An Effective Approach of Extracting Local Documents from the Distributed Representation of Text using Doc2vec and LSA

Vikas Chib
Applied Machine Learning
Infineon Technologies
Bangalore,India
Vikas.chib@infineon.com

Ahsan Jafri
Applied Machine Learning
Infineon Technologies
Bangalore,India
Ahsan.jafri@infineon.com

Abstract: All neural embedding models learn distributed representation of text and match the results in the latent semantic space on a given query, but searching documents from the distributed representation will lose the relevance of local representation of a given query. We propose a novel information retrieval system, which uses doc2vec model to give top N similar documents with a relevant ranking using Latent Semantic Indexing to give the top K (documents score is greater than a soft threshold) documents which are the local representation of given query. We can use these K documents to find the most similar ones.

We can show that this 'dual' combination performs better than other traditional information retrieval algorithm or recently developed neural network models

Keywords: Information Retrieval, LSA (Latent Semantic Analysis), Doc2Vec, Embedding

I. INTRODUCTION

Information retrieval is the way to getting relevant information on the base of a search, it is the science of searching for information in a document, searching for images, sounds, metadata that describe data. Automating the information retrieval system will reduce the information overload, it provides access to journals, books, articles, and newspapers etc. There are mainly two challenges in the Information retrieval [1] process one is to defined the right search method, by which you can get a relevant material at the top of the search and the second one is the data warehouse is too big which contain a different branches of information, to find the right branch with a given query is one of a difficult task. In this paper, we are focusing on a second challenge to select a right information.

In the past, research on text extraction mainly used traditional methods like the Vector Space Model (VSM) [3] which are based on the frequencies of a word or TF-IDF weighting mechanism. The main drawback of traditional methods is word matrix cannot fully represent the meaning of a word. For example, if you search for apple, now you model doesn't know whether it is a fruit or company because the context is missing to overcome this Mikolov [5] who works in Google proposed an open source project named as Word2vec. It can map a word to a real vector and is considered as a perfect estimation of word representation in vector space. So, we can use a vector which represents a word better. In 2014, Le and Mikolov proposed Doc2vec as an extension to Word2vec to learn document embedding. The concept behind is Doc2vec is same as Word2vec, the only difference has they added another vector (paragraph ID). So, when training the word vectors W, the document vector D is trained as well and after the training it holds vector representation of a document, but this embedding learns distributed representation of text which is good when you have to do clustering, classification [2] but it Information retrieval task it is not able to map local representation of a given query. One easy method to overcome this is to keep your source data limited which will be relevant to your domain but this always limits the scope of your application.

Our work is to combine the importance of LSA [4] and Doc2vec to get the relevant documents at the top of a search.

II. LSA + DOC2VEC

A. *Vector Space Model*

The most important challenge of document categorization is understanding the contents of documents [7]. The simple method is to use several words instead of the whole document to represent the content of it. But limited words cannot express the full meaning of documents. So, to overcome this problem we use document or word as a vector and in vector space, every word or document can be quantified and compared.

Vector space models is a collection of vectors, where these vectors can be added or multiplied. In vector space, a term by document matrix is made and the rows of which are words, the columns of which are documents. So, a document is represented as a vector, and each element of a vector always is a value related to the words in this document. TF-IDF weighting is a general way to find the value of the matrix element.

In the case of document-query, a document can be represented as a vector where the vector dimensions refer to the terms available in the document. Dimension's value is an occurrence of a term inside a document. A vector representation of document can be written as:

$$\vec{a} = (a_1, a_2, a_3, \ldots) \qquad (1)$$

As same as the document, the query of term can be described as a vector form as:

$$\vec{b} = (b_1, b_2, b_3, \ldots) \qquad (2)$$

Based on vector similarity, similarity between two vectors is the dot product between the two vectors which is represented as $\cos\theta$:

$$\vec{a} \cdot \vec{b} = \|\vec{a}\|\|\vec{b}\| \cos\theta \quad (3)$$

$$\cos\theta = \frac{\vec{a} \cdot \vec{b}}{\|\vec{a}\|\|\vec{b}\|} \quad (4)$$

B. Latent Semantic Analysis

In VSM, frequencies of words appear in a document are considered as the feature of documents, but it cannot express the information at a semantic level. For example, one document contains "autonomous" and second document contain "vehicle". According to VSM both words are different, it categorize both document into two parts but they assemble same meaning.

