets may aid future work on few-shot prompting for legal entailment. Instruction tuning our Flan-T5 system on legal entailments or other legal data is also a fruitful direction for future work; we explored this option briefly but found that it required too much computing resources and training time to be viable. Nevertheless, directly training the model on this data might improve performance and generalization.

An earlier version that we experimented with used a chain-of-thought prompting approach [58], which asked the model to output explanations for why it thought the answer was 'Y' or 'N'. In our experience with this approach, open-textured terms ended up being the primary problem: without further context (which may have come from additional articles that were not included in the set of provided articles A), the LM didn't know how to interpret certain terms of art or jargon that appeared in the articles or query. This is consistent with our view of interpretive reasoning, which suggests that properly interpreting open-textured legal terms often requires examples of how the term has been interpreted in the past. However, it should be noted that it is not clear whether explanations given through chain-of-thought prompting actually provide insight into how the language model came up with the answers, or whether it was a sort of post-hoc rationalization.

Although our approach did not outperform other submissions on the test set, it was a successful endeavor overall. As stated in this report's introduction, automated reasoning over rules is extremely important for the future of human interaction with AI, and competitions like COLIEE allow us to better understand the strengths and limitations of current natural language processing tools toward that goal. However, replicability is necessary to improve the broader impacts of the competition's efforts. Thus, in the future, we strongly recommend that certain measures be taken to ensure the integrity of the competition and to maximize its impact on the broader research community.

We recommend that all entrants require the release of full source code. The possibility of unintentionally selecting models, parameters, and hyperparameters that maximize performance on the test set is too great (even though the competition organizers explicitly disallowed the use of the test set for any of these). If code release is too limiting, a full description of methods, algorithms, parameters, and hyperparameters should be released before finalizing competition rankings in time for independent replication. This also allows for confirmation that the results listed in the final competition rankings were not simply due to luck—in our experience, many of the language models we used had non-deterministic output, and this required multiple runs in order to confirm that extremely good (or extremely bad) results were not simply flukes. With this spirit in mind, we publicly release our full source code for this competition.[12]

REFERENCES

[1] Amro Abbas and Stéphane Deny. 2022. Progress and Limitations of Deep Networks to Recognize Objects in Unusual Poses. arXiv:2207.08034 [cs.CV]

[2] Michał Araszkiewicz. 2021. Critical Questions to Argumentation Schemes in Statutory Interpretation. *Journal of Applied Logics - IfCoLog Journal of Logics and Their Applications* 8, 1 (2021).

[3] Kevin D. Ashley and Vern R. Walker. 2013. Toward Constructing Evidence-Based Legal Arguments Using Legal Decision Documents and Machine Learning. In *ICAIL '13: Proceedings of the Fourteenth International Conference on Artificial Intelligence and Law*. Association for Comuting Machinery, 176–180. https://doi.org/10.1145/2514601.2514622

[4] Colleen M. Berryessa, Itiel E. Dror, and Chief Justice Bridget McCormack. 2023. Prosecuting From the Bench? Examining Sources of Pro-Prosecution Bias in Judges. *Legal and Criminal Psychology* 28, 1 (2023), 1–14.

[5] Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. 2021. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258* (2021).

[6] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems* 33 (2020), 1877–1901.

[7] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language Models are Few-Shot Learners. arXiv:2005.14165 [cs.CL]

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

[9] Yanda Chen, Chen Zhao, Zhou Yu, Kathleen McKeown, and He He. 2023. On the Relation between Sensitivity and Accuracy in In-Context Learning. arXiv:2209.07661 [cs.CL]

[10] Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2017. Deep reinforcement learning from human preferences. *Advances in neural information processing systems* 30 (2017).

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

[12] Corinna Cortes and Vladimir Vapnik. 1995. Support-vector networks. *Machine learning* 20, 3 (1995), 273–297.

[13] Stefania Costantini and Gaetano Aurelio Lanzarone. 1995. Explanation-Based Interpretation of Open-Textured Concepts in Logical Models of Legislation. *Artificial Intelligence and Law* 3 (1995), 191–208. https://doi.org/10.1007/BF00872530

[14] Thomas G Dietterich. 2000. Ensemble methods in machine learning. In *Multiple Classifier Systems: First International Workshop, MCS 2000 Cagliari, Italy, June 21–23, 2000 Proceedings 1*. Springer, 1–15.

