Legal Assistance using Word Embeddings

S. Kayalvizhi, D. Thenmozhi and Chandrabose Aravindan

Department of CSE, SSN College of Engineering, Chennai
{kayalvizhis,theni_d,aravindanc}@ssn.edu.in

Abstract. The legal counsellors will always look up on the prior cases and statutes to ensure justice. Referring all prior cases is a time consuming process since they are vast. Artificial intelligence can be made use to select the most relevant among all the documents. The existing systems have made use of different word embeddings, machine learning and deep learning methods to select the relevant ones. In our approach, different vectorization methods such as Word2Vec, Glove, Tf-Idf and count vectorizer are used and then similarity is calculated using Jaccard similarity and cosine similarity for ranking the prior cases and statutes. The work is evaluated on the AILA@FIRE-2019 dataset, in which it provides two tasks of finding the prior documents namely the relevant cases and statutes.

Keywords: Arifical Intelligence · Machine learning · Cosine similarity · Vectorization · Word Embeddings

1 Introduction

The population which needs to attain any legal assistance have to search for the prior cases and statutes for the current case. The search goes innumerable since there are many. Aiding the help of artificial intelligence for the search and retrieval seems to be a effective idea when compared to the manual search retrieval. Different machine learning and deep learning methods include Doc2Vec, Tf-Idf, LSTM , etc. are made use for retrieving the prior cases [2, 6, 8, 5, 7] . AILA@FIRE-2019 [3] proclaimed two tasks namely identifying relevant prior cases and relevant statutes. Identifying relevant prior cases refers to the retrieving similar prior cases for the given cases and identifying relevant prior statutes refers to the retrieving similar prior statutes for the given statute.

2 Proposed Methodology

2.1 Dataset Description

The AILA@FIRE-2019 dataset has 50 queries whose related statutes and prior cases has to be found out among given 197 statutes and 3000 prior cases. The first 10 queries are given the gold standard set which can be taken as training set and the remaining 40 queries are considered as test set.

For retrieving the relevant cases and statutes, initially the documents are vectorized using Word2Vec, Tf-Idf, count vectorizer and a ensembling method of Glove and Word2Vec. After vectorizing, the documents are all ranked by finding the similarity using cosine similarity and jaccard similarity.

2.2 Task 1: Identifying relevant cases

Word2Vec:
The query document and the case document are initially vectorized using Word2Vec model of dimension ’300’. After vectorizing the whole document, the max-min vector of the documents are considered to represent each document as a vector. Max-min vector of query is considered as ’a’ and ’b’ be the max-min vector of case document. The case documents are ranked by finding the cosine similarity [4] between a and b.

Ensembling Word2Vec and Glove:
In this method, the query and documents are all vectorized using both Word2Vec and Glove and then they are concatenated. Considering a single query and case document, vectorize the case document using Glove which forms the vector ’a1’ and vectorize the case document using Word2Vec model which forms the vector ’a2’. Concatenate the two vectors a1 and a2 which becomes the vector ’a’. The same process of vectorization is done for the query document which forms vector ’b’. Then, the case documents are ranked by finding the cosine similarity [4] between a and b.

Tf-idf vectorizer:
In this method, the documents and queries are all vectorized using Tf-Idf vectorizer. The entire corpus (queries and cases) is made use to form vocabulary which is used to fit the documents. The vector of query forms ’a’ and that of document forms ’b’. Rank the documents by finding the cosine similarity [4] between a and b.

2.3 Task 2: Identifying relevant statutes

Tf-idf vectorizer:
In this method, the documents and queries are all vectorized using Tf-Idf vectorizer. The entire corpus (queries and statues) is made use to form vocabulary which is used to fit the documents. The vector of query forms ’a’ and that of document forms ’b’. Rank the statute documents by finding the cosine similarity [4] between a and b.

Jaccard Similarity:
This method is done by finding out the Jaccard similarity. The vocabulary of query document forms ’a’ and vocabulary of statute document which forms ’b’. The documents are ranked by finding the jaccard similarity [1] between a and b.

