Information Retrieval for Legal Documents

By Michel Custeau

Supervised by Diana Inkpen

Introduction:

In this report, I will discuss the experimentation and results from implementing an Information Retreival (IR) system, using the BM25+ ranking function and the Bert language model, on two collections of legal documents from the COLIEE-2021 task 1 dataset and the FIRE-2017-IRLeD task 2 dataset. These results will be analyzed by comparing the different performance metrics, with F1 score as our main one, on the set of ranked documents based on relevancy towards a query.

Task description:

For both datasets, the queries were composed of legal documents. Any references or citations towards other legal documents from these queries had the source redacted. However, all documents that each query referenced were present in the dataset of candidate legal documents. My goal was to find the source of those references and citations by ranking the candidate documents in terms of relevancy to the query, with the goal of having the most relevant documents as the most likely to be the source of the references and citations. The image below shows an example of this, where the highlighted parts are the removed sources.

IV.
Analysis
A.
Whether the Immigration Officer applied the correct test in determining the best interests of the children and, if so, whether his determination was reasonable
[9]
The Applicant maintains that the correct approach to conducting an analysis of the best interests of the child is found in
REFERENCE_SUPPRESSED:
63 When assessing a child's best interests an Officer must establish
first
what is in the child's best interest,
second
the degree to which the child's interests are compromised by one potential decision over another, and
then finally, in light of the foregoing assessment determine the weight that this factor should play in the ultimate balancing of positive and negative factors assessed in the application. [Emphasis original]
[10]
In
REFERENCE_SUPPRESSED, Justice Mosley observed that "the
CITATION_SUPPRESSED
formula provides a useful guideline for officers to follow where it may be helpful in assessing a child's best interests but it is not mandated by the governing authorities from the Supreme Court and the Federal Court of Appeal." Ultimately, the correct legal test is whether the Immigration officer was "alert, alive and sensitive" to the best interests of the child:
REFERENCE_SUPPRESSED.

Data Cleaning:

The COLIEE-2021 task 1 dataset was comprised of 250 queries and 900 documents for the test set, and 650 queries and 2655 documents for the training set, all them Canadian legal documents. The FIRE-2017-IRLeD task 2 dataset was comprised of 200 queries and 2000 documents, all legal documents for Indian court cases. Each of the queries and documents were cleaned by removing stop words and unnecessary punctuation, with words lemmatised, lower cased, and tokenised into a list.

dataset	COLIEE-2021 train set	COLIEE-2021 test set	FIRE-2017-IRLeD
Number of query cases	650	250	200
Number of documents	2655	900	2000

When referring to the results from the COLIEE-2021 dataset, we will strictly be looking at the results achieved using the test set, and not the training set.

Evaluation:

I will now go into the methods used to perform information retrieval. For all methods, I will use the BM25 ranking function. There are different iterations of BM25, such as Okapi BM25, BM25L and BM25+. For this project, I decided to use BM25+ as it is designed to be used for long documents by adding an extra parameter that stops long documents from being unfairly scored in comparison to shorter documents. Below is the equation for BM25+.

$$\text{score}(D,Q) = \sum_{i=1}^{n} \text{IDF}(q_i) \cdot \left[\frac{f(q_i, D) \cdot (k_1 + 1)}{f(q_i, D) + k_1 \cdot \left(1 - b + b \cdot \frac{|D|}{\text{avgdl}}\right)} + \delta \right]$$

Method 1: Scoring Entire Document with BM25+:

The first method we will analyze is scoring entire documents with BM25+. This means that, after the steps of data cleaning, the entire word count for both the query and candidate document is put into the scoring function to find its relevancy towards the query. The

advantage to this method is that its very simple to put into code, however, a lot of words that we could consider useless are used for scoring relevancy, hence creating a lot of noise. Here are the results we get by retrieving the top 10 documents per query:

	COLIEE-2021	FIRE-2017-IRLeD
Recall	0.3889	0.3290
Precision	0.1400	0.1645
F1	0.2059	0.2193

As we can see, both datasets scored similar on the F1 metric. We will use these results as baseline for the next methods.

