Removing Named Entities to Find Precedent Legal Cases

Ravina More[1], Jay Patil[2], Abhishek Palaskar[2] and Aditi Pawde[1]

[1] Tata Consultancy Services, Tata Research Development and Design Centre, Pune, India
[2] College of Engineering, Pune, India
ravina.m@tcs.com

Abstract. In this paper, we present the solution of the team TRDDC Pune for the Artificial Intelligence in Legal Assistance(AILA) track 1 task on Precedent Retrieval in FIRE 2019. The task was to identify relevant legal prior cases for a legal query from a dataset of about 2,914 documents of cases that were judged in the Supreme Court of India. We used Named Entity Recognition to preprocess the case documents and the input query. We then ranked the preceding case documents using TF-IDF and BM25 algorithms. The results of our approach are comparable to the top ranked run on the task leaderboard.

Keywords: Legal Analytics, Information Retrieval, Legal Precedents, Named Entity Recognition, TF-IDF, BM25

1 Introduction

In countries following the *'Common Law System'* (e.g. UK, USA, Canada, Australia, India), prior cases – also known as *Precedents*, are a primary repository of information for lawyers. By understanding how the Court[1] has dealt with similar scenarios in the past, a lawyer can prepare the legal reasoning accordingly.

When a lawyer is presented with a new case, she/he has to go through the *Precedents* to find out where does his legal problem fit and what was the outcome of similar cases in the past. Going through all the *Precedents* manually involves scanning a large repository, reading through the cases, and finding out the most relevant part in the case document. This process is time consuming. Thus, it is beneficial to have a system that can automatically and efficiently search for a case that you are interested in and find the most relevant *Precedents* We present here our solution that uses Natural Language Processing and Information Retrieval Techniques to find relevant *Precedents* for a given Query for the FIRE 2019[1] Challenge Task 1 of identifying relevant prior cases.

2 Related Work

In the past, substantial work has been done on designing and constructing the corpora of legal cases for legal retrieval. Ontologies and Natural Language Processing are being used to extract case factors and participant roles[2]. Yin et. al[3], demonstrate an approach to query search engines using a document. Our problem statement is similar to theirs as it involves querying using a set of sentences. Their approach works on extracting and scoring key phrases from the query, expanding them with related key phrases and using these in the search engine to find documents containing these concepts. While their approach is based on finding noun key phrases in the query, we are more interested in the overall situation of a given query. We took inspiration from their work to select *interesting* portions in the query and perform ranking of case documents based on them.

3 Problem and Data Description

Artificial Intelligence for Legal Assistance (AILA) track challenge had 2 subtasks. Sub-task 1 was about identifying relevant prior cases. The participants were provided with 2,914 case documents that were judged in the Supreme Court of India. The participants were provided with 50 legal queries, each describing a situation. The task was to retrieve the most relevant *Precedents* among the 2,914 case documents for a given query.

A set of 2-3 relevant case documents was provided per query for the first 10 queries as test data. The participants had to perform relevance ranking for the remaining 40 queries. Refer [1] for more details. For the submission, each query returned a ranked set of prior cases that were judged to be relevant to the query. The relevance of a case document was ranked between 0 to 1 (1 indicating most relevant). The results were evaluated using trec_eval.

4 Methodology

To find the relevant *Precedents* for a given query we followed the following steps:

Step 1: Pre-process all the case documents to build a search corpus (Section 4.2)
Step 2: Pre-process the query (Section 4.3)
Step 3: Rank the *Precedents* from the corpus using the query (Section 4.4)

4.1 Intuition

The queries and the case documents contained substantial information about names, places, organizations, currencies, time, etc. that are specific to the case (E.g. `'Gov. of Tamil Nadu'`, `'Indian Oil Corporation'`, `'13 Rs.'`, `'January afternoon'`, etc.). Such information can be ignored to focus on events such as `'murder'`, `'bribery'`, `'stole'`, etc. that give primary information about the situation to perform relevance ranking.

4.2 Pre-processing of Case Documents:

As the first step, we prepared the corpus according to our intuition for query extraction. We used spaCy[7] for preprocessing. We performed the following steps on the 2,914 case documents:

1. **Paragraph Splitting of case documents:**
   A case document can contain 10-40 paragraphs on an average. These paragraphs give information about the background of the case, the situation and the judgements. We were interested to compare the results of performing a match on the whole case document versus on these individual paragraphs. So, we split every case document into individual paragraphs.
   - ***Improvement after submission:*** *We decided to take the entire case document without splitting it into paragraphs*
2. **Tokenization and Named Entity Recognition (NER) of paragraphs:**
   We used spaCy's tokenization to break down the paragraphs into individual words called tokens. We performed NER on the tokenized sentences to find named entities such as places, things, person, currency, time, etc.
3. **Removal of Named Entities and Stop Words:**
   Using the Named Entities identified in the previous step and a predefined list of stop words by spaCy, we removed the Named Entities and the Stop Words from the case documents (Fig. 1).

