Retrieving Legal Cases from a Large-scale Candidate Corpus

Yixiao Ma, Yunqiu Shao, Bulou Liu, Yiqun Liu*, Min Zhang, Shaoping Ma
yiqunliu@tsinghua.edu.cn
Department of Computer Science and Technology, Institute for Artificial Intelligence,
Beijing National Research Center for Information Science and Technology,
Tsinghua University, Beijing 100084, China

ABSTRACT

This paper presents approaches employed in COLIEE 2021 Task 1, a legal case retrieval task that aims to retrieve all noticed cases given a large-scale candidate case corpus. Of all two methods, the first method is a traditional language model for information retrieval (IR) and the second is a neural-based method named refined BERT-PLI. In addition, we design a filter to remove unreasonable candidates from the result list. The official Task 1 results show that our Run 1 has the best performance of all 15 runs and is significantly better than the second-place method. Besides, all of our three runs have a top-5 performance.

CCS CONCEPTS

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

KEYWORDS

legal case retrieval, language model, neural-based method

ACM Reference Format:
Yixiao Ma, Yunqiu Shao, Bulou Liu, Yiqun Liu*, Min Zhang, Shaoping Ma. 2021. Retrieving Legal Cases from a Large-scale Candidate Corpus. In *Proceedings of COLIEE 2021 workshop: Competition on Legal Information Extraction/Entailment (COLIEE 2021).* ACM, New York, NY, USA, 5 pages.

1 INTRODUCTION

Legal case retrieval is of vital significance to the legal domain. Under different law systems, a relevant case can be directly or indirectly involved in the final decision as a crucial reference for judges. Lawyers or judges used to manually search for previous cases as supporting materials of the case in trial. However, as the number of precedents continues to grow, it is time-consuming to manually collect relevant cases. Hopefully, with the development of information retrieval (IR), adopting IR methods to automatically retrieve legal information in need, especially in the legal case retrieval domain, has currently received increasing attention. An efficient method proposed for the relevant case retrieval task can alleviate the heavy material preparation work. Therefore, competitions and evaluations [1, 2] are held to promote such methods used in AI & Law.

As one of the popular competitions, the Competition on Legal Information Extraction/Entailment (COLIEE) [4] is an evaluation competition held annually to develop IR and document entailment methods in the legal domain since 2014. In the latest COLIEE 2021, there are in total five tasks, among which Task 1 is a legal case retrieval task, Task 2 is a legal case entailment task, Task 3 is a statute law retrieval task, Task 4 is a legal textual entailment task, and Task 5 is a legal question answering (QA) task. Both Task 1 and Task 2 are based on a database of predominantly Federal Court of Canada case laws provided by Compass Law, while Task 3 and Task 4 are based on Japanese legal bar exams. This year, our team, Tsinghua Legal Information Retrieval (TLIR) participates in task1. Notably, compared with previous COLIEE competitions that each query of Task 1 only contains 200 candidate cases, all queries of Task 1 in COLIEE 2021 share a 4415-case candidate pool, which significantly increases the task difficulty. In total, COLIEE 2021 Task 1 received 15 submissions from seven teams.

In this paper, we mainly introduce our approaches corresponding to three runs of Task 1. For the approach corresponding to Run 1, we first adopt a series of data mining methods to clean the raw dataset and collect a word-level corpus. Then the corpus is used to train a language model for IR. Finally, all top-scoring outputs are passed through a filter to get our final results. Run 2 and Run 3 are all results of a refined BERT-PLI [11]. For the approach corresponding to Run 2 and Run 3, we first use the first approach to sample top-30 relevant cases from the candidate pool. Then, unlike the BERT-PLI in COLIEE 2020, we only adopt part of paragraphs of a query case for training leveraging both the final performance and the expansion of the Task 1 dataset this year. As a result, the placements of our three runs in Task 1 are: **1st place** (Run 1), 3rd place (Run 3), and 5th place (Run 2). The evaluation F1 score of our Run 1 is more than twice the F1 score of the 2nd place run.

2 TASK OVERVIEW

2.1 Task 1 Description

Task 1 is a legal case retrieval task related to a database of predominantly Federal Court of Canada case laws provided by Compass Law. Given a query case Q, the target is to extract all supporting cases $S = \{S_1, S_2, ...S_n\}$ from the entire case law corpus. The supporting case, which is also called 'noticed cases', denotes precedents that can support the judgment for the query case Q. There are in total 650 query cases with noticed case labels as the training set and 250 query cases without noticed case labels as the test set.

