A Text Retrieval System Based on Distributed Representations

Zhe Zhao, Tao Liu(✉), Jun Chen, Bofang Li, and Xiaoyong Du

School of Information, Renmin University of China, Beijing, China
{helloworld,tliu,chenjun2013,libofang,duyong}@ruc.edu.cn

Abstract. Most text retrieval systems are essentially based on bag-of-words (BOW) text representations. Despite popularity of BOW, it ignores the internal semantic meanings of words since each word is treated as an atomic unit. Recently, distributed word and text representations become increasingly popular in NLP literatures. They embed syntactic and semantic information of words and texts into low-dimensional vectors, thus overcome the weaknesses of traditional BOW representations to some extent. In this paper, we implement a text retrieval system that are totally supported by distributed representations. Our new system no longer relies on the matchings of words in queries and texts, but uses semantic similarity to judge if a text is relevant to a query and to what extent, which provides better user experience compared with traditional text retrieval systems.

Keywords: Text retrieval · Distributed text representation · Hierarchical paragraph vector

1 Introduction

Bag-of-words (BOW) is a simple but surprisingly powerful approach for text representation. Nowadays mainstream text retrieval systems such as Lucene[1] still rely on BOW representations. In traditional text retrieval systems, the relevance of query and text is essentially based on words matchings. And the heuristic weighting schemes and smoothing techniques are usually required to obtain better performance [3]. In our new text retrieval system, all objects (include word, text, etc.) are represented by low-dimensional vectors, each dimension of which represents high-level semantics rather than the occurrence of concrete word. The degree of relevance between query and text is measured by distance between their distributed representations.

Paragraph Vector (PV) is a popular approach for generating distributed text representations (embeddings), where text embeddings are trained to be useful to predict words in the texts [1]. PV models are trained in unsupervised framework. However, in many cases, label (supervised) information is available such

[1] http://lucene.apache.org/.

F. Li et al. (Eds.): APWeb 2016, Part II, LNCS 9932, pp. 537–541, 2016.
DOI: 10.1007/978-3-319-45817-5_60

as the news categories and hashtags in Twitter. To exploit this kind of knowledge, we extend traditional PV by adding label layer upon it, which is called Hierarchical Paragraph Vector (HPV). HPV regards each label as a pseudo text and its 'words' is the texts that process the corresponding label. As a result, the framework of HPV is illustrated in Fig. 1. HPV takes supervised information into consideration and can generate better text representations. When labels are not available, traditional PV is used for training in our system.

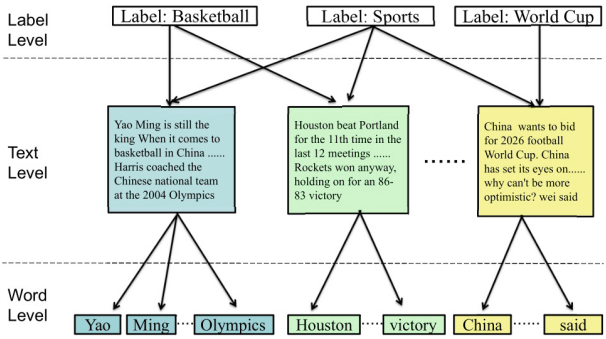

Fig. 1. Framework of hierarchical paragraph vector. Arrows denote 'predict'. Labels are treated as pseudo texts and their embeddings are trained to predict texts.

2 Overview Framework

Our system consists of two components: (1) Off-line component, where words, texts and labels are embedded into low-dimensional vectors. (2) On-line component, where queries are embedded into low-dimensional vectors and compared with trained text embeddings to return the ranked texts list. The framework of the system is illustrated in Fig. 2.

A. Off-line component: Firstly, word embeddings and parameters in neural network are pre-trained by external large-scale datasets such as Wikipedia. This component is included mainly for two reasons: (1) word embeddings trained in external large-scale datasets are 'universal' features, which already capture syntactic and semantic information of words well and can be used for many NLP tasks directly. (2) when the query is composed of words that never occur in our dataset (a very common case), directly utilizing word embedding trained in external large-scale dataset can provide satisfying results.

Texts and their labels are then fed into Hierarchical Paragraph Vector model. Embeddings and parameters in neural model are fine-tuned to make labels and texts to be useful to predict their corresponding texts and words.

B. On-line component: When a query is provided, we embed it into the same semantic space with text embeddings. Specifically, an embedding is initialized randomly and trained to be useful to predict the words in the query.

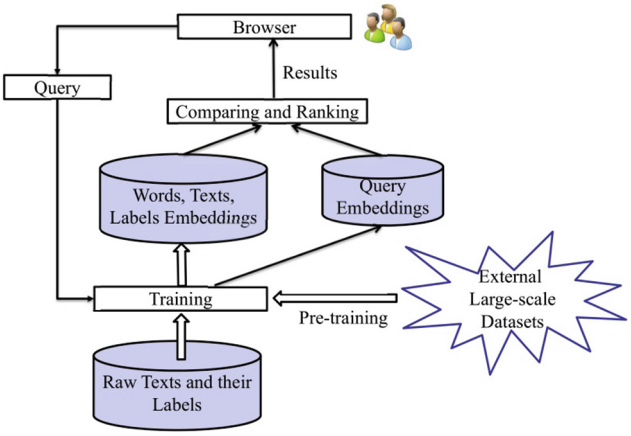

Fig. 2. System overview

During the training, parameters in the model are fixed. When training process is finished, the query embedding is compared with texts embeddings and top k texts are retrieved for users.