[15] Logan Fields and John Licato. 2023. Player Identification for Collectible Card Games with Dynamic Game States. In *Proceedings of The 36th International Florida Artificial Intelligence Research Society Conference (FLAIRS-34)*. AAAI.

[16] James Franklin. 2012. Discussion paper: How Much of Commonsense and Legal Reasoning is Formalizable? A Review of Conceptual Obstacles. *Law, Probability and Risk* 11, 2-3 (June-September 2012), 225–245.

[17] H.L.A. Hart. 1961. *The Concept of Law*. Clarendon Press.

[18] Tin Kam Ho. 1995. Random decision forests. In *Proceedings of 3rd international conference on document analysis and recognition*, Vol. 1. IEEE, 278–282.

[19] Jaehun Jung, Lianhui Qin, Sean Welleck, Faeze Brahman, Chandra Bhagavatula, Ronan Le Bras, and Yejin Choi. 2022. Maieutic prompting: Logically consistent reasoning with recursive explanations. *arXiv preprint arXiv:2205.11822* (2022).

[20] 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*, Yasufumi Takama, Katsutoshi Yada, Ken Satoh, and Sachiyo Arai (Eds.). Springer Nature Switzerland, Cham, 51–67.

[21] 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*, S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (Eds.), Vol. 35. Curran Associates, Inc., 22199–22213. https://proceedings.neurips.cc/paper_files/paper/2022/file/8bb0d291acd4acf06ef112099c16f326-Paper-Conference.pdf

[22] Jure Leskovec, Anand Rajaraman, and Jeff Ullman. 2014. *Mining of Massive Datasets* (3rd ed.). Stanford University.

[23] John Licato. 2021. How Should AI Interpret Rules? A Defense of Minimally Defeasible Interpretive Argumentation. *arXiv e-prints* (2021).

[24] John Licato. 2022. Automated Ethical Reasoners Must be Interpretation-Capable. In *Proceedings of the AAAI 2022 Spring Workshop on "Ethical Computing: Metrics for Measuring AI's Proficiency and Competency for Ethical Reasoning"*.

[25] John Licato. 2022. War-Gaming Needs Argument-Justified AI More Than Explainable AI. In *Proceedings of the 2022 Advances on Societal Digital Transformation (DIGITAL) Special Track on Explainable AI in Societal Games (XAISG)*.

[26] John Licato, Logan Fields, and Brayden Hollis. 2023. Code Generation for Collectible Card Games with Complex APIs. In *Proceedings of The 36th International Florida Artificial Intelligence Research Society Conference (FLAIRS-34)*. AAAI Press.

[27] John Licato and Zaid Marji. 2018. Probing Formal/Informal Misalignment with the Loophole Task. In *Proceedings of the 2018 International Conference on Robot Ethics and Standards (ICRES 2018)*.

[28] John Licato, Zaid Marji, and Sophia Abraham. 2019. Scenarios and Recommendations for Ethical Interpretive AI. In *Proceedings of the AAAI 2019 Fall Symposium on Human-Centered AI*. Arlington, VA.

[12] https://github.com/Advancing-Machine-Human-Reasoning-Lab/COLIEE-2023-Task4

[29] Valentin Liévin, Christoffer Egeberg Hother, and Ole Winther. 2023. Can Large Language Models Reason About Medical Questions? arXiv:2207.08143 [cs.CL]

[30] John Zhuang Liu and Xueyao Li. 2019. Legal Techniques for Rationalizing Biased Judicial Decisions: Evidence from Experiments with Real Judges. *Journal of Empirical Legal Studies* 16, 3 (2019), 630–670.

[31] Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2021. Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing. *ArXiv* abs/2107.13586 (2021).

[32] 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. *CoRR* abs/1907.11692 (2019). arXiv:1907.11692 http://arxiv.org/abs/1907.11692

[33] Jiaying Lu, Jiaming Shen, Bo Xiong, Wenjing Ma, Steffen Staab, and Carl Yang. 2023. HiPrompt: Few-Shot Biomedical Knowledge Fusion via Hierarchy-Oriented Prompting. *arXiv preprint arXiv:2304.05973* (2023).

