Document Level Embeddings for Identifying Similar Legal Cases and Laws (AILA 2020 shared Task)

Intisar Almuslim, Diana Inkpen

School of Electrical Engineering and Computer Science
University of Ottawa, ON, Canada
ialmu068@uottawa.ca, diana.inkpen@uottawa.ca

Abstract

In this age of legal big data, there are massive amounts of legal documents available online. The Information Retrieval systems play an important role in accurately retrieving the related information based on a given query. Therefore, there is a need for automated systems which can identify the set of the most relevant precedents (prior cases) and suitable laws (statues) for any situation. This automatic linking of the relevant documents helps lawyers to properly deal with related cases. This work seeks to develop such an automatic system to identify relevant prior cases and statutes for a given query. It has been submitted as a participation in the shared task track named Artificial Intelligence for Legal Assistance (AILA 2020). In this work, we propose a text similarity approach to retrieve previous cases and statues that are similar to a given case from a set of legal documents. For this legal document retrieval task, we present our approach that finds the similarity between a given query and prior cases/statutes documents by applying three variations of word representation based on Glove, Doc2Vec, and TFIDF methods. After feature extraction and documents vectorization, the similarity between each query document and all the prior cases/statute documents are determined using cosine similarity scores. Then, a ranked list of candidate documents is retrieved for each query. Our experiments demonstrate that TFIDF method achieved reasonable results when compared to Doc2Vec and Glove methods which usually need large training datasets.

Keywords

Legal Information Retrieval, Document similarity, Word representation

1. Introduction

In those countries with the Common Law systems like the United States, Canada, and India, there are two main sources of law: the established laws (or statues) and the precedents (or prior cases) [1]. In the judgement of the Common Law system, the primary consideration is to recognize the most related legal principles and the similar cases of a given situation. The method of identifying such representative and associated documents, however, is a time-consuming process. As a consequence, automating this procedure to be used for legal purposes is quite important.

In this work, we describe our methods that were submitted to the Artificial Intelligence for Legal Assistance (AILA 2020) track [2] of the Forum for Information Retrieval Evaluation (FIRE 2020). This track defined two main tasks such as Task1: *Precedent & Statute Retrieval* and Task2:

Forum for Information Retrieval Evaluation 2020, December 16-20, 2020, Hyderabad, India

Rhetorical Role Labelling for Legal Judgements. Our team participated only in Task1 that contains two subtasks. Subtask 1A: *Identifying relevant prior cases for a given situation* and Subtask 1B: *Identifying the most relevant statutes for a given situation.* For both subtasks, a dataset of Indian legal documents have been provided to the participants, with Indian statutes and prior cases decided by Indian courts of Law. In order to find the similarity between the terms in the query and the legal documents, we calculated the relevance between the query and the document by utilizing three different word representations: GloVe, Doc2Vec, and TF-IDF.

The rest of this paper is organized as follows: Section 2 presents related works in the area of legal document retrieval; task description and dataset details are provided in Section 3; Section 4 explains the proposed methods; Section 5 relates to the evaluation results. Finally, Section 6 concludes the work with some directions of future improvements.

2. Related Work

With the increased availability of legal text in digital form, the focus on developing automated legal information retrieval models and applications have received attention from the research community. In recent years, there have been some competitions on the topic of legal information retrieval. The FIRE 2017 IRLeD Track [3] defined two tasks; Catch Phrase Extraction and Precedence Retrieval. The FIRE 2019 AILA Track [1] also focused on two tasks: to identify a set of related prior cases as well as relevant statutes for a given situation. In the same manner, COLIEE-2019 (Competition on Legal Information Extraction and Entailment) [4] has been held as an associated event of the International Conference on Artificial Intelligence and Law (ICAIL). The main objective of this competition is to develop techniques of retrieval, extraction and entailment in the legal field.