1. These appeals are filed against the order dated 29.3.2001 passed by the Madras High Court allowing Crl.O.P. Nos.2418 of 1999. 2. The appellant (Indian Oil Corporation, for short 'IOC') entered into two contracts, one with the first respondent (NEPC India Ltd.) and the other with its sister company Skyline NEPC Limited ('Skyline' for short). According to the appellant, in respect of the aircraft fuel supplied under the said contracts, the first respondent became due in a sum of Rs.5,28,23,501 and Skyline became due in a sum of Rs.13,12,76,421 as on 29.4.1997.	['~~1~~', '', 'these', 'appeals', 'are', 'filed', 'against', 'the', 'order', 'dated', '~~29.3.2001~~', 'passed', 'by', '~~the~~', '~~madras~~', '~~high~~', '~~court~~', 'allowing', '~~crl.o.p~~.', '~~nos.2418~~', 'of', '~~1999~~'] ['~~2~~', '~~the~~', 'appellant', '~~indian~~', '~~oil~~', '~~corporation~~,', 'for', 'short', '~~ioc~~', 'entered', 'into', 'two', 'contracts,', 'one', 'with', 'the', 'first', 'respondent', '~~nepc~~', '~~india~~', '~~ltd~~', 'and', '~~the~~', 'other', 'with', 'its', 'sister', 'company', '~~skyline~~', '~~nepc~~', '~~limited~~', '~~skyline~~', 'for', 'short', 'agreeing', 'to', 'supply', 'to', 'them', 'according', 'to', '~~the~~', 'appellant,', 'in', 'respect', 'of', '~~the~~', 'aircraft', 'fuel', 'supplied', 'under', '~~the~~', 'said', 'contracts,', '~~the~~', 'first', 'respondent', 'became', 'due', 'in', '~~a~~', 'sum', 'of', '~~rs.5,28,23,501~~', 'and', '~~skyline~~', 'became', 'due', 'in', 'a', 'sum', 'of', '~~rs.13,12,76,421~~', 'as', 'on', '~~29.4.1997.~~']
Before Preprocessing of Case Doc	After Preprocessing of Case Doc

Fig. 1. Pre-processing of Paragraph

4.3 Pre-processing of the Query

On reading the queries, we found out that the queries contained information such as background about the situation, the situation itself, subject of the appeal and participants. We define *Appeal Context* as the set of sentences in the query that describe the appeal. As we were interested in the information of the appeal only, we extracted this information from the query by finding the *Appeal Context* and then preprocessing this context.

1. **Extract the *Appeal Context:***
   We observed that most of the queries contained some key-words that help us to identify the context. We used the following list of appeal related key-words:[`'appeal', 'appeals', 'trial', 'hearing', 'plead', 'pleaded', 'appealing', 'cross-appeal', 'quash'`].
   We selected 15 sentences per query containing and surrounding these key words. For queries that did not contain any of these key words or were shorter than 15 sentences, we selected the entire query as the *Appeal Context.*

- ***Improvement after submission:*** *We decided to take all the sentences in the query as the Appeal Context.*

2. **Tokenization, Removal of Named Entities and Stop Words:**
   We performed tokenization of the selected sentences, remove Named Entities and Stop Words of the *Appeal Context*(similar to pre-processing of case documents).

4.4 Performing *Precedent* Retrieval

BM25[4] is 'bag-of-words' ranking function that estimates the relevance of documents provided to a search query. Term Frequency Inverse Document Frequency(TF-IDF)[5] is a measure that helps to identify words in collection of documents that aid to defining the topic of the document. We used the gensim[6] implementations of BM25 and TF-IDF. We used the cleaned appeal as query, cleaned case documents as corpus and BM25 and TF-IDF algorithms to rank the case documents.

Using BM25, TF-IDF and an ensemble of BM25-TF-IDF, we found the score for every paragraph in every case document for a given query. The final score of a case document for a given query is the mean of the scores of the top 3 paragraphs of the case document. We ranked the case documents on a scale of 0 (least relevant) to 1 (most relevant) based on these scores.

- ***Improvement after submission:*** *We cleaned and used the whole case documents (without paragraph splitting) and the entire query (without selecting the Appeal Context) for relevance ranking using BM25 and TF-IDF.*

5 Result and Analysis

Table 1 shows the performance of the runs that we submitted. The results of *'HLJIT2019-AILA_task1_2'* run which topped the leaderboard are given for reference. Our runs appeared in the top 10 in the leaderboard.