2.2 Data Corpus

The dataset of Task 1 is drawn from an existing collection of predominantly Federal Court of Canada case law. Statistics of the dataset are shown in Table 1. All query cases share a large-scale candidate

*Corresponding author

COLIEE 2021, June 21, 2021, Online

Table 1: Dataset statistics of COLIEE Task 1.

Statistic	Training	Testing
# queries	650	250
# candidate cases	4415	4415
# noticed cases per query	5.09 (0.12%)	3.60 (0.08%)

pool with 4415 case documents in total, and even the queries themselves are sampled from such a pool. By comparison, each query in previous COLIEE Task 1 has a similar number of noticed cases to retrieve from a much smaller independent candidate pool (e.g., 200 candidates per query in COLIEE 2020). Therefore, retrieving noticed cases this year is more challenging.

2.3 Evaluation Metrics

The evaluation metrics of Task 1 are precision, recall and F1 score. Definition of these measures are as follows:

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

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

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

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

3 METHODS

3.1 Run 1: Language Models for IR

Traditional retrieval models such as BM25 [8], TF-IDF [9], and Language Model for Information Retrieval (LMIR) [6] rank candidates by statistical probabilistic framework based on the bag-of-words representations. Shao et al. [11] demonstrate that traditional retrieval methods has competitive results in legal case retrieval. Therefore, in COLIEE 2021, we choose traditional retrieval models as our first run for Task 1.

The original case document mainly has two types of structures: headings and paragraphs. Headings include title, court, date, summary, and other information. Paragraphs with a number tag at the beginning are the main content of the case. Considering that most information in headings is irrelevant to the retrieval task, we only use paragraphs for both of our approaches in this competition.

The first step is data pre-processing. Since the data is drawn from Canada law, some of the case documents contain French text. After analysis, most of these French parts are the translation of the English statement in the same document. Therefore, removing such French text from their belonging documents does not influence the overall information obtained in documents. In this paper, we adopt Langdetect [12] to remove all French paragraphs. Then, we convert all English letters to lowercase. Finally, we apply Nltk [5] to split words, remove stopwords and punctuation, and stem. After pre-processing, we get all cases in the form of tokens. All tokens are gathered together as the training corpus C for traditional retrieval models.

In this competition, we adopt LMIR as the Run 1 model, where computing the relevance between candidates and queries is considered a query generation process. In other words, given a candidate case c, the probability of generating the correct query q $P(q \mid c)$ is denoted as:

$$P(q \mid c) \propto \prod_{t \in q} P(t \mid c) \tag{4}$$

where t is the token in a query generated from data pre-processing step and $P(t \mid c)$ represents the probability of generating token t given c. For LMI, there are multiple methods to estimate $P(t \mid c)$. In this paper, we choose the linear interpolation language model and $P(t \mid c)$ is defined as:

$$P(t \mid c) = \lambda P_{ml}(t \mid M_c) + (1 - \lambda) P_C(t \mid M_C) \tag{5}$$

where $P_{ml}(t \mid M_c)$ denotes token probability in case document c, $P_C(t \mid M_C)$ denotes token probability in the whole training corpus C, and λ is a smoothing parameter ranging from 0 to 1.

When computing the relevance score between queries and candidate cases, instead of taking all tokens of a query as the input q of LMIR, we only identify tokens from paragraphs that are more likely to cite noticed cases. Specifically, according to the data format, sentences with a placeholder such as 'FRAGMANT_SUPPRESSED' or 'REFERENCE_SUPPRESSED' etc. are citations or references from other noticed cases. These sentences are directly relevant to a noticed case. Furthermore, considering the context of the reference sentence may also have a connection with noticed cases, we take all tokens from paragraphs to which a placeholder belongs as our input q of LMIR.

Although our LMIR model is already able to output a ranking list given query input q, there are still regulations that can filter some unreasonable candidates. In this competition, we design a two-step candidate filter. First, a query case can only cite cases judged before the query case itself. Therefore, we extract dates from cases containing time information. Besides the trial date, some case documents also record other dates such as its previous judgment date or date of the case. To avoid mistakenly filtering noticed cases, we define our first regulation as:

$$Rank_1 = \{c \mid c \in Rank_0 \wedge max(dates(c)) \leq min(dates(q))\} \tag{6}$$

where $dates(d)$ is the set of dates appeared in document d and $Rank_0$ is the original output of LMIR.