For achieving these tasks of retrieving the required legal documents, the submitted works have used various traditional methods of Information Retrieval, as well as different Machine Learning (ML) and Deep Learning (DL) methods. Among many techniques that have been reported in the existing literature, the most applied ones are TF-IDF [5], BM25 [6] or ensembles of the two [7]. Also, embedding methods like Word2Vec, Sent2Vec, and Doc2Vec were widely applied by many of the related studies [8, 9, 10]. Other researchers try to use ML methods [11, 12] to capture semantic similarity. With the recent increase of DL technology and Natural Language Processing (NLP), many researchers applied Neural Network (NN) models to legal information retrieval [9, 13] . In addition to these methods, some studies focused on solving the challenges by employing techniques such as keyphrase extraction [14] and text summarization [15, 16].

3. Dataset Details and Tasks Description

In this work, we have carried out experiments using the dataset released by the organizers of AILA 2020 shared task[1]. This dataset includes a collection of Indian legal documents, i.e., statutes in India and prior cases decided by Indian courts of Law. In addition, a set of 50

[1] https://sites.google.com/view/aila-2020

queries (Q1 to Q50) that contain the descriptions of the legal situations have been given to the participants.

The objective of the Task1A is to identify the relevant prior cases for the given query. To perform it, a set of 3257 labeled prior cases (C1 to C3257) documents have been provided. The participants of this task have to retrieve the most similar/ relevant case documents with respect to the situation in the given query. For performing Task1B, 197 statues (S1 to S197) were released that contain titles and textual descriptions and the task is to identify the most relevant statues for each query.

4. Methods Description

In our approach, we have used different document embedding models with the cosine similarity measure [17] to find the similarity between documents. We submitted 3 runs for each subtask and used three different variations of document representations in these runs. For both subtasks, the considered methods are (i) GloVe, (ii) Doc2Vec and (iii) TF-IDF in the three runs, respectively. We have implemented our methodology in Python and the steps used in our approach are presented in Figure 1.

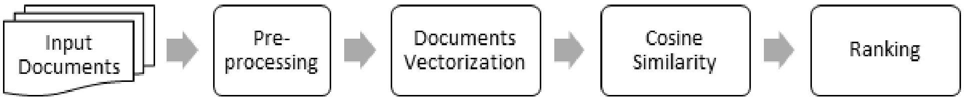

Figure 1: the followed steps in our approach

As input, the collection of prior cases, statutes and query documents are provided in the form of text files. For all runs, the first step is the same which is the pre-processing of the text. For every query and candidate document, we kept only words with letters and all the other characters are removed as well as stop-words. All documents were tokenized, and every token was converted to lowercase and lemmatized using the WordNet Lemmatizer from NLTK [2]. After the pre-processing phase, the steps performed in all the three methods are explained in detail in the following subsections.

4.1. GloVe-based Method

In the first experiment, we have applied the pre-trained Global Vectors for Word Representation (GloVe) [3] to represent the documents of prior cases/statutes and queries as vectors. GloVe is provided by the Stanford NLP team [18] as various models from 25, 50, 100, 200 to 300 dimensions base on 2-, 6-, 42-, and 840-billion tokens. We used the one with 50 dimensions and 6 billion tokens. The words with respect to a particular document are vectorized using

[2] http://www.nltk.org

[3] https://nlp.stanford.edu/projects/glove

GloVe. For those words that do not have pre-trained embeddings, we calculated and assigned to them the average vector of all the word vectors. The average of all the word vectors of each document is determined and that average represents the vector for that document. As a result, the vector representations of all the prior case/statute documents and the query documents are obtained.

After that, the obtained vectors of queries and prior cases/statutes are compared using cosine similarity. As illustrated, for subtask 1A, cosine similarity scores between each query and prior case documents are calculated, sorted, and ranked. Similarly, for subtask 1B, cosine similarities between each query and statute documents are found. The similarity scores are then sorted and ranked. The final result gives a list of ranked documents for each query document.

4.2. Doc2Vec-based Method

In this experiment, we first trained a Doc2Vec [19] model using the Gensim [4] library (Doc2Vec and TaggedDocument from gensim.models.doc2vec) on 3257 case documents for subtask 1A, and on 197 statute documents for subtask 1B. Then, this model is used for documents vectorization. The Doc2Vec method is based on the Word2Vec method that was originally proposed by [20]. While the Word2Vec algorithm builds distributed semantic representations of unique words, the Doc2Vec method is generally applicable to the text segments of any length – from sentences to whole documents.

