COL-764 Project Final Report
Precedent Retreival of Legal Cases

Pratik, Prawar, Rohan Debbarma,
Indian Institute of Techonology, Delhi
cs5180415@iitd.ac.in, cs5180417@iitd.ac.in

Abstract

This report evaluates the precedent retreival of legal cases with reference to FIRE-2017 dataset which has citation markers as well as the FIRE-2019 dataset which has no such citation references in the query. Various retrieval methods like BM-25, TF-IDF based vector spaced models, Latent Dirichlet Allocation, Word2Vec, Doc2Vec, Pre-trained Transformer Models are applied to this task and the models compared with respect to metrics such as MAP and MRR. It was found that BM-25 based models modified to rank on sub-queries through rank aggregation performed the best. In the context of retrieval with citation references, sub-query approach which models each query a set of sub-queries corresponding to each citation provided a good accuracy when applied to BM-25 as well as other models such as TF-IDF and Doc2Vec.In case of queries with no citation references, BM-25 was again the best performing model with respect to MAP. Doc2Vec which is a neural model when trained on FIRE2017 dataset with 300 dimensions in an embedding also provided good results and was the 2nd best performing model.

1 Introduction

This report describes the implementation details and the results obtained on applying various models along with some comments and analysis on the performance of various models applied on the task of precedent case retrieval.

The code related to the project can be found at the following link: https://github.com/builder2000/COL764-Project

The approaches used for the two specific situations which are - presence of citation markers and no citation markers are briefly discussed in the sections below along with the results obtained.

2 Approaches Used (Citation Markers Present)

In this section, a brief description of the approaches used (BM-25 and TF-IDF) in this part along with the results is provided.

2.1 BM-25 based models

2.1.1 Citation Context + BM25

As we had already mentioned in the previous report, the FIRE-2017 dataset had citation markers which helped us to identify the regions of query document which are most relevant to a particular citation. In other words, we can view each query document as a collection of sub-queries, each referring to one citation. Each of the sub-queries could then be treated as an individual query in BM-25 and be used to rank the documents.

To combine the rankings of each of the sub-queries in a query, we take the maximum BM-25 score obtained for a document for all the sub-queries. The documents are then sorted in descending order by their BM-25 score.

The procedure is as follows:

1. The sections of text corresponding to each citation marker namely, $[?CITATION?]$ was identified and the text on either side of it upto some number of spaces on both sides each was captured as the sub-query for each query. We experimented with different text fragments with the number of space characters as a parameter and found moderately long queries gave better results. The results of this query length analysis are also prvided below.
2. The query was then tokenized with $word\_tokenizer$ of nltk and experimentation was done with stemming (with PorterStemmer) and stopword removal (Nltk.stopwords).
3. Each of the prior cases which need to be ranked were then tokenized using nltk $word\_tokenizer$.
4. The BM-25 model was then trained on the set of prior cases.
5. Each of the sub-queries of a query case were then fed as queries to obtain a ranking and BM-25 scores for each of the prior cases.
6. For each prior case, we took the maximum of the BM-25 scores it obtained from all the sub-queries.
7. So, using all the subqueries, we now have a final ranking of all the prior cases in terms of their BM-25 scores.
8. Using these scores, a ranking was generated and the metrics such as MAP, MRR and P@10 evaluated for them.

Note- For BM-25, the value of k1 and b used after optimisation was 1.75 and 0.95

2.1.2 Citation Context + BM25 + IDF Screening

In this variant, instead of using the whole query text as the query in a subquery, top 50 $\%$ of the terms in the text ranked by their IDF score are used for rankings. This is done to check whether some of the terms in that query which are specific to the document can be a better representative of the query text.

Table 1: Results of BM-25 and variants with 100 spaces as query length

Method Used	Stopword	Stemming	MAP	MRR	P@10
Citation Context + BM-25	Yes	No	0.477	0.821	0.281
Citation Context + BM-25	No	No	0.477	0.819	0.280
Citation Context + BM-25	Yes	Yes	0.411	0.768	0.249
Citation Context + BM-25	No	Yes	0.419	0.777	0.249
Citation Context + BM-25 + IDF Screening	Yes	No	0.467	0.787	0.276
Citation Context + BM-25 + IDF Screening	No	No	0.466,	0.797	0.271
Citation Context + BM-25 + IDF Screening	Yes	Yes	0.413	0.345	0.253
Citation Context + BM-25 + IDF Screening	No	Yes	0.402	0.720	0.252

2.1.3 Analysis of query length vs MAP

For the best performing method which is Citation Context + BM25 with no stemming, query length was varied to analyse how the various metrics change with length of a query. This would help to get an idea of how much text close to a citation marker is related to the citation. The following table shows the variation of the metrics with the parameter being number of spaces on either side of the marker.