Method 2: Targeting words around removed reference locations with BM25+:
The second method is to target the words around any location that contains the indications of a removed reference or citation source, such as 'CITATION_SUPPRESSED' or 'REFERENCE_SUPPRESSED'. The logic behind this method is that words in these locations should be similar to the content of the referenced document, hence reducing the word count for the query significantly. The number of words to use for each location is a parameter one can edit. I found that 128 tokens around each removed reference location seemed to work well for both datasets.

By implementing this, we obtain the matrix shown below, where each row is a fragment of words around the targeted removed reference location in the text, and each column the documents in the corpus of candidates. Each entry in the matrix is the score of the query fragment in relation to the document.

$$\left[\begin{array}{c|ccccc} & d_1 & d_2 & d_3 & \cdots & d_n \\ q_1 & v_{1\,1} & v_{1\,2} & v_{1\,3} & \cdots & v_{1\,n} \\ q_2 & v_{2\,1} & v_{2\,2} & v_{2\,3} & \cdots & v_{2\,n} \\ \vdots & \vdots & \vdots & \vdots & \cdots & \vdots \\ q_n & v_{n\,1} & v_{n\,2} & v_{n\,3} & \cdots & v_{n\,n} \end{array}\right]$$

This matrix must then be turned into a vector, where each entry of the vector is the relevancy score of the entire query in relation to each document. I came up with two function to do this:

1) $$Sum(d_j) = \sum_i v_{ij}$$

2) $$Max(d_j) = Max(v_{1j}, \ldots, v_{nj})$$

These two functions, which are applied at each column, will turn the column into one single score. For the first equation, the sum of the column for a document is taken to be used as the score, while the second equation uses the max value of the column as the score. Hence, when applying either these equations on each column of the matrix, we get one single vector with its elements as the relevance score for each document. This opens the question on which equation is better to use. This depends on the dataset. I think the first equation is better to use if most queries reference more than one document, as the score for each fragment of the query in relation to the document is taken into account, while the second equation probably works best if most queries only reference one document. For the remainder of this paper, I will refer to the scoring function 1) as the Sum function and 2) as the Max function. Here are the results using top 10 documents retrieved per query:

dataset	COLIEE-2021		FIRE-2017-IRLeD	
Scoring function	Sum	Max	Sum	Max
Recall	0.4222	0.3967	0.411	0.4790
Precision	0.1520	0.1424	0.2055	0.2395
F1	0.2235	0.2102	0.2740	0.3196

As we can see, both datasets achieved a higher f1 score by using this method rather than simply taking entire documents, which we saw previously in method 1. In the case of the COLIEE-2021 dataset, the Sum function performed slightly better than the Max function, while FIRE-2017-IRLeD had significantly better results with the Max function than the Sum function.

Method 3: Using the summary and introductions with BM25+:

Another way to rank legal documents could be by taking advantage of the way they are structured. In the case of COLIEE-2021 specifically, we can see that a lot of the documents have a summary or an introduction. By targeting those paragraphs, one can find the main subjects of the legal documents with minimal word count. Hence, for this method, I truncated the queries and documents in the corpus to 512 tokens from their summary or introduction (my program looks for a summary first, and if it doesn't have one, looks for an introduction, and then takes 512 tokens from either these sections). Here are the results achieved for the COLIEE-2021 dataset with top 10:

	COLIEE-2021
Recall	0.4187
Precision	0.1508
F1	0.2218

We can see that we are getting better results than method 1 and also better results than using the max function in method 2. However, using the Sum function with method 2 remains the best performing method. I didn't use this method on the legal cases from the FIRE-2017-IRLeD dataset as I didn't find any meaningful structure from them. Unlike the cases from COLIEE-2021, the cases from FIRE-2017-IRLeD didn't seem to be divided into clear sections with a summary or introduction, making it hard to pinpoint any singular meaningful paragraph from the text.

Method 4: Using Bert along side BM25+:

For this next method, I decided to use a Bert model for re-ranking. This means that I used BM25+, specifically method 2 with Sum function as it gives the best results, and then re-rank the top 300 documents using a Bert model.

At its core, Bert is a contextual neural language model that is designed to pre-train unlabeled text into deep bidirectional representations. By using a model that has already been pre-trained on english text, we can generate document embeddings. A document embedding is a vector of real numbers which represent a document in a vector space. Embeddings close to each other inside the vector space should, ideally, be similar to each other. Hence, by getting

the embedding for our query and documents, and then computing the cosine similarity between the query embedding and each document embedding, we can find which document embedding is the closest to the query embedding, which is the method I used for this part of the project.