In addition, we also find 222 document pairs describing the same case but have different document ids. For example, document '008447.txt' and document '089987.txt'. As one case never cites itself as a noticed case, we remove such documents from $Rank_1$ to get our final rank list $Rank_2$. The final submission consists of top-K cases from $Rank_2$ for each query. K is a hyperparameter.

3.2 Run 2 & Run 3: Refined BERT-PLI

Shao et al. [11] propose BERT-PLI to tackle challenges in legal case retrieval scene that case documents have extended length and complex structure. This method divides a document into paragraph

Table 2: Recall rate of the top-30, top-50, and top-100 scored candidate cases using different traditional retrieval models. Avg. Rank is the average rankings of all noticed cases.

Method	30	50	100	Avg. Rank
TF-IDF	0.333	0.420	0.560	332.0
BM25	0.373	0.463	0.583	384.8
LMIR	**0.437**	**0.537**	**0.660**	**243.1**

level and computes interactions between paragraphs using BERT [3]. Compared with other neural models, BERT-PLI can take long-text representation as an input without cutting off long documents in the middle. Previous COLIEE Task 1 results [10] illustrate that BERT-PLI has competitive performance. Thus, Run 2 and Run 3 are mainly based on BERT-PLI but have some revisions according to the feature of COLIEE 2021 dataset.

The overall model structure is shown in Figure 1. In general, the model consists of three stages. In Stage 1, we first sample top-N candidates from the whole candidate pool by traditional retrieval models. In order to choose a traditional retrieval model with a better recall, we first conduct a pre-experiment between TF-IDF [9], BM25 [8], and LMIR[6]. We compute the overall recall rate and average rankings of all noticed cases in the training set. As shown in Table 2, LMIR has both the highest recall and lowest average rankings. Therefore, we adopt LMIR to sample candidates. In practice, we sample all top-30 candidates and other noticed cases ranking more than 30 to be the training set for BERT-PLI.

In Stage 2, we fine-tune the BERT [3] with a case-entailment dataset in COLIEE 2019 [7] Task2 which aims to identify paragraphs entailing the decision paragraph in a document. The fine-tuning process is handled on a next sentence prediction task. Composed of a decision paragraph and a candidate paragraph separated, the input sentence pair is separated by a [SEP] token and appended by [CLS] token. Vectors from the final hidden layer are fed into a classification layer to get the final prediction.

In Stage 3, the original idea in Shao et al. [11] computes interactions between all query paragraphs and all top-30 candidate case paragraphs. This strategy may be practical for a test set within 200 candidates, but this year the size of the candidate pool is over 20 times larger than before. As shown in Table 2, the top-30 recall is only less than 50 percent which is far below our expectation. Therefore, in this competition, we take top-200 candidates into consideration. Following the idea in Run 1, we only compute the interactions between query paragraphs with a citation token and all top-30 candidate paragraphs (top-200 candidate paragraphs for test set). We apply BERT to infer semantic connections between these paragraph pairs and generate a paragraph-level interaction map. Then, the map is fed into a max-pooling layer to generate the most representative value for each query paragraph. Finally, an RNN combined with an attention layer is utilized to encode a paragraph-level sequential feature p_{qk} into a document-level feature d_{qk}, where q and k represent the query and candidate respectively. Finally, d_{qk} is passed through a fully connected layer to output a two-dimensional prediction vector l.

Table 3: Results of LMIR ($\lambda = 0.95$) on the training set with different hyperparameter K.

K	Precision	Recall	F1
5	**0.1809**	0.1776	0.1792
6	0.1718	0.2024	**0.1858**
7	0.1596	0.2193	0.1847
8	0.1508	0.2368	0.1842
9	0.1429	0.2525	0.1825
10	0.1349	**0.2649**	0.1788

Similar to Run 1, we also apply a filter to our model. The first two steps are exactly the same as Run 1 filter. An additional filter step is adopted due to the fact that our model tends to overestimate the relevance between queries and all candidates. Specifically, suppose l is the prediction vector of a single candidate and L is a list that contains all l predicted as noticed. Then, the whole list L is fed into a filter function f to get the final prediction y. The filter function f is defined as:

$$f(L) = \begin{cases} \arg\max_{l} \ Softmax(l)[1] & |L| = 0 \\ L & 0 < |L| \leq 10 \\ \{l \mid l \in L \ \wedge \ Softmax(l)[1] > T\} & |L| > 10 \end{cases} \quad (7)$$

where $Softmax(l)$ is a two-dimensional vector generated by a Softmax layer and $Softmax(l)[1]$ denotes the value on the second dimension which is the probability of a positive (noticed) prediction. T is the threshold to control the number of overall cases predicted as noticed.