C. Doc2Vec

In this case, a document is what you make of it, be it a sentence, a paragraph, an article, an essay, and so on. This is the Document Vector model analogous to Word2Vec [6] CBOW. The doc-vectors are obtained by training a neural network on the synthetic task of predicting a center word based an average of both context word-vectors and the full document's doc-vector.

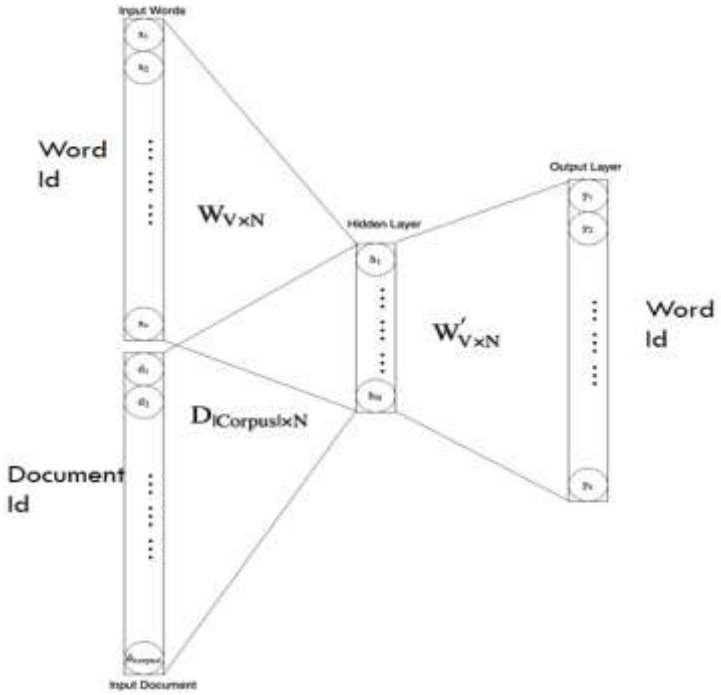

Fig. 1. Doc2vec and word2vec feature representation

III. Experiments

We conducted multiple sets of experiments on crunchbase dataset and perform a test on various document retrieval techniques.

A. Datasets for Experiments

To evaluate the performance of a new framework, we used crunch base dataset 2015 export version. We trained doc2vec with 400-dimensional vectors with window size 15 with 120 iterations. We test the performance of the model on LSA, doc2vec and on our new framework doc2vec.

We trained doc2vec model on a different hyperparameter and test its accuracy on a question-words dataset to get a relevant score. Below the graph is showing the improvement in the model.

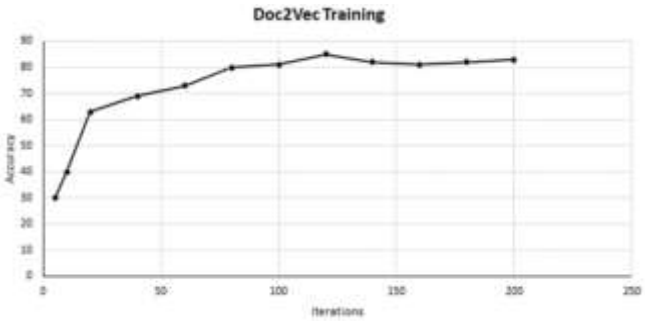

Fig. 2. Doc2vec training accuracy graph

B. Experiments of LSA + Doc2Vec Model

In our new LSA + doc2vec model, every document is transformed to a 2-dimensional vector through our new LSA + doc2vec model and it can be represented in full document content. This code is written in Python, and all our experiments are conducted on a computing farm with 28 cores and 200 GB RAM.