Run ID	P@10	MAP	BPREF	Reciprocal Rank
HLJIT2019-AILA_task1_2 (1st)	0.07	0.1492	0.1286	0.288
TFIDF (5th)	0.05	0.0956	0.067	0.203
Ensemble (7th)	0.04	0.0817	0.0591	0.162
BM25 (8th)	0.0375	0.0773	0.0547	0.151

Table 1. Comparison of the performance of the different ranking approaches

5.1 Improvements after submission

After the organizers made the test data public, we performed ablation analysis and realized that the splitting of case documents to paragraphs and selection of the *Appeal Context* were not improving the results, and were in fact deteriorating it. This could be because narrowing down the query and restricting the query search to just the paragraphs led to missing out some key information for comparison. In fact, the simple removal of Named Entities (NE) in both case documents as well as queries improved the ranking results substantially. Table 2 shows the results.

	Removed NE from Case Docs	Removed NE from Query	P_10	MAP	BPREF	Recip. Rank
TFIDF	TRUE	TRUE	**0.07**	**0.1743**	**0.1535**	**0.2771**
	TRUE	FALSE	**0.07**	0.1723	0.1504	0.2738
	FALSE	TRUE	0.0575	0.1319	0.1204	0.1949
	FALSE	FALSE	0.0625	0.1644	0.1468	0.2449
BM25	TRUE	TRUE	0.0575	0.128	0.1163	0.2424
	TRUE	FALSE	0.0575	0.1261	0.1123	0.238
	FALSE	TRUE	0.05	0.1274	0.11	0.2545
	FALSE	FALSE	0.05	0.1487	0.1362	0.2679

Table 2. Comparison of Results after Submission

Removal of Named Entities from both query and case helped in making the comparison more generic. For example, this resulted in all the bribery cases whether they happened in a police station, bank or some private company to be treated equally. According to Table 2. TF-IDF as well as BM25 performed the best when the named entities were removed from the query as well as the case documents. At the same time, TF-IDF performed better than BM25 in all the cases.

6 Conclusion and Future Work

We have presented our approach for finding the relevant *Precedents* in the Task 1 in AILA track in FIRE 2019. After improvements, we found out that simply removing the named entities gave the best results. These results are comparable to the highest ranked approach on the leaderboard.

The BM25 and TF-IDF algorithms used in this approach are both word-matching based algorithms for relevance ranking. As a result, a query containing `'kill'` does not get matched to a case document containing `'murder'`. The lack of exact matches prevented some of the case documents from getting a higher rank in spite of the situation being the same. In the future, we plan to further improve our technique by considering the meaning of the words using word vectors while performing relevance rankings.

7 Acknowledgement

We would like to thank Girish Palshikar, Sachin Pawar, Dr. Kripabandhu Ghosh and Nitin Ramrakhiyani from TRDDC, Pune for their guidance during our brainstorming sessions. We also thank Dr. Vahida Attar, HOD, Department of Computer and IT, COEP, Pune for her support.

References

1. P. Bhattacharya, K. Ghosh, S. Ghosh, A. Pal, P. Mehta, A. Bhattacharya., P. Majumder, Overview of the Fire 2019 AILA track: Artificial Intelligence for Legal Assistance. In Proc. of FIRE 2019 - Forum for Information Retrieval Evaluation, Kolkata, India, December 12-15, 2019.
2. Wyner A., Mochales-Palau R., Moens MF., Milward D. (2010) Approaches to Text Mining Arguments from Legal Cases. In: Francesconi E., Montemagni S., Peters W., Tiscornia D. (eds) Semantic Processing of Legal Texts. Lecture Notes in Computer Science, vol 6036. Springer, Berlin, Heidelberg
3. Yang, Y., Bansal, N., Dakka, W., Ipeirotis, P., Koudas, N., & Papadias, D. (2009, February). Query by document. In Proceedings of the Second ACM International Conference on Web Search and Data Mining (pp. 34-43). ACM.
4. Robertson, S., & Zaragoza, H. (2009). The probabilistic relevance framework: BM25 and beyond. Foundations and Trends® in Information Retrieval, 3(4), 333-389.
5. Rajaraman, A.; Ullman, J.D. (2011). Data Mining. Mining of Massive Datasets. pp. 1–17. doi:10.1017/CBO9781139058452.002. ISBN 978-1-139-05845-2.
6. Rehurek, R., & Sojka, P. (2010). Software framework for topic modelling with large corpora. In In Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks.
7. Honnibal, Matthew, and Ines Montani. "spacy 2: Natural language understanding with bloom embeddings, convolutional neural networks and incremental parsing." To appear 7 (2017).