3 Key Technologies

In this section, we give a detailed discussion about HPV. Firstly some notations are established. Texts collection is denoted by $C = \{t_1, t_2, ..., t_{|C|}\}$ and i_{th} text in C is denoted by $t_i = \{w_{i1}, w_{i2}, ..., w_{i|t_i|}\}$. V is vocabulary set and $|WN|$ is the number of words in the whole collection. Local context of word w is denoted by $w^{context}$. Label set is denoted by $L = \{l_1, l_2, ..., l_{|L|}\}$. T_i is the collection of texts that possess label l_i. The training objective of our system consists of four components:

$$\begin{aligned} &\sum_{i=1}^{|C|} \sum_{j=1}^{|t_i|} logP(w_{ij}|t_i) + \sum_{i=1}^{|WN|} logP(w_i|w_i^{context}) \\ &\quad + \sum_{i=1}^{|L|} \sum_{t_k \in T_i} logP(t_k|l_i) \\ &\quad + \sum_{w_k \in V} Sim\_reg(w_k) \end{aligned} \tag{1}$$

where conditional probability $P(|)$ is defined by negative sampling softmax [2]. The first two components is just the objective of PV, where target words are predicted by texts and local contexts respectively [1]. The third component introduces supervised information into the models by maximizing the conditional probabilities of texts given their labels. By taking the label information into consideration, we can not only improve the quality of text embeddings, but also obtain the trained label embeddings.

In the last part of objective, *Sim_reg(w)* is used to denote the similarity between embeddings of word w trained in our dataset and pre-trained by external large-scale corpus. This component can be viewed as a regularization term which prevents the model from over-fitting our dataset. This strategy is very effective when handling small-scale dataset. For medium or large scale dataset, this part can be discarded.

4 Demonstration

We demonstrate scenarios where our system can provide better user experience than traditional text retrieval systems. We input the query 'Kobe Bryant', a basketball player in NBA (as shown in Fig. 3). Traditional system can only retrieve those texts that contain input words. However, in our new system, the related texts that do not contain the query words can also be retrieved and ranked basically according to the semantic similarities with the query: News about NBA is ranked at relatively top positions, the following is sports news and in turn followed by news in other fields.

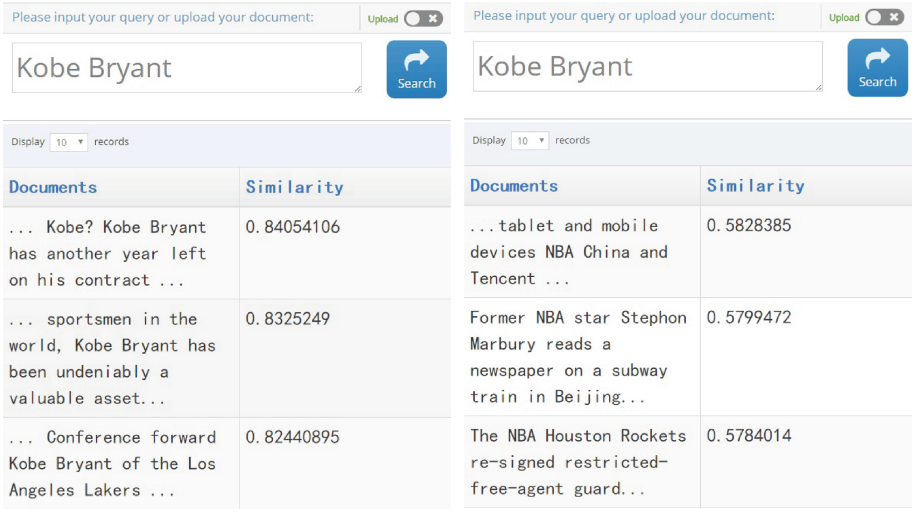

Fig. 3. The left snapshot shows the top 3 texts in response to the query. The texts that contain the key words are given very high similarities with query and are ranked at high positions. As we scroll down, our system returns texts that are related to the query but do not contain query words, which is shown in the right snapshot.

Acknowledgements. This work is supported by the Fundamental Research Funds for the Central Universities, the Research Funds of Renmin University of China No. 14XNLQ06.

References

1. Le, Q.V., Mikolov, T.: Distributed representations of sentences and documents. In: ICML 2014, Beijing, China, 21–26 June 2014, pp. 1188–1196 (2014)
2. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality, pp. 3111–3119 (2013)
3. Mogotsi, I.C., Christopher, D.M., Prabhakar, R., Hinrich, S.: Introduction to Information Retrieval, 482 p. Cambridge University Press, Cambridge, (2008). Inf. Retr. **13**(2), 192–195 (2010). ISBN: 978-0-521-86571-5