Table 2: Results of BM-25 at various query lengths

Number of spaces	MAP	MRR	$P@10$
50	0.437	0.765	0.264
100	0.478	0.826	0.277
150	0.482	0.823	0.277
200	0.473	0.822	0.274
250	0.458	0.804	0.270
300	0.450,	0.795	0.266
350	0.444	0.791	0.262
400	0.437	0.785	0.254
450	0.429	0.771	0.254
500	0.417	0.753	0.250

As we can see, the MAP, MRR and P@10 metrics increased till query length corresponded to 150 spaces and then decreased slowly. This shows that for moderately long queries, the model performs the best. This behavior is somewhat expected since the text around a citation is only relevant upto some length and if we include text from longer distances, they may not be even relevant to that particular citation.

This kind of results show that a legal document especially judgments though are on a specific broad topic but consist of various sub-topics like precedent cases many of which may not be relevant to each other but are relevant to the current judgement.

2.1.4 Discussion on results

From the above table, it is clear that BM-25 model applied on sub-queries of each query and then using the max BM-25 score of each document gives good result and it has outperformed best method mentioned in [1] by a margin of 0.08 which is a significant improvement. For faster processing, IDF screening variant of BM-25 described above also performs well and gives a MAP of 0.467.

Applying stemming on the query text has yielded lower MAP because in many legal scenarios, some nouns such as named entities and verbs which are generally used in judgements have been degraded to their root forms due to stemming. This has resulted in lower MAP scores. The best MAP achieved with stemming is 0.419 which is around 0.05 less than optimal MAP found.

Stop word Removal has negligible impact on MAP. This is may be because the IDF scores in BM-25 have already made their weights close to zero in an optimally tuned case.

Another key observation is longer queries have provided better MAP in this model. This is may be because longer queries capture more information and relevant words related to the topic of each citation.

Therefore, using max ranking using BM-25 scores of sub-queries has provided significant improvement over the best method mentioned in [1].

2.2 TF-IDF + Pos-tagging based methods

In this particular method, we first created TF-IDF vectorizer and used this model to obtain a ranking for the whole dataset without taking the citation marker into consideration. We used three metrics as the evaluation metrics, Mean Average Precision(MAP), Mean Reciprocal Rank(MRR) and P@10. The model was then modified for several different strategies as discussed below:-

2.2.1 Citation Context

In the model developed above, we had considered the whole document as our dataset. In this part, we consider only a fixed number of tokens(in our case it is 400) around the citation markers in the document which helps us create multiple subqueries. Now, based on the cosine similarity of each subquery, we chose the subquery which gives us the maximum similarity score for a particular query. Using this subquery, we calculate the ranking, which is then further used to calculate the above given evaluation metrics.

2.2.2 Stopwords

Then we modify our model to include stopword removal as well. Hence, we test our model for both stopword containing and stopword removed content. The stopwords consist of the general 'english' language stopwords of nltk library.

2.2.3 Pos-Tagging

After testing for the above two strategies, another new strategy was tested which included the use of pos-tagging on top of the above models. Hence, in this model, we separate the terms into their particular parts of speech and then use only specific parts of speech for ranking the documents. The different combinations used in this method are as follows:-

1. No Pos-Tagging
2. Only using Nouns present in the text
3. Use nouns and verbs present in the text

In the last part, the nouns and verbs are seperately marked(so that similar words which can be both nouns and verbs can be seperated). This helps in improving the overall score.

2.2.4 Results

The scores using different metrics obtained using the different combinations of the above model are as follows:-

Table 3: Results of TF-IDF and Pos-Tagging

Method Used	Stopword	Pos-Tagging	MAP	MRR	P@10
TF-idf vectorizer + citation context	Yes	No	0.365	0.698	0.229
TF-idf vectorizer + citation context	No	No	0.256	0.618	0.164
TF-idf vectorizer + citation context	Yes	Nouns only	0.282	0.587	0.185
TF-idf vectorizer + citation context	No	Nouns only	0.244	0.486	0.173
TF-idf vectorizer + citation context	Yes	Nouns and verbs	0.224	0.450	0.153
TF-idf vectorizer + citation context	No	Nouns and verbs	0.204	0.401	0.148
TF-idf vectorizer + without citation context	Yes	No	0.284	0.556	0.190
TF-idf vectorizer + without citation context	No	No	0.173	0.424	0.124
TF-idf vectorizer + without citation context	Yes	Nouns only	0.247	0.466	0.171
TF-idf vectorizer + without citation context	No	Nouns only	0.204	0.386	0.144
TF-idf vectorizer + without citation context	Yes	Nouns and verbs	0.203	0.368	0.146
TF-idf vectorizer + without citation context	No	Nouns and verbs	0.175	0.331	0.119

It can be observed that the best performance comes when using the vectorizer with citation context and stopword removal with no pos-tagging. Another interesting thing to note here is that when using citation context, the overall scores are higher compared to using full document as the context.