[34] Qing Lyu, Shreya Havaldar, Adam Stein, Li Zhang, Delip Rao, Eric Wong, Marianna Apidianaki, and Chris Callison-Burch. 2023. Faithful chain-of-thought reasoning. *arXiv preprint arXiv:2301.13379* (2023).

[35] Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. 2022. A Survey on Bias and Fairness in Machine Learning. *Comput. Surveys* 54, 6 (2022), 1–35. https://doi.org/10.1145/3457607

[36] OpenAI. 2022. Introducing ChatGPT. https://openai.com/blog/chatgpt.

[37] OpenAI. 2023. GPT-4 Technical Report. *ArXiv* (2023). https://arxiv.org/pdf/2303.08774.pdf

[38] 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:2203.02155 [cs.CL]

[39] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* 12 (2011), 2825–2830.

[40] Ethan Perez, Douwe Kiela, and Kyunghyun Cho. 2021. True Few-Shot Learning with Language Models. In *Advances in Neural Information Processing Systems*, M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (Eds.).

[41] Henry Prakken. 2017. On the problem of making autonomous vehicles conform to traffic law. *Artificial Intelligence and Law* 25, 3 (01 Sep 2017), 341–363. https://doi.org/10.1007/s10506-017-9210-0

[42] Yingqi Qu, Yuchen Ding, Jing Liu, Kai Liu, Ruiyang Ren, Wayne Xin Zhao, Daxiang Dong, Hua Wu, and Haifeng Wang. 2021. RocketQA: An Optimized Training Approach to Dense Passage Retrieval for Open-Domain Question Answering. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, Online, 5835–5847. https://doi.org/10.18653/v1/2021.naacl-main.466

[43] Ryan Quandt and John Licato. 2020. Problems of Autonomous Agents following Informal, Open-textured Rules. In *Human-Machine Shared Contexts*, William F. Lawless, Ranjeev Mittu, and Donald A. Sofge (Eds.). Academic Press.

[44] 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*, Sebastian Padó and Ruihong Huang (Eds.). Association for Computational Linguistics, Hong Kong, 3982–3992.

[45] Ohad Rubin, Jonathan Herzig, and Jonathan Berant. 2022. Learning To Retrieve Prompts for In-Context Learning. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 2655–2671.

[46] Victor Sanh, Albert Webson, Colin Raffel, Stephen H. Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, Manan Dey, M Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal Nayak, Debajyoti Datta, Jonathan Chang, Mike Tian-Jian Jiang, Han Wang, Matteo Manica, Sheng Shen, Zheng Xin Yong, Harshit Pandey, Rachel Bawden, Thomas Wang, Trishala Neeraj, Jos Rozen, Abheesht Sharma, Andrea Santilli, Thibault Fevry, Jason Alan Fries, Ryan Teehan, Stella Biderman, Leo Gao, Tali Bers, Thomas Wolf, and Alexander M. Rush. 2021. Multitask Prompted Training Enables Zero-Shot Task Generalization. arXiv:2110.08207 [cs.LG]

[47] Giovanni Sartor, Douglas Walton, Fabrizio Macagno, and Antonino Rotolo. 2014. Argumentation Schemes for Statutory Interpretation: A Logical Analysis. In *Legal Knowledge and Information Systems. (Proceedings of JURIX 14)*. 21–28.

[48] Thibault Sellam, Dipanjan Das, and Ankur P Parikh. 2020. BLEURT: Learning Robust Metrics for Text Generation. 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, 7881–7892. https://doi.org/10.18653/v1/2020.acl-main.704

[49] Chengyu Song, Fei Cai, Mengru Wang, Jianming Zheng, and Taihua Shao. 2023. TaxonPrompt: Taxonomy-aware curriculum prompt learning for few-shot event classification. *Knowledge-Based Systems* 264 (2023), 110290. https://doi.org/10.1016/j.knosys.2023.110290

[50] Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, et al. 2022. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *arXiv preprint arXiv:2206.04615* (2022).

[51] Joost Jacob Vecht. 2020. Open texture clarified. *Inquiry* 0, 0 (2020), 1–21. https://doi.org/10.1080/0020174X.2020.1787222 arXiv:https://doi.org/10.1080/0020174X.2020.1787222

[52] Sandra Wachter, Brent Mittelstadt, and Chris Russell. 2020. Bias Preservation in Machine Learning: The Legality of Fairness Metrics Under EU Non-Discrimination Laws. *West Virginia Law Review* 123 (2020), 735–790.