For this method, each tokenized document is labelled with a tag to uniquely identify documents. The hyperparameters applied are as follows: vector dimension: 100, number of epochs: 50. The trained model is then used to obtain the vector representation of each case document for relevant prior cases task or of each statute document for relevant statutes task. Then, the obtained vectors of queries and cases/statutes are compared using cosine similarity as the description illustrated above, and the candidates are returned in decreasing order of similarity score with the query.

4.3. TFIDF-based Method

In the last experiment, we used the TF-IDF [17] feature weighting measure. In this method, the vector for each document is constructed using TF-IDF weights (or scores) which give high value for a given term if that term appears often in that particular document and not frequently in other documents. As mentioned earlier, after the vectorization step, the cosine similarity scores between each query document and all the prior cases/statutes documents are determined and then ranked.

For each current query, the prior cases and the statutes are ranked based on the similarity scores (the candidate document with highest similarity score is retrieved first). The TF-IDF scores are obtained for the features by using scikit-learn library (TfidfVectorizer from sklearn.feature_extraction.text). The similarity between each current query and the prior cases/statutes are obtained using cosine_similarity from sklearn.metrics.pairwise.

[4] https://radimrehurek.com/gensim/models/doc2vec.html

Table 1
Results obtained in Task 1A using three methods.

Run_ID	MAP	BPREF	recip_rank	P @ 10
run1_Glove	0.0123	0	0.0222	0
run2_Doc2Vec	0.0029	0	0.0051	0
run3_TFIDF	0.0837	0.0399	0.1157	0.05

Table 2
Results obtained in Task 1B using three methods.

Run_ID	MAP	BPREF	recip_rank	P @ 10
run1_Glove	0.1462	0.084	0.345	0.1
run2_Doc2Vec	0.0441	0.008	0.067	0.02
run3_TFIDF	0.2506	0.186	0.3144	0.12

5. Results and Discussion

This section describes the results obtained in the two tasks by our proposed approaches. To evaluate the performance of the submitted works, different evaluation metrics like MAP (Mean Average Precision), P@10 (Precision@10), BPREF (Binary PREFerence-based measure) and recip_rank (Reciprocal Rank) have been used [1]. The primary measure was the MAP score, as reported by the organizers, because this metric involves both Precision and Recall. The performance metrics of all our models on the testing data are shown in Table 1 and Table 2.

According to the results, Tf-IDF vectorization seems to perform better than the other vectorization methods, namely GloVe and Doc2Vec, for the tasks of identifying the relevant prior cases/statutes. Among all 26 runs submitted for Task 1A and 29 runs for Task 1B, our Tf-IDF model got a rank of 17 and 11 on Task 1A and Task 2B, respectively based on the MAP metric.

The embedding methods GloVe, and Doc2Vec have not achieved good results. The main reason is probably because these methods require a huge amount of data to be trained on. In addition, pre-trained general-domain models, like GloVe, do not perform well in very specialized domains such as the legal domain. From the released results of all participants, the best performing methods achieved a MAP score of 0.1573 and 0.3851 on Task1A and Task1B, respectively. These results confirm that the tasks of finding relevant precedent cases and statutes are undeniably challenging tasks.

6. Conclusion

This paper presents the system description of our participation in the Artificial Intelligence for Legal Assistance (AILA 2020) track, which is an associated event of Forum for Information Retrieval Evaluation (FIRE 2020). The track provided two main tasks. Our team only participated in Task1. The shared task defined Task1 with two subtasks as follows: Task 1A,identifying relevant prior cases for a given query and Task 1B, identifying the most relevant statutes for a given query. We submitted three runs for each subtask. In both subtasks, our models are based on Glove, Doc2Vec and TFIDF. For retrieving the relevant cases and statutes, initially the documents were vectorized using the three methods. After vectorizing, the documents were all ranked by their similarity to the query using cosine similarity.

It can be seen from the final evaluation results that the TF-IDF-based method achieved satisfactory results when compared to our other submitted methods. The models' performance can be improved by extracting different features or by applying different similarity measures. Further improvement might also be achieved by using state-of-the-art deep learning models pre-trained on legal text such as LEGAL-BERT [21].