Also, stopword removal increases the overall accuracy of the model in all cases as can be seen from the above table. However, on introducing pos-tagging in the model, the scores decrease. Using no pos-tagging gives the maximum scores, followed by using only nouns as context, and lastly, using both nouns and verbs as context.

Now, varying length of the citation context on the best model obtained above is as follows:- It can

Table 4: Results on changing context length

Length	MAP	MRR	P@10
50	0.307	0.600	0.202
100	0.347	0.652	0.217
150	0.354	0.668	0.228
200	0.363	0.683	0.231
250	0.363	0.693	0.225
300	0.364	0.704	0.224
350	0.368	0.710	0.225
400	0.365	0.698	0.229
450	0.363	0.703	0.225
500	0.359	0.693	0.219

be seen that the metrics scores increases on increase in the context length till 350 and the starts decreasing. This is because the average context length in the dataset is around 350.

Hence, we can see that the best performance comes in the model with citation context with english stopwords with no pos-tagging and a context length of 350 characters.

2.3 Latent Dirichlet Allocation

2.3.1 Implementation Details

In this approach, each document is viewed as a collection of topics with some topic distribution. These topic distibutions corresponding to a document are then modeled as a vector. The cosine similarity of these topic distribution vectors was then used to rank the documents.

The procedure is as follows:

1. Each of the document was first parsed into tokens with a tokenizer and fed to a gensim LDA model. The various parameters in the lda model were experimented with such as number of topics and no of documents in a chunk.
2. Coherence values for each model was calculated. It was observed that LDA model with 20 topics had the best coherence score.
3. Corresponding to each query and prior case, their topic distribution vectors which is the % contribution of each topic in a document is identified.
4. Cosine similarity between the query and document vectors was then used to rank the documents. We also experimented with documents modeled as collection of sub-queries as well as document modeled as a single stand-alone query.

Table 5: Results of LDA-based approaches

Method Used	MAP	MRR	P@10
Citation Context + LDA	0.099	0.238	0.070
Without Citation Context + LDA	0.103	0.245	0.075

2.3.2 Analysis of results

From the results, we can see that LDA has not performed well in this case. The MAP given by is 4 times less than that of the best model (BM25 + Citation Context).

It could be due to the fact that a fixed number of topics don't represent a legal document adequately. The citation information in a query is often quite specific and often can't be approximated by a particular set of topics.

Another interesting observation from this kind of LDA based model is that citation context based approach performs slightly worse than LDA model evaluated on entire query case modeled as one query. This indicates that the topic distributions have more semantic meaning for a full document which is generally generated from a collection of topics rather than a single sub-query/citation in the document which corresponds to a single topic in the query.

2.4 Word2Vec

Recently, word embeddings with context information have shown to be effective in ranking documents. In this setting, Word2Vec algorithm was experimented by training a Word2Vec model on our training dataset. After training on the dataset, word embeddings of size 300 were obtained for each token in the training dataset (which was taken to be the set of prior cases). These word embeddings were converted into text embeddings which represent the whole text.

The procedure is as follows:

1. Each of the documents in the prior cases was tokenized and fed to gensim Word2vec model to generate word embeddings. It was observed that higher the dimension of each word embedding, more was the MAP which is expected since higher dimensional embeddings can capture more contextual information.
2. The number of embeddings corresponding to a word was set to 300 (size used in Google News Word2Vec embeddings, a popular pre-trained Word2Vec model).
3. Each document is then modeled as weighted combination of its word embeddings. The weights in these embeddings are the TF-IDF scores of each word in that document. So, those words with higher TF-IDF score will contribute more to the overall embedding of a document.

Table 6: Result of Word2Vec approach

Method Used	MAP	MRR	P@10
Word2vec	0.083	0.111	0.093

2.4.1 Analysis of Results

From the above result, we can see that Word2vec doesn't provide good results in this setting. One of the principal reasons behind it is the fact that these Word2vec type-neural models require huge corpus to train but in our case the corpus is fairly small and thus meaningful embedding may not have been achieved for all words.