[53] Douglas Walton, Fabrizio Macagno, and Giovanni Sartor. 2021. *Statutory Interpretation: Pragmatics and Argumentation*. Cambridge University Press.

[54] Douglas Walton, Giovanni Sartor, and Fabrizio Macagno. 2018. Statutory Interpretation as Argumentation. In *Handbook of Legal Reasoning and Argumentation*, Giorgio Bongiovanni, Gerald Postema, Antonino Rotolo, Giovanni Sartor, Chiara Valentini, and Douglas Walton (Eds.). Springer Netherlands, Dordrecht, 519–560. https://doi.org/10.1007/978-90-481-9452-0_18

[55] Ben Wang and Aran Komatsuzaki. 2021. GPT-J-6B: A 6 Billion Parameter Autoregressive Language Model. https://github.com/kingoflolz/mesh-transformer-jax.

[56] Shuohang Wang, Yichong Xu, Yuwei Fang, Yang Liu, Siqi Sun, Ruochen Xu, Chenguang Zhu, and Michael Zeng. 2022. Training Data is More Valuable than You Think: A Simple and Effective Method by Retrieving from Training Data. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Dublin, Ireland, 3170–3179. https://doi.org/10.18653/v1/2022.acl-long.226

[57] Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. 2021. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652* (2021).

[58] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed H. Chi, Quoc Le, and Denny Zhou. 2022. Chain of Thought Prompting Elicits Reasoning in Large Language Models. *CoRR* abs/2201.11903 (2022). arXiv:2201.11903 https://arxiv.org/abs/2201.11903

[59] 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://www.aclweb.org/anthology/2020.emnlp-demos.6

[60] BigScience Workshop, :, Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, Jonathan Tow, Alexander M. Rush, Stella Biderman, Albert Webson, Pawan Sasanka Ammanamanchi, Thomas Wang, Benoît Sagot, Niklas Muennighoff, Albert Villanova del Moral, Olatunji Ruwase, Rachel Bawden, Stas Bekman, Angelina McMillan-Major, Iz Beltagy, Huu Nguyen, Lucile Saulnier, Samson Tan, Pedro Ortiz Suarez, Victor Sanh, Hugo Laurençon, Yacine Jernite, Julien Launay, Margaret Mitchell, Colin Raffel, Aaron Gokaslan, Adi Simhi, Aitor Soroa, Alham Fikri Aji, Amit Alfassy, Anna Rogers, Ariel Kreisberg Nitzav, Canwen Xu, Chenghao Mou, Chris Emezue, Christopher Klamm, Colin Leong, Daniel van Strien, David Ifeoluwa Adelani, Dragomir Radev, Eduardo González Ponferrada, Efrat Levkovizh, Ethan Kim, Eyal Bar Natan, Francesco De Toni, Gérard Dupont, Germán Kruszewski, Giada Pistilli, Hady Elsahar, Hamza Benyamina, Hieu Tran, Ian Yu, Idris Abdulmumin, Isaac Johnson, Itziar Gonzalez-Dios, Javier de la Rosa, Jenny Chim, Jesse Dodge, Jian Zhu, Jonathan Chang, Jörg Frohberg, Joseph Tobing, Joydeep Bhattacharjee, Khalid Almubarak, Kimbo Chen, Kyle Lo, Leandro Von Werra, Leon Weber, Long Phan, Loubna Ben allal, Ludovic Tanguy, Manan Dey, Manuel Romero Muñoz, Maraim Masoud, María Grandury, Mario Šaško, Max Huang, Maximin Coavoux, Mayank Singh, Mike Tian-Jian Jiang, Minh Chien Vu, Mohammad A. Jauhar, Mustafa Ghaleb, Nishant Subramani, Nora Kassner, Nurulaqilla Khamis, Olivier Nguyen, Omar Espejel, Ona de Gibert, Paulo Villegas, Peter Henderson, Pierre Colombo, Priscilla Amuok, Quentin Lhoest, Rheza Harliman, Rishi Bommasani, Roberto Luis López, Rui Ribeiro, Salomey Osei, Sampo Pyysalo, Sebastian Nagel, Shamik Bose, Shamsuddeen Hassan Muhammad, Shanya Sharma, Shayne Longpre, Somaieh Nikpoor, Stanislav Silberberg, Suhas Pai, Sydney Zink, Tiago Timponi Torrent, Timo Schick, Tristan Thrush, Valentin Danchev, Vassilina Nikoulina, Veronika Laippala, Violette Lepercq, Vrinda Prabhu, Zaid Alyafeai, Zeerak Talat, Arun Raja, Benjamin Heinzerling, Chenglei Si, Davut Emre Taşar,

