Distributed Representation in Information Retrieval - AMRITA_CEN_NLP@IRLeD 2017

Barathi Ganesh HB, Reshma U, Anand Kumar M and Soman KP
Center for Computational Engineering and Networking
Amrita University
Coimbatore, India
barathiganesh.hb@gmail.com,reshma.anata@gmail.com,m_anandkumar@cb.amrita.edu
soman_kp@amrita.edu

ABSTRACT

In this contemporary research era, the science of retrieving required information from the stored database is extending its applications in the legal and life science domains. With the exponential growth of the digital data available in the legal domain as an electronic media, there is a great demand for efficient and effective ways to retrieve required information from the stored document collection. This paper details our experimented approach in Information Retrieval from Legal Documents (IRLeD 2017) task. The task includes two subtasks, where the subtask 1 deals with information extraction and the subtask 2 deals with document retrieval. Text representation being a core component in any of the text analytics solution, we have experimented on the provided dataset to observe the performance of distributed representation of text in the Information Retrieval task. The distributed representation of text attained 3rd position in subtask 1 and attained satisfactory score in subtask 2.

CCS CONCEPTS

• **Information systems → Information retrieval**; • **Computing methodologies → Natural language processing**;

KEYWORDS

Information Extraction, Information Retrieval, Text Representation, Distributed Representation, Legal Documents, Catchphrase Extraction, Doc2Vec

1 INTRODUCTION

The success of Optical Character Recognition (OCR) and the availability of digital documents in legal domain, enforces the researchers to automate the process involved in legal domain. Among these processes Information Retrieval (IR) is a fundamental process [4], where in legal domain it helps in retrieving the prior cases related to the current cases (precedence retrieval) and it can act as a supporting reference to the legal practitioners [3].

In legal documents, more than the functional words (commonly used uninformative words), the frequency of the content words (domain dependent informative words) are high. The complex structure of these legal documents reduces the effectiveness of the representation as well as retrieval. Thus, by storing these documents with the meta data instead of the raw data will enhance the performance of the retrieval process. One such meta data are the catchphrases (list of legal terms) and these can be extracted through the Information Extraction (IE) task [3].

As told earlier this shared task involves two subtasks. The subtask 1 deals with the catch phrase extraction from the legal documents and subtask 2 deals with retrieval of documents related to the current case document from the prior case documents. [3].

Text representation is a principal component in any of the text analytics problem. This has the direct proportion with the performance of the system. Most of the current systems in the retrieval process follows the frequency based representation methods [1]. This is ineffective, when we need to retrieve the documents with respect its context. Representation of the context of the document is ineffective in the count based representation methods (Document - Term Matrix and Term Frequency - Inverse Document Frequency Matrix) and Distributional Representation methods (Count based representation followed by the Matrix Factorization) [1].

The cons stated above helped us to observe the performance of distributed representation in IR and IE. Here document to vector (doc2vec) is used to get the distributed representation of the documents and the phrases. For this experiment the data set has been provided by the Information Extraction for Legal Documents (IRLeD) shared task organizers[1]. On successive representation, we have utilized cosine distance for ranking the retrieved documents as well as extracted phrases. The remaining part of the paper discusses the distributed representation in Section 2 and the experiments, observations are detailed in Section 3.