Moreover, the method of weighted combination of embeddings by their TF-IDF score inspired from [3] may not be the most suitable way to represent text embeddings out of individual embeddings. Viewing a sentence as a set of word embeddings is also not exactly accurate, since the word embeddings may not be independent of each other in a sentence.

Another key reason behind compartively worse performance of Word2vec algorithm is the handling of Out of Vocabulary words (OOV). Those words which are not present in the prior cases on which Word2Vec was trained is done by random or mean contribution which is not ideal.

2.5 Doc2Vec

Another neural network based approach which has been proposed to create embeddings of an entire document is Doc2Vec inspired from the Word2vec algorithm.

In this setting, each document is then fed to a Doc2vec gensim model to obtain a text embedding vector of 300 dimensions. Cosine similarity between query document pairs is then used to rank the prior cases. Sub-query based approach of modeling each query as a collection of shorter queries has shown to be effective in achieving good results in this case. The citation length corresponding to highest MAP is 150 spaces on both side which points to the fact that moderately long queries have performed better similar to BM-25 model.

The following results are obtained for the Doc2Vec model:

Table 7: Result of Doc2Vec approach

Method Used	MAP	MRR	P@10
Citation Context + Doc2Vec	0.386	0.747	0.229
Without Citation Context + Doc2Vec	0.163	0.364	0.111

2.5.1 Analysis of query length vs MAP

For the best performing method which is Citation Context + Doc2Vec, query length was varied to analyse how the various metrics change with length of a query. The following table shows the variation of the metrics with the parameter being number of spaces on either side of the marker.

Table 8: Results of Doc2Vec at various query lengths

Number of spaces	MAP	MRR	P@10
50	0.250	0.554	0.170
100	0.368	0.695	0.227
150	0.386	0.747	0.229
200	0.383	0.762	0.227
250	0.384	0.761	0.229
300	0.375	0.726	0.226
350	0.366	0.736	0.218
400	0.351	0.714	0.210
450	0.343	0.700	0.202
500	0.338	0.700	0.205

The table shows similar behaviour to the case of BM-25 with maximum accuracy at 150 spaces and decreasing then. From the table, we can see the change in MAP for higher query lengths is not very significant. However, for short querues, there is a huge variation of MAP as we go to higher lengths. (eg, 50 to 100 spaces query correspond to MAP change of 0.11 from 0.250 to 0.368).

2.5.2 Analysis of results

From the above table, we can see that a Doc2Vec trained on the prior cases gives good results (2nd best in terms of MAP). This is in contrast to the submissions in [1] in which no such neural model achieved such high rank. We also observed that higher number of dimensions in a text embedding results in better MAP scores since they capture more contextual information as compared to shorter vectors.

Another interesting observation related to this model citation context information has a significant impact on MAP as we can see the MAP has jumped from 0.163 to 0.386.

2.6 Pre-trained Transformer Model

In this approach, a pre-trained transformer model known as "Legal-BERT" which is trained on European Legal Documents is used to rank the model. Transformers library is used to load this model and define a concept of sentence embedding for it.

The document embedding thus obtained for query and document is then compared with cosine similarity to rank the documents.

2.6.1 Implementation Details

The following steps were involved in inferring the document embedding from the documents.

1. For each document, [CLS] tokens were inserted to fine-tune the model and generate word embeddings corresponding to the tokens.
2. Text embedding was then calculated as the mean of the embeddings for the words.
3. The generated text embeddings were used to compute cosine similarities between document and queries to rank the documents.

Table 9: Result of Doc2Vec approach

Method Used	MAP	MRR	P@10
Pre-trained Transformer + Embedding	0.060	0.175	0.0432

2.7 Analysis of results

From the results, it can be seen that transformer based model has the least MAP for this dataset.

One of the reasons behind the low MAP value is the obtaining an embedding for the words. BERT can at max work on 512 tokens. However, in our case, many of the documents have number of tokens significantly higher than 512 and the truncation of such documents has not been very accurate.

Another reason for its worse performance is due to its handling of OOV words for embedding creation by taking the mean. This might not be the ideal solution but is a limitation of pre-trained models.

Moreover, though the BERT is trained on legal documents, it corresponds to a set of European Legal and legislative documents. Many of the India-specific named entities may not have accurate word embeddings. The approach to take the mean of word embeddings to represent a sentence embedding may also not be the best representation.

3 Approaches Used (Citation Markers Absent)

The approaches used in this scenario are the best two methods corresponding to the FIRE-2017 dataset i,e, BM25 and Doc2Vec models.