Elizabeth Salesky, Sabrina J. Mielke, Wilson Y. Lee, Abheesht Sharma, Andrea Santilli, Antoine Chaffin, Arnaud Stiegler, Debajyoti Datta, Eliza Szczechla, Gunjan Chhablani, Han Wang, Harshit Pandey, Hendrik Strobelt, Jason Alan Fries, Jos Rozen, Leo Gao, Lintang Sutawika, M Saiful Bari, Maged S. Al-shaibani, Matteo Manica, Nihal Nayak, Ryan Teehan, Samuel Albanie, Sheng Shen, Srulik Ben-David, Stephen H. Bach, Taewoon Kim, Tali Bers, Thibault Fevry, Trishala Neeraj, Urmish Thakker, Vikas Raunak, Xiangru Tang, Zheng-Xin Yong, Zhiqing Sun, Shaked Brody, Yallow Uri, Hadar Tojarieh, Adam Roberts, Hyung Won Chung, Jaesung Tae, Jason Phang, Ofir Press, Conglong Li, Deepak Narayanan, Hatim Bourfoune, Jared Casper, Jeff Rasley, Max Ryabinin, Mayank Mishra, Minjia Zhang, Mohammad Shoeybi, Myriam Peyrounette, Nicolas Patry, Nouamane Tazi, Omar Sanseviero, Patrick von Platen, Pierre Cornette, Pierre François Lavallée, Rémi Lacroix, Samyam Rajbhandari, Sanchit Gandhi, Shaden Smith, Stéphane Requena, Suraj Patil, Tim Dettmers, Ahmed Baruwa, Amanpreet Singh, Anastasia Cheveleva, Anne-Laure Ligozat, Arjun Subramonian, Aurélie Névéol, Charles Lovering, Dan Garrette, Deepak Tunuguntla, Ehud Reiter, Ekaterina Taktasheva, Ekaterina Voloshina, Eli Bogdanov, Genta Indra Winata, Hailey Schoelkopf, Jan-Christoph Kalo, Jekaterina Novikova, Jessica Zosa Forde, Jordan Clive, Jungo Kasai, Ken Kawamura, Liam Hazan, Marine Carpuat, Miruna Clinciu, Najoung Kim, Newton Cheng, Oleg Serikov, Omer Antverg, Oskar van der Wal, Rui Zhang, Ruochen Zhang, Sebastian Gehrmann, Shachar Mirkin, Shani Pais, Tatiana Shavrina, Thomas Scialom, Tian Yun, Tomasz Limisiewicz, Verena Rieser, Vitaly Protasov, Vladislav Mikhailov, Yada Pruksachatkun, Yonatan Belinkov, Zachary Bamberger, Zdeněk Kasner, Alice Rueda, Amanda Pestana, Amir Feizpour, Ammar Khan, Amy Faranak, Ana Santos, Anthony Hevia, Antigona Unldreaj, Arash Aghagol, Arezoo Abdollahi, Aycha Tammour, Azadeh HajiHosseini, Bahareh Behroozi, Benjamin Ajibade, Bharat Saxena, Carlos Muñoz Ferrandis, Danish Contractor, David Lansky, Davis David, Douwe Kiela, Duong A. Nguyen, Edward Tan, Emi Baylor, Ezinwanne Ozoani, Fatima Mirza, Frankline Ononiwu, Habib Rezanejad, Hessie Jones, Indrani Bhattacharya, Irene Solaiman, Irina Sedenko, Isar Nejadgholi, Jesse Passmore, Josh Seltzer, Julio Bonis Sanz, Livia Dutra, Mairon Samagaio, Maraim Elbadri, Margot Mieskes, Marissa Gerchick, Martha Akinlolu, Michael McKenna, Mike Qiu, Muhammed Ghauri, Mykola Burynok, Nafis Abrar, Nazneen Rajani, Nour Elkott, Nour Fahmy, Olanrewaju Samuel, Ran An, Rasmus Kromann, Ryan Hao, Samira Alizadeh, Sarmad Shubber, Silas Wang, Sourav Roy, Sylvain Viguier, Thanh Le, Tobi Oyebade, Trieu Le, Yoyo Yang, Zach Nguyen, Abhinav Ramesh Kashyap, Alfredo Palasciano, Alison Callahan, Anima Shukla, Antonio Miranda-Escalada, Ayush Singh, Benjamin Beilharz, Bo Wang, Caio Brito, Chenxi Zhou, Chirag Jain, Chuxin Xu, Clémentine Fourrier, Daniel León Periñán, Daniel Molano, Dian Yu, Enrique Manjavacas, Fabio Barth, Florian Fuhrimann, Gabriel Altay, Giyaseddin Bayrak, Gully Burns, Helena U. Vrabec, Imane Bello, Ishani Dash, Jihyun Kang, John Giorgi, Jonas Golde, Jose David Posada, Karthik Rangasai Sivaraman, Lokesh Bulchandani, Lu Liu, Luisa Shinzato, Madeleine Hahn de Bykhovetz, Maiko Takeuchi, Marc Pàmies, Maria A Castillo, Marianna Nezhurina, Mario Sänger, Matthias Samwald, Michael Cullan, Michael Weinberg, Michiel De Wolf, Mina Mihaljcic, Minna Liu, Moritz Freidank, Myungsun Kang, Natasha Seelam, Nathan Dahlberg, Nicholas Michio Broad, Nikolaus Muellner, Pascale Fung, Patrick Haller, Ramya Chandrasekhar, Renata Eisenberg, Robert Martin, Rodrigo Canalli, Rosaline Su, Ruisi Su, Samuel Cahyawijaya, Samuele Garda, Shlok S Deshmukh, Shubhanshu Mishra, Sid Kiblawi, Simon Ott, Sinee Sang-aroonsiri, Srishti Kumar, Stefan Schweter, Sushil Bharati, Tanmay Laud, Théo Gigant, Tomoya Kainuma, Wojciech Kusa, Yanis Labrak, Yash Shailesh Bajaj, Yash Venkatraman, Yifan Xu, Yingxin Xu, Yu Xu, Zhe Tan, Zhongli Xie, Zifan Ye, Mathilde Bras, Younes Belkada, and Thomas Wolf. 2023. BLOOM: A 176B-Parameter Open-Access Multilingual Language Model. arXiv:2211.05100 [cs.CL]