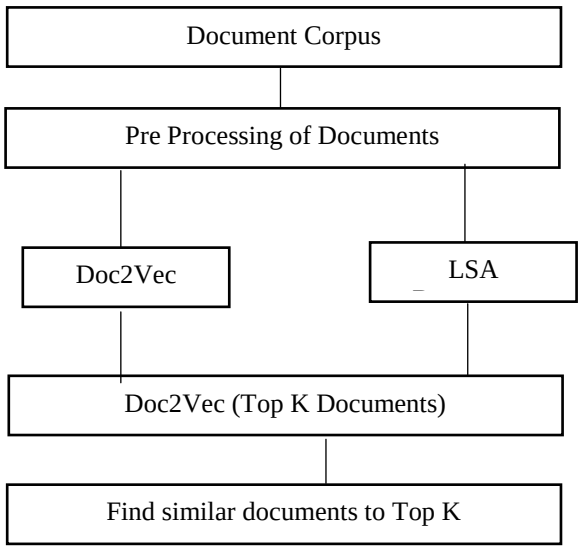

Fig. 3. LSA+Doc2vec architecture

We used the cosine similarity score to find the relevant document and compare the performance of model.

C. Analysis of Experiments

In the traditional models, VSM can only represent the frequencies of a word in document content and meaning, it cannot express the syntactic and contextual meaning. The classification accuracy of these model is 41.3 %. LSA is proposed to solve this problem and it executes SVD and reduces the dimensions on term and document relation matrix. Generated document vector will give the semantic meaning of the document, and it became more reliable. To some extent, it solved the problem and get the 44 % accuracy. Then we tried doc2vec model proposed by Mikolov in Google provides an efficient representation of docvecs and able to find the similarity between the documents and words, which helps us to get an accuracy of 61.5%

Model	**Accuracy (%)**
VSM	41.3
LSA	44.5
Doc2Vec	61.5
LSA + Doc2Vec	78.6

Fig. 4. Model comparison result

Our new LSA+doc2vec model combines the advantage of LSA, word2vec, and doc2vec. Combining these models, where we will get a top document using doc2vec and rank those documents using LSA, will help us to get the local context of the search. Then, we will find similar documents to above the top K document, which will result into a good accuracy of 78.6 % which is higher than the above-mentioned techniques

ACKNOWLEDGMENT

This research is supported by Infineon Technologies.

REFERENCES

[1] W. Lam, M. Ruiz, and P. Srinivasan, “Automatic Text Categorization and Its Application to Text Retrieval”, IEEE Transactions on Knowledge and Data Engineering, vol. 11, pp. 865-879, 1999.

[2] L. Wu, Z. Li, M. Li, W. Y. Ma, and N. Yu, “Mutually Beneficial Learning with Application to On-line News Classification”, In Conference on Information and Knowledge Management Archive Proceedings of the ACM First Ph.D. Workshop in CIKM, pp. 85-92, 2007.

[3] G. Salton, A.Wong, and C.S. Yang, “A Vector Space Model for Automatic Indexing”, Communications of the ACM, vol. 18, pp. 613-620, 1975.

[4] S. T. Dumais, “Latent semantic analysis”, Annual review of information science and technology, vol. 38, pp. 188-230, 2004.

[5] T. Mikolov, K. Chen, G. Corrado, and J. Dean, “Efficient Estimation of Word Representations in Vector Space”, In Proceedings of Workshop at ICLR, arXiv preprint arXiv:1301.3781, Oct. 21, 2013.

[6] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, “Distributed Representations of Words and Phrases and their Compositionality”, Advances in NIPS, pp. 3111-3119, 2013.

[7] C. H. Li, W. Song, and S. C. Park, “An Automatically Constructed Thesaurus for Neural Network Based Document Categorization”, Expert Systems with Applications 36, Elsevier, pp. 10969-10975, 2009.