It is evaluated on FIRE-2019 dataset which does not have any citation markers.

The approaches used is same as the section described above. However, the document is now a single query instead of a collection of sub-queries.

3.1 Results

The following result was obtained for the FIRE-2019 dataset.

Table 10: Overall Results of best models for FIRE-2019 dataset

Method Used	MAP	MRR	P@10	Rec@100
BM-25	0.112	0.197	0.044	0.034
Doc2Vec	0.067	0.174	0.024	0.034

3.2 Analysis of Results

In comparision to methods submitted in [2], our best method in this case is less than the best method in [2]. The best method in [2] uses weighted re-ordering of results (some kind of relevance feedback) which is not taken into account by us, resulting in lesser MAP score.

4 Discussion on overall Results

From the above work, we can see that BM-25 with citation context has significantly outperformed other retrieval models. Doc2Vec trained on citation context was also gave good results in adherence of the findings of [3] in regard to legal documents.

Table 11: Overall Results of best models for FIRE-2017 dataset

Method Used	MAP	MRR	P@10	Rec@100
Citation Context + BM-25	0.482	0.823	0.277	0.794
Citation Context + Doc2Vec	0.386	0.747	0.229	0.705
TF-idf vectorizer + citation context	0.368	0.710	0.225	0.754
LDA	0.103	0.205	0.745	0.508
Word2Vec + citation context	0.083	0.111	0.093	–
Pre-trained Transformers + citation context	0.060	0.175	0.0432	–

Citation context markers which each query case as a set of sub-queries have been shown to be effective in better retrieval of legal documents. This type of modeling has significant impact on the MAP since using the whole query document as a single query has not provided us good accuracy due to the fact that each query case is quite large and two different citations in a case are maybe on entirely different type of prior cases.

This report has been an attempt to comprehensively perform query retrieval using the various models based methods and the methods reported here has been able to outperform the best methods used in [1].

One of the biggest reasons for the better performance of our models compared to [1]. is the introduction of citation context in our models. Most of the models in [1] work on the whole document. However, different citations can be treated as different prior cases which and hence considering them separately provides with a major boost in the performance of our models.

Now, since we have introduced a new parameter(citation context) in our model, we need to optimize the other parameters in each of the models accordingly in order to get the best performance, which once done, provides us with the above obtained results.

Another important thing to note is that the original paper included pos-tagging in some models. However, using pos-tagging in our models significantly reduced the performance of the models. This is because a lot of significant adjectives and prepositions are getting removed due to pos-tagging which reduces the performance of the models.

Our work shows Doc2Vec trained on the dataset when combined with citation context can provide good results and it has ranked 2nd best among our models. This is in contrast to [1] which did not have any doc2vec based models with high MAP score since the citation context information was not used by those methods.

5 Concluding Discussion

In this project, we experimented with various methodoligies for retrieving precedent/prior cases for a given case. Both standard Vector Space Models as well as Neural Models such as Doc2Vec have been experimented with.

We observed that citation context information in a current case helps in better retrieval of cases in many of the settings which is in coherence to the view of legal experts who have viewed a legal document as a collection of different legal issues. The text based similarity models implemented here have taken advantage of this fact for better quality retrieval.

We have been able to outperform the best method in [1] by a significant margin of 0.09 through citation information and also show the power of neural models such as Doc2vec in quality retrieval of documents.

However, the problem becomes more challenging when no such context information is available. In that case, the MAP score achieved is not very high (only 0.112) which reflects the challenges of legal document retrieval and this is a possible exploration area to look into.

6 References

1. Mandal, A., Kripabandhu Ghosh, Arnab Bhattacharya, Arindam Pal and Saptarshi Ghosh. "Overview of the FIRE 2017 IRLeD Track: Information Retrieval from Legal Documents." FIRE (2017).
2. Paheli Bhattacharya, Kripabandhu Ghosh, Saptarshi Ghosh, Arindam Pal, Parth Mehta, Arnab Bhattacharya, and Prasenjit Majumder. 2019. FIRE 2019 AILA Track: Artificial Intelligence for Legal Assistance. In Proceedings of the 11th Forum for Information Retrieval Evaluation (FIRE '19). Association for Computing Machinery, New York, NY, USA, 4–6.
3. Mandal, Arpan Chaki, Raktim Saha, Sarbajit Ghosh, Kripabandhu Pal, Arindam Ghosh, Saptarshi. (2017). Measuring Similarity among Legal Court Case Documents. 1-9. 10.1145/3140107.3140119.