4 EXPERIMENTS

For Run 1, we set the hyperparameter λ to be 0.95 based on a comparison between the performances with different λ values. $\lambda = 0.95$ achieving the best performance illustrates that when the query has a short text length (only paragraphs with a token in Run 1) and the candidate pool has a relatively large size, $P_{ml}(t \mid M_c)$ plays a more important role in determining the relevance between queries and candidates. In other words, term frequencies in the query are decisive to retrieve noticed cases in Task 1 this year. Another key hyperparameter that needs to be determined before further experiments is K, the number of retrieved cases per query. According to the evaluation results on the training set shown in 3, we set K to be 6. For Run 2 and Run 3, we set the maximum input query paragraph number to be 30 and the maximum input candidate paragraph number to be 40. The rest of some important hyperparameters are as follows: learning rate = $5e-4$ for both runs, weight decay = $1e-6$ for Run 2 and 0 for Run 3, threshold T = 0.83 for Run 2 and 0.56 for Run 3. On the validation set, Run 2 has a more balanced performance (precision = 0.3171, recall = 0.3212, F1 = 0.3191), while Run 3 mainly focuses on precision (precision = 0.3785, recall = 0.1223, F1 = 0.1848). We train a model like Run 3 because the training set (top-30 candidate + other noticed case) has a higher ratio of noticed cases to all cases than the test set (0.08%). Therefore, controlling recall and improving precision on the validation set can have a better F1 score on the test set.

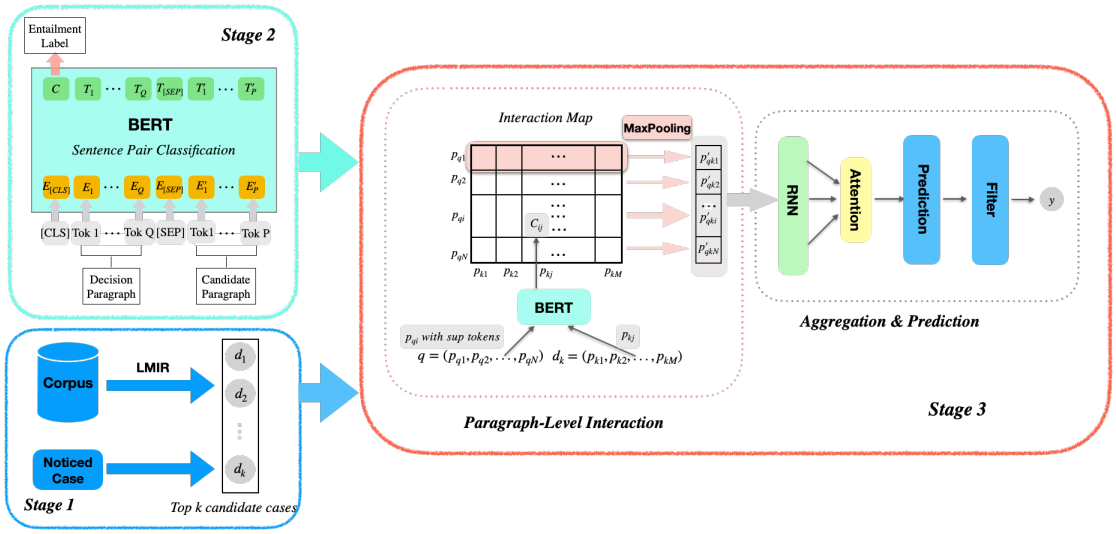

Figure 1: The overall structure of refined BERT-PLI.

Table 4: Final top-5 results of Task 1 on the test set.

