Legal Knowledge and Information Systems
E. Francesconi et al. (Eds.)

doi:10.3233/FAIA220471

Autosuggestion of Relevant Cases and Statutes

Saran PANDIAN [a,1], and Shubham JOSHI [a,2]
[a] *Lawnics Technologies, India*

Abstract. In this paper, we describe a method to help legal practitioners in citing the relevant case laws and statute laws for the specified legal issue. In this method, we consider the cited case and statute law as single tokens where we try to find the relevant tokens based on the words around these tokens. We observed that context-based representations outperformed lexical-based representations and distributional representations. Also, we observed that the method works better for statute law retrieval compared to case law retrieval.

Keywords. Precedents, Statutes, Masked Language Model, Citation Recommendation, Distributional Representation, Context-Based Representation

1. Introduction

To understand which statutes or/and cases are the most relevant to the legal issue, we need to check what laws have been frequently cited in solving contextually similar legal issues or queries in past. In this paper, we try to investigate novel approaches to create an autosuggestion tool to predict the most relevant cases and statutes for similar contextual legal issues/queries. When a legal citation can be represented based on the context around it, it can be easily retrieved using a search engine when words with similar context are given as queries. Moreover, we also study the effect of the frequency of citations impacts the performance of the model.

2. Related Work

Traditional techniques for citation recommendation usually include BM25[1], Indri[2], etc. Among these BM25 is considered to be a strong baseline when it comes to the legal domain[3]. However, BM25 considers only lexical matching and not semantic matching. With the advent of deep learning techniques, research in direction of including the semantics of the documents for citation prediction tasks is widely done[4]. BM25+BERT[5] has shown better results than BM25. Our challenge was to come up with an approach for citation recommendation that finds relevant citations from a larger pool of candidate citations and recommends citations regardless of the length of the cited documents.

[1] Saran Pandian, saran@lawnics.com
[2] Corresponding Author: Shubham Joshi, sshubham@lawnics.com

3. Dataset Creation

The key requirement of the dataset was to find legal texts that was rich with the citations. Judges who wants to use laws as a reference in their judgements generally cites cases and statutes around the legal issues that the judgements deals with, hence we decided to include paragraphs of judgements that has citations in it. Since the Supreme Court of India is considered to be the highest authority and is cited multiple times through out all the courts we decided to include only judgments passed by this court. To prepare the dataset, we considered more than 55000 supreme court cases[3] from the year 1947 to 2021. Despite India having many statutory laws at central and state level, we have included only 1260 statutes as these were passed by central government having effect all over India. Data used from these statutes[4] were also included along with SC cases to create the legal corpus.

All these cases and statutes consist of multiple paragraphs. It was found that a total of 16,37,897 paragraphs were available. Using regular expressions we found paragraphs having case citations (paras containing words vs., Vs, Vs.) and statute citations (paras containing words Section, sec.,s., Article, Art., art., Act, act, etc.). 54541 paragraphs were found to have case citations and 242484 had statute citations. These paragraphs were then manually reviewed for additional annotation(especially for acts corresponding to sections) using 10 volunteers from the legal industry. Legal Volunteers were provided with proper guidelines and tools[5] to find out the citations and annotate them for higher accuracy. Also, each volunteer was then asked to review the annotations of other volunteers to confirm the quality of the dataset. We decided not to go for an Inter-annotator agreement as the task of finding citations is of elementary level for legal volunteers. Then using our proprietary database, each cited case and statute is given its ID using pattern matching with help of the name, year of judgment (in case of precedent law), and enactment(in case of statute law).

The citations are not spelled uniformly throughout the corpus; for example, the cited case could be *Maneka Gandhi v. Union of India*. But the annotated title of this cited case is *Maneka Gandhi vs UOI*. To normalize the text, we use fuzzywuzzy[6] string matching algorithm to match the annotated citation with the actual document name from a lookup table and replace it with the corresponding ID. Scores for matches were given on a scale of 100, based on the number of overlapping tokens and the order of tokens. In order to avoid string mismatches, the threshold of score 70 was set. The final dataset contains preprocessed legal text with citations being replaced with IDs.