References

[1] P. Bhattacharya, K. Ghosh, S. Ghosh, A. Pal, P. Mehta, A. Bhattacharya, P. Majumder, Fire 2019 aila track: Artificial intelligence for legal assistance, in: Proceedings of the 11th Forum for Information Retrieval Evaluation, 2019, pp. 4–6.

[2] P. Bhattacharya, P. Mehta, K. Ghosh, S. Ghosh, A. Pal, A. Bhattacharya, P. Majumder, Overview of the FIRE 2020 AILA track: Artificial Intelligence for Legal Assistance, in: Proceedings of FIRE 2020 - Forum for Information Retrieval Evaluation, 2020.

[3] A. Mandal, K. Ghosh, A. Bhattacharya, A. Pal, S. Ghosh, Overview of the fire 2017 irled track: Information retrieval from legal documents., in: FIRE (Working Notes), 2017, pp. 63–68.

[4] J. Rabelo, M.-Y. Kim, R. Goebel, M. Yoshioka, Y. Kano, K. Satoh, A summary of the coliee 2019 competition, in: JSAI International Symposium on Artificial Intelligence, Springer, 2019, pp. 34–49.

[5] R. Rameshkannan, R. Rajalakshmi, Dlrg@ aila 2019: context-aware legal assistance system, Proceedings of FIRE (2019).

[6] Z. Zhao, H. Ning, L. Liu, C. Huang, L. Kong, Y. Han, Z. Han, Fire2019@ aila: Legal information retrieval using improved bm25., in: FIRE (Working Notes), 2019, pp. 40–45.

[7] R. More, J. Patil, A. Palaskar, A. Pawde, Removing named entities to find precedent legal cases., in: FIRE (Working Notes), 2019, pp. 13–18.

[8] S. Kayalvizhi, D. Thenmozhi, C. Aravindan, Legal assistance using word embeddings., in: FIRE (Working Notes), 2019, pp. 36–39.

[9] S. Mandal, S. D. Das, Unsupervised identification of relevant cases & statutes using word embeddings., in: FIRE (Working Notes), 2019, pp. 31–35.

[10] S. Renjit, S. M. Idicula, Cusat nlp@ aila-fire2019: Similarity in legal texts using document level embeddings., in: FIRE (Working Notes), 2019, pp. 25–30.

[11] T. F. Paulino-Passos, G., Retrieving legal cases with vector representations of text, Proceedings of the 6th Competition on Legal Information Extraction/Entailment. COLIEE (2019).

[12] T. A. El Hamdani, R., Coliee case law competition task 1: The legal case retrieval task, Proceedings of the 6th Competition on Legal Information Extraction/Entailment. COLIEE (2019).

[13] J. Rossi, E. Kanoulas, Legal information retrieval with generalized language models, Proceedings of the 6th Competition on Legal Information Extraction/Entailment. COLIEE (2019).

[14] A. Mandal, K. Ghosh, A. Pal, S. Ghosh, Automatic catchphrase identification from legal court case documents, in: Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, 2017, pp. 2187–2190.

[15] Y. Shao, Z. Ye, Thuir@ aila 2019: Information retrieval approaches for identifying relevant precedents and statutes., in: FIRE (Working Notes), 2019, pp. 46–51.

[16] V. Tran, M. L. Nguyen, K. Satoh, Building legal case retrieval systems with lexical matching and summarization using a pre-trained phrase scoring model, in: Proceedings of the Seventeenth International Conference on Artificial Intelligence and Law, 2019, pp. 275–282.

[17] C. D. Manning, H. Schütze, P. Raghavan, Introduction to information retrieval, Cambridge university press, 2008.

[18] J. Pennington, R. Socher, C. D. Manning, Glove: Global vectors for word representation, in: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), 2014, pp. 1532–1543.

[19] Q. Le, T. Mikolov, Distributed representations of sentences and documents, in: International conference on machine learning, 2014, pp. 1188–1196.

[20] T. Mikolov, K. Chen, G. Corrado, J. Dean, Efficient estimation of word representations in vector space, arXiv preprint arXiv:1301.3781 (2013).

[21] I. Chalkidis, M. Fergadiotis, P. Malakasiotis, N. Aletras, I. Androutsopoulos, Legal-bert: The muppets straight out of law school, arXiv preprint arXiv:2010.02559 (2020).