[61] Xi Ye and Greg Durrett. 2023. Explanation Selection Using Unlabeled Data for In-Context Learning. *arXiv preprint arXiv:2302.04813* (2023).

[62] Douglas Yeung, Inez Khan, Nidhi Kalra, and Osonde A. Osoba. 2021. *Identifying Systemic Bias in the Acquisition of Machine Learning Decision Aids for Law Enforcement Applications*. RAND Corporation, Santa Monica, CA. https://doi.org/10.7249/PEA862-1

[63] Fangyi Yu, Lee Quartey, and Frank Schilder. 2022. Legal Prompting: Teaching a Language Model to Think Like a Lawyer. *arXiv preprint arXiv:2212.01326* (2022).

[64] Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah Goodman. 2022. Star: Bootstrapping reasoning with reasoning. *Advances in Neural Information Processing Systems* 35 (2022), 15476–15488.

[65] Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. 2022. OPT: Open Pre-trained Transformer Language Models. arXiv:2205.01068 [cs.CL]

[66] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. BERTScore: Evaluating Text Generation with BERT. In *International Conference on Learning Representations*.

[67] Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate Before Use: Improving Few-shot Performance of Language Models. In *Proceedings of the 38th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 139)*, Marina Meila and Tong Zhang (Eds.). PMLR, 12697–12706. https://proceedings.mlr.press/v139/zhao21c.html

[68] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. 2022. Learning to Prompt for Vision-Language Models. arXiv:2109.01134 [cs.CV]

Author Index