The total number of unique citations were found to be 29010 out of which 7457 were statute citation and 21553 were case citations. The highest statute citation was found to be 11935 and the case citation was 220. We would like to convey that the dataset is part of a proprietary dataset for a commercial application. Hence the dataset cannot be published.

For final preprocessing we converted raw text to lowercase, followed by the removal of stopwords, numerals, and punctuations. It is critical to note that no preprocessing is needed to be done on citation IDs. For our approach, we used 16,37,897 paragraphs

[3] https://main.sci.gov.in/judgments
[4] https://indiacode.nic.in/
[5] https://ubiai.tools/
[6] https://pypi.org/project/fuzzywuzzy/

scraped from Indian Supreme court cases and statutes for this experiment. These paragraphs consist of text where we replaced the title of cited law with corresponding citation IDs. Further, we decided to split paragraphs in an 80:20 ratio randomly for training and testing respectively.

4. Methodology and Experiments

For a given context around a citation, the task is to find the corresponding citation c_i from citation set C. We try to Autosuggest the token based on the context. For this experiment, we consider only citations that have been cited more than 3 times in the corpus. this paper, we provide techniques for the recommendation of citation thus auto-suggesting statutes and cases based on textual context. In legal literature, citations are based on text. The whole purpose of the experiment is to make AI models understand the context and recommend citations for similar contextual queries by going through the text of cited law. In NLP, Predict Distributional Semantics Model(DSM) and context-based Distributional Semantics Model use co-occurring words around a word to arrive at a representation of the semantics of a word. Words occurring in similar environments tend to be close w.r.t semantic representation[6]. Predict DSM representation is unique for each word whereas context-based DSM representations differ with respect to sentences. Each citation is considered to be a single token, to arrive at a semantic-based representation for each citation. Two models namely LawCite2vec (Predict DSM method) and Bert4LawCite (Context-based DSM method) were trained and each of these DSM models is compared with BM25 and BM25+BERT.

4.1. LawCite2vec and Bert4LawCite

4.1.1. Training

Mikolov et al[7] used the skip-gram technique in their paper to get vector representation for words. Rather than training the LawCite2Vec model from scratch, the existing Law2Vec model[8][7], which is pre-trained on a substantial legal corpus was further fine-tuned on our training data, where we considered each citation to be a single token. We completed text preprocessing as mentioned in 3. To carry out the task, we considered a window size of 10 to train the model for 30 epochs, as some citations are scarcely cited. We trained the model with the help of 12 GB RAM processing power.

Lately, BERT[9] has become a prominent state-of-the-art model for all downstream NLP tasks. We also decided to use BERT for token prediction tasks where we try to predict tokens consisting of citation Ids. For each query consisting of the masked token, we are trying to predict the citation Ids as an output. We decided to use pre-trained LegalBERT[10], as this model has already been trained on a large corpus of legal texts[8]. We loaded the pre-trained weights to finetune it for our downstream task as it is a context-based DSM and can change the representation of the masked token based on the context around it. For training Bert4LawCite, We need to pick sentences from paragraphs that have citations within them. 298,946 sentences from train paragraphs with statute cita-

[7]https://archive.org/details/Law2Vec

[8]https://huggingface.co/nlpaueb/legal-bert-base-uncased

tions and 16377 sentences with case citations were picked with citations replaced with [MASK] token. Two separate models one for statute recommendation and the other for case recommendation were trained in order to avoid bias. We trained the model with the help of a K80 GPU with 12 GB RAM processing power.