2 DISTRIBUTED REPRESENTATION

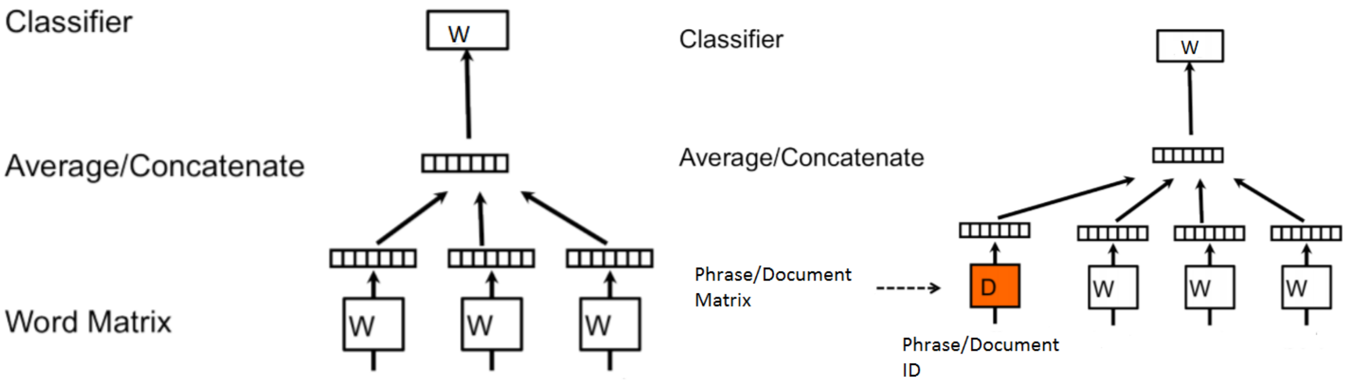
Though the Count based methods and Distributional Representation methods has ability to include the word's context through n-grams, it suffers from the selection of n-gram phrases, sparsity and curse of dimensionality [5]. To overcome the above stated cons, distributed representation is used to compute the fixed size dense vector representation of texts [2]. This representation method has the capability of representing the context of the text with variable-length into fixed size dense vector. The dimension of the vector is dynamic and typically its value ranges from hundred to thousand.

Word to Vectors (Word2Vec) is a framework for learning word vectors and it is shown in Fig 1a. The architecture is similar to the Auto Encoder, where input is the one hot encoded context words and output is one hot encoded target word to be predicted. The intermediate learning weights, maps context to the target to be predicted [2]. In Fig. 1a, the context of three words is mapped to

[1] https://sites.google.com/view/fire2017irled/track-description?authuser=0

Mean R Precision	Mean Precision @10	Mean Recall @100	Mean Average Precision	Overall Recall
0.168037101	0.1443333333	0.5352269964	0.1995772889	0.652431732

Table 1: Results for subtask 1



(a) Learning word vectors

(b) Learning phrase/document vectors

Figure 1: Distributed Representation

predict the fourth word by learning the matrix W. The column vectors in the matrix W is known as word embedding (dense word vectors).

Doc2Vec is a frame work for learning documents or sentence vectors and it is shown in Fig. 1b. The architecture is similar to the Word2Vec architecture shown in Fig. 1a. The only change is introducing a matrix D along with the matrix W to map the context words to the target words to be predicted [2]. Here concatenation or average of column vectors from D and W will be used to predict the target word. In Word2Vec the word itself act as the symbol to retrieve the corresponding vectors from the matrix W but in Doc2Vec a symbolic label will be assigned to each documents for the retrieval the corresponding vectors from the matrix D.

3 EXPERIMENTS AND OBSERVATIONS

Dataset for both the subtasks are provided by the Information Retrieval for Legal Documents (IRLeD) shared task organizers [3]. In subtask 1, we were provided with the 100 legal case document and its corresponding catch phrases for training. The objective is to extract catch phrases for 300 test documents and ranking them with respect to its relevance with the corresponding documents. The given training and test documents (400) are represented as vectors using Doc2Vec as explained in Section 2. In both the subtasks we have utilized Distributed Memory model for computing the document vectors. The file name of the documents are taken as the label for the documents. Similar to the documents, each catch phrase in the training documents are considered as the document itself and represented as a vector by assigning unique label. There is totally 98 unique catch phrases available in the given training set. This can be represented as,

$$d = \{d\_1, d\_2, \ldots, d\_400\} \tag{1}$$

$$D = doc2vec(\{D\_1, D\_2, \ldots, D\_400\}) \tag{2}$$

$$c = \{c\_1, c\_2, \ldots, c\_98\} \tag{3}$$

$$C = doc2vec(\{C\_1, C\_2, \ldots, C\_98\}) \tag{4}$$

In above equation D represents the document matrix, C represents the Catch Phrase Matrix, D_i represents the document vector and C_i represents the catch phrase vectors. On successive representation, we have computed the cosine distance between the catch phrase vector and the document vectors. Based on this cosine distances we have ranked the catch phrase for making final submission. The results are shown in following Table 1. For few of the application the basic count based methods performs better than the advanced representation methods. In order to observe the performance we experimented the same approach with the document - term matrix also.