Team	Precision	Recall	F1 (official)	Rank
TLIR (Run1)	0.1533	0.2556	**0.1917**	1
NM	-	-	0.0937	2
TLIR (Run3)	0.0350	0.0656	0.0456	3
DSSIR	-	-	0.0411	4
TLIR (Run2)	0.0259	0.0456	0.0330	5

The final top-5 results of COLIEE Task 1 are illustrated in Table 4. The organizers only publish F1 scores, and we further evaluated our three runs by precision and recall after the test set labels are released. Of all 15 runs, our Run 1 has the best F1 score and significantly outperforms other runs. Besides, Run 3 has the third placement and Run 2 has the fifth placement. From the results above, we can conclude that while in previous COLIEE Task1, neural methods have a slightly better performance than traditional retrieval models [7], this year traditional retrieval models (rank 1, 2) outperforms neural methods. Therefore, traditional retrieval models are robust and still competitive in the legal search domain, especially when the candidate pool size is relatively large (e.g. 4415).

5 CONCLUSION AND DISCUSSION

In this paper, we presented two retrieval methods for the legal case retrieval task in COLIEE 2021. For the first approach, we utilize LMIR and design a filter to remove unreasonable candidates from the result list. For the second approach, we refine a competitive neural method BERT-PLI and also design a filter to control positive predictions. Competition results show that Run 1 has the best performance of all runs and is significantly better than the second-place method. In addition, all of our three runs have a top-5 performance.

On the other hand, as the size of the candidate case pool per query is changed from 200 to 4415 this year, Task 1 in COLIEE 2021 becomes more challenging than previous legal case retrieval tasks in COLIEE. Consequently, the overall performances of Task 1 this year decrease to a large extent. In addition to the pool size, there are other reasons for this decline: First, as mentioned in Section 3.1, there exist some documents describing the same documents. If such document pairs are query documents, their noticed cases can even be totally different. For example, documents '067501.txt' and '030050.txt' are about the same case, but the noticed cases of '067501.txt' are '038025.txt' and '072553.txt' while the noticed case of '030050.txt' is '028189.txt'.

The second possible explanation is that with the growth of candidate case number, the ratio of noticed cases to top-k scored candidate cases decreases if we still use semantic-based or term-level methods to retrieve legal cases. In other words, existing methods do not effectively support large-scale legal case retrieval. Therefore, future works need to explore method which can utilize more than semantic or term-level information in legal documents.

REFERENCES

[1] Jason R Baron, David D Lewis, and Douglas W Oard. 2006. TREC 2006 Legal Track Overview.. In *TREC*. Citeseer.
[2] Paheli Bhattacharya, Kripabandhu Ghosh, Saptarshi Ghosh, Arindam Pal, Parth Mehta, Arnab Bhattacharya, and Prasenjit Majumder. 2019. Overview of the FIRE 2019 AILA Track: Artificial Intelligence for Legal Assistance.. In *FIRE (Working Notes)*. 1–12.
[3] 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).
[4] Yoshinobu Kano, Mi-Young Kim, Masaharu Yoshioka, Yao Lu, Juliano Rabelo, Naoki Kiyota, Randy Goebel, and Ken Satoh. 2018. Coliee-2018: Evaluation of the competition on legal information extraction and entailment. In *JSAI International Symposium on Artificial Intelligence*. Springer, 177–192.
[5] Edward Loper and Steven Bird. 2002. Nltk: The natural language toolkit. *arXiv preprint cs/0205028* (2002).
[6] Jay M Ponte and W Bruce Croft. 1998. A language modeling approach to information retrieval. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*. 275–281.
[7] Juliano Rabelo, Mi-Young Kim, Randy Goebel, Masaharu Yoshioka, Yoshinobu Kano, and Ken Satoh. 2019. A Summary of the COLIEE 2019 Competition. In *JSAI International Symposium on Artificial Intelligence*. Springer, 34–49.

[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] Gerard Salton, Anita Wong, and Chung-Shu Yang. 1975. A vector space model for automatic indexing. *Commun. ACM* 18, 11 (1975), 613–620.
[10] Yunqiu Shao, Bulou Liu, Jiaxin Mao, Yiqun Liu, Min Zhang, and Shaoping Ma. 2020. THUIR@ COLIEE-2020: Leveraging Semantic Understanding and Exact Matching for Legal Case Retrieval and Entailment. *arXiv preprint arXiv:2012.13102* (2020).
[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 *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20.* 3501–3507.
[12] Nakatani Shuyo. 2010. Language detection library for java. *Retrieved Jul* 7 (2010), 2016.