4.1.2. Testing

Firstly candidate citations need to be indexed. For recommendation using LawCite2Vec embeddings, the candidate citation embeddings were indexed in Elasticsearch[9] service. For testing purpose we decided to index the embeddings of citations that have been cited more than 3 times as the citation is relevant to legal context only when it has been cited multiple times. It was found that 3133 statute citations and 5174 case citations (denoted by Cs and Cc respectively) had been cited more than 3 times. To come up with queries, around we can considered 39000 sentences from 20 percent of paragraphs chosen for testing that contains 4800 citations. Then these citations were removed for the queries. Queries preprocessed as the mentioned in 3 .The query representation is done by taking the mean average of the LawCite2Vec representation of words in the sentence. The citations with the closest vector representation to this query representation are retrieved through Elasticsearch. The experiments are done for statute retrieval and case retrieval separately.

To test Bert4LawCite, we used similar test data to test the LawCite2Vec model. We decided to remove citation tokens and place a [MASK] token in the middle of the sentence. As with every other prediction model, BERT also considers each word in the whole dictionary and provides the most relevant scores. However, for our citation recommendation task, we want to restrict the vocabulary to the statute citation set Cs and case citation set Cc. After feeding the query that consists of a masked token, we get the score for each citation ID from C_s and C_c sets and consider the highest ranked as the final output of the model.

5. Baseline

We decided to use BM25 as a baseline after going through previous research papers in the legal domain[3]. To make a fair comparison of BM25 output with the other two approaches, we indexed data of cases, sections, and acts in Elasticsearch that were masked and cited at least three times in the test documents. Moreover, we also preprocessed data and queries as per the steps mentioned in 3. We further retrieved the relevant data as per queries and ranked the same based on the BM25 score. On top of BM25, we also tried using BERT for re-ranking by taking the mean of LegalBERT[10] representation of every paragraph used to represent the whole document. The documents were ranked based on the distance between the LegalBERT representations of the query and the documents.

6. Results and Analysis

A citation that has been cited multiple times for a similar legal context can also be considered a trustworthy and relevant candidate for a query with a similar context. As we

[9]https://www.elastic.co/

are only considering a frequently cited candidate as the most relevant candidate for the given mask or BM25 score, we understand that relevance values shall be binary only. We considered Mean Reciprocal Rank(MRR) and Hit Rate(HR) as the most suitable metrics for evaluation as they are prominent with binary outputs. MRR score is based on how far the first relevant item is present in the recommended list (In our case only consider one document is the most relevant). Reciprocal Rank(RR) is the reciprocal of the rank at which the first relevant document was retrieved. 1 is the best RR score which means the relevant document is retrieved in the first place. HR score is based on the ratio of relevant recommended items to the total number of relevant items for a query. 1 is the best HR score which means all relevant documents are present in the recommended list. Further, we evaluated the results for the statute and the case law recommendations separately as we wanted to analyse the score difference between the two thoroughly. For an in-depth analysis of the results, we conducted a citation frequency analysis.

6.1. Results

6.1.1. Statute Law Recommendation

LawCite2Vec and Bert4LawCite performed better than BM25 and BM25+BERT, as per hypothesis. As shown in 1, the statute laws performed better as the frequency of the citations was very high, allowing the models to give better representations for the statute citations. BM25+BERT gave poor as the BERT failed to have good representations for lengthy legal documents. Bert4LawCite gave better results as it considers every citation to be a single token and uses a SOTA language model with higher perplexity[9] to predict the citations.

Table 1. Evaluation Scores

	Statute Law Recommendation			Case Law Recommendation		
	HR@1	HR@10	MRR@10	HR@1	HR@10	MRR@10
BM25	0.0916	0.2571	0.1400	0.1397	**0.3303**	0.1993
BM25+BERT	0.0157	0.08773	0.03182	0.0068	0.0459	0.0148
LawCite2Vec	0.1499	0.3930	0.2147	0.0222	0.0848	0.0385
Bert4LawCite	**0.3607**	**0.6730**	**0.4620**	**0.2120**	0.2987	**0.2398**

6.1.2. Case Law Recommendation

Though the number of case IDs was more than statute IDs, each case ID's overall frequency was less than statute IDs. Bert4LawCite managed to beat BM25 with a considerable difference, but LawCite2Vec performed poorly for the auto suggestion for Cases.

