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.

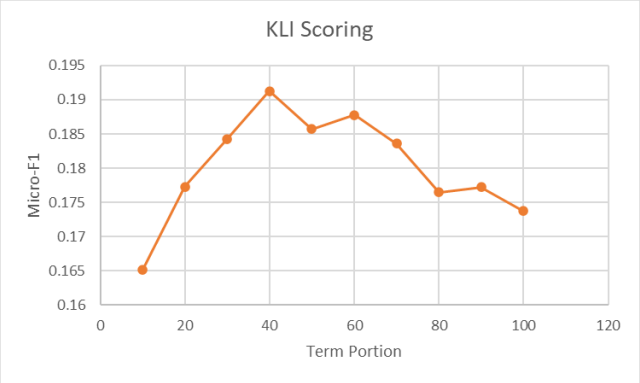

Figure 1: Results with BM25 opt for the validation set varying term portion in KLI scoring of terms.

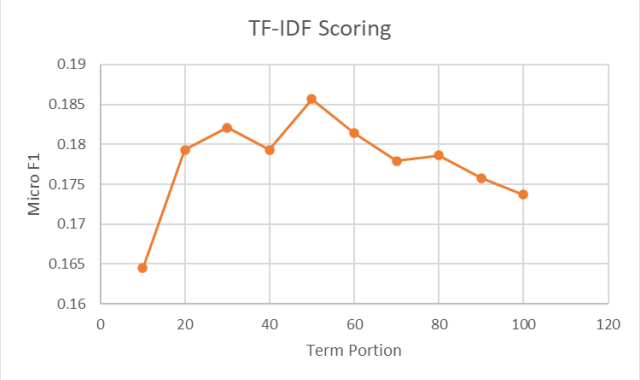

Figure 2: Results with BM25 opt for the validation set varying term portion in TF-IDF scoring of terms.

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]

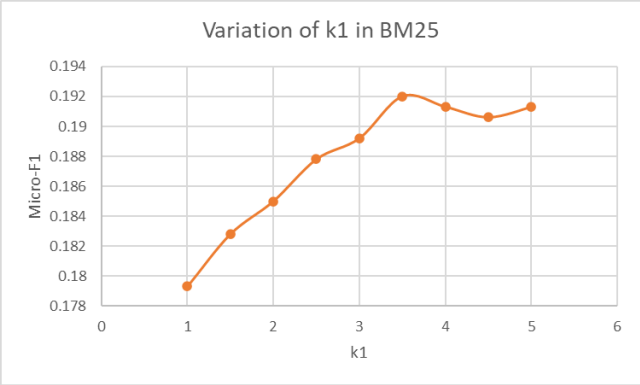

Figure 3: Results with BM25 for the validation set varying parameter k1 keeping b = 1.

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.