In subtask2, the objective is to retrieve the relevant documents from the prior case documents by taking current documents as the query. We have been provided with the 2000 prior case documents and 200 current case documents. Both the documents sets are represented as a matrix through Doc2vec. This can be represented as,

$$prior = \{p\_1, p\_2, \ldots, p\_2000\} \tag{5}$$

$$Prior = doc2vec(\{P\_1, P\_2, \ldots, P\_2000\}) \tag{6}$$

$$current = \{c\_1, c\_2, \ldots, c\_200\} \tag{7}$$

$$Current = doc2vec(\{C\_1, C\_2, \ldots, C\_200\}) \tag{8}$$

In above equation $Prior$ represents the prior documents matrix, $Current$ represents the current documents matrix, P_i represents the prior document vector and C_i represents the current document vector. Similar to the subtask 1, here also cosine distance between the current and prior document vectors are measured and ranked. We have used cosine distance from python scipy pacakge[2]. The measured cosine distance given below,

$$distance = 1 - \frac{u \cdot v}{\|u\|_2 \, \|v\|_2} \tag{9}$$

In order to compare the different representation methods, we have experimented the same approach using Term Frequency - Inverse Document Frequency Matrix and Document - Term Matrix followed by a Singular value Decomposition. While computing SVD the reduced dimension is 200. The obtained results are shown in following Table 2.

Mean Average Precision	Mean Reciprocal Rank	Precision @10	Recall @100
0.0058	0.0145	0.0025	0.058

Table 2: Results for subtask 2

The Document - Term Matrix, Term Frequency - Inverse Document Frequency Matrix and Singular value Decomposition are computed using Scikit Learn python library[3]. The Doc2Vec is computed using Gensim python library[4]. In both the tasks Doc2Vec is computed using the parameters - $dimension = 50$, $minimumcount = 1$, $windowsize = 5$, $model = distributedmemory$. The recall should be higher for the real time application. In subtask 1, though the system attains less precision, it is able to attain the highest accuracy comparing other participated system.

4 CONCLUSIONS

The documents and the phrases provided by the organizers are represented as a matrix using distributed representation method. In subtask 1, n-grams are extracted and its relevance with the documents is measured using cosine distance. Similarly, in subtask 2 the relevance between current and prior documents are ranked based on the cosine distance.

This approach yields 3rd position in subtask 1 by attaining 0.199 as a Mean Average Precision and has also obtained highest overall recall (0.652) among the other participated systems. It has attained 0.0058 as a Mean Average Precision in subtask 2. The absence of gold-data in the training phase constrained to tune the system using hyper parameters in doc2vec. Hence the future work will be to focus more on developing a performance measurement method for unsupervised retrieval system.

REFERENCES

[1] Barathi Ganesh HB, Anand Kumar M, and Soman KP. 2016. Distributional Semantic Representation for Text Classification and Information Retrieval. (2016).

[2] Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*. 1188–1196.

[3] Arpan Mandal, Kripabandhu Ghosh, Arnab Bhattacharya, Arindam Pal, and Saptarshi Ghosh. 2017. Overview of the FIRE 2017 track: Information Retrieval from Legal Documents (IRLeD). In *Working notes of FIRE 2017 - Forum for Information Retrieval Evaluation (CEUR Workshop Proceedings)*. CEUR-WS.org.

[4] Mandar Mitra and BB Chaudhuri. 2000. Information retrieval from documents: A survey. *Information retrieval* 2, 2-3 (2000), 141–163.

[5] Peter D Turney and Patrick Pantel. 2010. From frequency to meaning: Vector space models of semantics. *Journal of artificial intelligence research* 37 (2010), 141–188.

[2] https://docs.scipy.org/doc/scipy-0.14.0/reference/generated/scipy.spatial.distance.cosine.html

[3] scikit-learn.org

[4] https://radimrehurek.com/gensim/