6.2. Citation Frequency Analysis

To understand the effect of the Frequency of citations on the performance of the model the test data was divided into chunks of equal ranges based on the frequency of citations. It was found that frequency has a strong effect on Performance. Even for Case Law Retrieval, where LawCite2Vec was found to perform worse compared to BM25, performed better than BM25 for chunks with higher frequency ranges as shown in Fig 1. We are neglecting BM25+BERT for analysis because of poor performance.

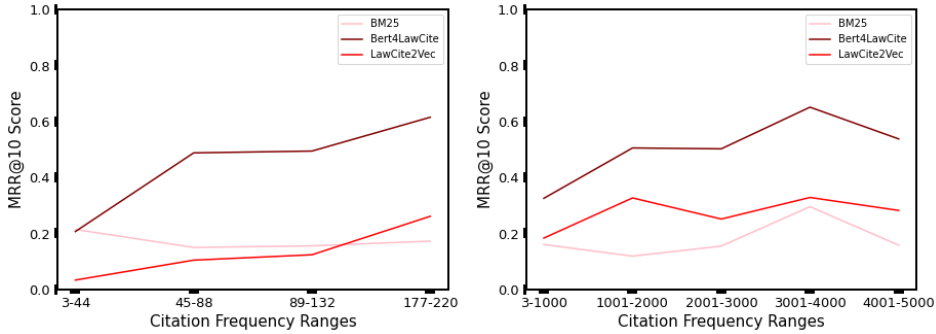

Figure 1. Citation Frequency Analysis for Statute Law and Case Law Recommendation

7. Conclusion and Future Scope

We have introduced a novel approach for citation autosuggestion/ recommendation task by turning multiple-length citations with a more straightforward single token ID and using it to train models like LawCite2Vec and Bert4LawCite. Through this approach, we found that these systems can be used for commercial applications in the field of law where it can be used as stand alone recommendation system for a legal question as a query to recommend relevant cases and statutes. Also, it can be used as re-ranking tool to improve the mean average precision of legal information retrieval tool. We believe future research directions can be conducted by including the content of the cited laws along with context to reduce the dependence on citation frequency and increase the diversity and serendipity of data to reduce the bias on highly cited citations.

References

[1] G. Amati, *BM25*. Boston, MA: Springer US, 2009, pp. 257–260. [Online]. Available: https://doi.org/10.1007/978-0-387-39940-9_921

[2] T. Strohman, D. Metzler, H. Turtle, and W. B. Croft, "Indri: A language-model based search engine for complex queries (extended version)," *Rapport technique. CIIR*, 2005.

[3] G. M. Rosa, R. C. Rodrigues, R. Lotufo, and R. Nogueira, "Yes, bm25 is a strong baseline for legal case retrieval," *arXiv preprint arXiv:2105.05686*, 2021.

[4] Z. Huang, C. Low, M. Teng, H. Zhang, D. E. Ho, M. S. Krass, and M. Grabmair, "Context-aware legal citation recommendation using deep learning," in *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Law*, 2021, pp. 79–88.

[5] R. Nogueira and K. Cho, "Passage re-ranking with bert," *arXiv preprint arXiv:1901.04085*, 2019.

[6] Z. S. Harris, "Distributional structure," *Word*, vol. 10, no. 2-3, pp. 146–162, 1954.

[7] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[8] I. Chalkidis and D. Kampas, "Deep learning in law: early adaptation and legal word embeddings trained on large corpora," *Artificial Intelligence and Law*, vol. 27, no. 2, pp. 171–198, 2019.

[9] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[10] I. Chalkidis, M. Fergadiotis, P. Malakasiotis, N. Aletras, and I. Androutsopoulos, "LEGAL-BERT: The muppets straight out of law school," in *Findings of the Association for Computational Linguistics: EMNLP 2020*. Online: Association for Computational Linguistics, Nov. 2020, pp. 2898–2904. [Online]. Available: https://aclanthology.org/2020.findings-emnlp.261