Count vectorizer:

In this method, the documents and queries are all vectorized using count vectorizer (i.e) the count of each word in the documents (query and statute). Frequent words of the query document forms the dictionary of the file whose vector forms 'a'. Frequent words of the statute document forms the dictionary of the file whose vector forms 'b'. The statutes are ranked by finding the cosine similarity [4] between a and b.

3 Results

Table 1 shows the result of Task 1 of identifying the relevant cases with respect to the given case and Table 2 shows the results of Task 2 of identifying the relevant statutes. Different evaluation metrices like MAP, P@10, BPREF and 1/rank of the first relevant document have been used to evaluate the performance in which MAP score have been reported here.

From the results, Word2Vec vectorization seems to perform better than the other vectorization methods for identifying the relevant cases and Tf-Idf vectorizing method seems to be better for identifying the relevant statutes.

Teams and Runs	**MAP score**
SSN_NLP Run 1	**0.0405**
HLJIT2019	0.1492
Jiaming Gao	0.1382
Baban Gain	0.0984
TRDDC Pune	0.0956
Yunqiu Shao	0.0689
Sara Renjit	0.0481
Soumil Mandal - JU_SRM	0.0478
Kavya S Ganesh	0.0416

Table 1. Final evaluation for Test Data - TASK 1

4 Conclusion

Artificial Intelligence helps us in many ways for identifying the relevant documents among the entire prior documents in legal domain. Different word embedding methods of finding out similarity are experimented on ALIA@FIRE-2019 dataset. Various word embeddings like Word2Vec, Glove, ensembling Word2Vec and Glove, Tf-Idf vectorizer and count vectorizer are used for vectorization. After vectorization, cosine similarity is calculated to rank the documents. Among these Word2Vec seems to perform better than the other vectorization process for Task 1 of identifying the relevant cases and Tf-Idf seems to perform better than the other vectorizing methods for task 2 of identifying the relevant prior

Teams and Runs	**MAP score**
SSN_NLP Run 1	**0.077**
SSN_NLP Run 2	**0.061**
SSN_NLP Run 3	**0.051**
Yunqiu Shao	0.156
UBLTM	0.102
Sara Renjit	0.096
Soumil Mandal - JU_SRM	0.083
HLJIT2019	0.081
Kavya S Ganesh	0.068
Baban Gain - IITP	0.036

Table 2. Final evaluation for Test Data - TASK 2

statutes. The performance can be further improved by other machine learning and deep learning methods.

Acknowledgement

We would like to thank the Science and Engineering Research Board (SERB), Department of Science and Technology for funding the GPU system (EEQ/2018/000262) where this work has been carried out.

References

1. Achananuparp, P., Hu, X., Shen, X.: The evaluation of sentence similarity measures. In: International Conference on data warehousing and knowledge discovery. pp. 305–316. Springer (2008)
2. BarathiGaneshH., B., Kumar, M.A., Soman, K.P.: Distributional semantic representation for text classification and information retrieval. In: FIRE (2016)
3. Bhattacharya, P., Ghosh, K., Ghosh, S., Pal, A., Mehta, P., Bhattacharya, A., Majumder, P.: Overview of the FIRE 2019 AILA track: Artificial Intelligence for Legal Assistance. In: Proceedings of FIRE 2019 - Forum for Information Retrieval Evaluation (December 2019)
4. Huang, A.: Similarity measures for text document clustering. In: Proceedings of the sixth new zealand computer science research student conference (NZCSRSC2008), Christchurch, New Zealand. vol. 4, pp. 9–56 (2008)
5. Locke, D., Zuccon, G.: Automatic cited decision retrieval: Working notes of ielab for fire legal track precedence retrieval task. In: FIRE (2017)
6. Sandeep, G.V., Bharadwaj, S.: An extraction based approach to keyword generation and precedence retrieval: Bits pilani - hyderabad. In: FIRE (2017)
7. Thenmozhi, D., Kannan, K., Aravindan, C.: A text similarity approach for precedence retrieval from legal documents.
8. Tian, L., Ning, H., Kong, L., Han, Z., Xiao, R., Qi, H.: Hljit2017@irled-fire2017: Information retrieval from legal documents. In: FIRE (2017)