As for the Bert model, I used LEGAL-BERT provided by Hugging Face, which has been pre-trained on 12gb of english legal text. Inspired by method 2, I created embeddings for each fragment of the query that contains words close to removed reference locations. As for the candidate documents, since Bert can't handle heavy word counts well, I divided the documents into groups of 512 tokens, and then embedded each of these paragraphs. To find the score ranking, I then had to find, for each document, which was the closest paragraph embedding to one of the query fragments, and then used the cosine similarity as the ranking score. Here are the results obtained from the top 10 documents:

dataset	COLIEE-2021	FIRE-2017-IRLeD
Recall	0.2078	0.1300
Precision	0.0748	0.0650
F1	0.1100	0.0867

As we can see, those results perform significantly less than our previous methods using solely bm25+.

Comparison to competitors:

I will now use this part of the paper to compare my results to the winners of the COLIEE-2021 competition. By looking at their paper "Retrieving Legal Cases from a Large-scale Candidate Corpus", they used three runs, where Run 1 is where they utilized a Language Model for Information Retrieval (LMIR), and Run 2 and Run 3 where they used the Bert model Bert-PLI for re-ranking. It is also worth noting that they targeted words around removed reference locations, similar to my Method 2. They set the number of cases retrieved per query as 6, and got these results:

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 group also mentioned something interesting at the end of their paper that could explain why my BM25+ based methods worked better than my method 4 which used Bert. As stated, “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.” This means that the contestants also got better results by using traditional retrieval methods, in their cases a LMIR, rather than using a Bert model.

I will now compare my results. It is important to note that for my results shown previously in this paper, I had the queries and candidate documents from the test set in two separate independent pools. However, for the COLIEE-2021 competition, it appears that the queries and documents, both from the train and test set, were mixed into one single pool, meaning that queries could also accidentally retrieve other queries instead of candidate documents, or also accidentally retrieve documents from the training set instead of the test set. Hence, to make my results fair, I mixed all queries and documents, both from the test and train set, into one single pool to replicate what the competition would normally look like. I did not make any modifications to my code except removing the top ranked document retrieved for each query, as it would naturally result in the query retrieving itself in top place. I also set my number of cases retrieved the same as the contestants, which in their case was 6. Using method 2 with the Sum function, this gave the results for top 6:

dataset	COLIEE-2021
Recall	0.2046
Precision	0.1226
F1	0.1532

As we can see, while my results did not manage to outperform the winners, as they got an F1 score of 0.1917 using a LMIR, while I got an F1 score of 0.1532 using BM25+, they came in relatively close with a 0.385 difference.

Improvements for future work:

For future improvements, I noticed that certain queries and documents in the COLIEE-2021 dataset contained french words, therefor, either translating or removing those words could potentially reduce noise and improve our results. I also think that taking advantage of the training set by using it to train our Bert model could also give positive improvements. I also want to further study Bert models and better understand some of their parameters, and also try the Bert-PLI model, which, as the contestants mentioned in their paper, was successful for previous competition, but not this one. I think also that the LMIR that the contestants used could be an interesting traditional retrieval model to compare for future competitions alongside BM25 and TI-IDF.

Conclusion:

I believe that this honours project was very valuable as it educated me on the different processes and ways to analyze documents and queries for legal information retrieval. It has also taught me that, while neural network based methods are important for information retrieval, they should not always be prioritized over traditional retrieval models, which as we saw, sometimes can perform better.

References:

Mandal, A., Ghosh, K., Bhattacharya, A., Pal, A., & Ghosh, S. (2017). (rep.). *Overview of the FIRE 2017 IRLeD Track: Information Retrieval from Legal Documents* (Vol. 2036, pp. 63–68). Bangalore, Karnataka: FIRE 2017 Working Notes.

Ma, Y., Shao, Y., Liu, B., Liu, Y., Zhang, M., & Shaoping, S. (n.d.). (rep.). *Retrieving Legal Cases from a Large-scale Candidate Corpus*. University of Alberta. Retrieved from https://sites.ualberta.ca/~rabelo/COLIEE2021/COLIEE2021proceedings.pdf.