Exploring Labels Discriminating Power of Words in Label Embeddings for Text Classification

Jiayuan Xie

School of Guangdong University of Technology, Guangzhou 510000, China

jiayuan.xie@qq.com

Abstract

Recent studies have shown that simple models are efficient and interpretable in text classification tasks , and have the potential to outperform sophisticated deep neural models. Existing approaches use the information of the label to take into account the importance of the words in the document. In this paper, we propose LEAM-bdc. It handles the initialization of the label if the label is not pre-trained or desensitized. Our experimental results on the several large text datasets show that the proposed framework outperform the state-of-the-art method in text categorization tasks.

Keywords

Attention, label initialization, text classification.

1. Introduction

Text classification is one of the fundamental task in Natural Language Processing (NLP). The goal is to assign labels to document. A high-quality text representation can help classifiers to categorize documents more effectively and achieve a better performance. Traditional methods of text classification represent documents with lexical statistical features, such as Balanced Distributional Concentration (bdc) (Wang et al.,2015), and then use a linear model on this representation. The Balanced Distributional Concentration (bdc) method can find the most distinguishing words in each category.

Several studies have shown that the success of deep learning in text classification depends on the effectiveness of the word embeddings[1][2][3][4]. Particularly, Shen et al. (2018a) quantitatively show that the word-embeddings-based text classification tasks can have the similar level of difficulty regardless of the employed models, using the concept of intrinsic dimension (Li et al., 2018). Thus, simple models are preferred. Word embeddings as the basic building block of neural-based NLP, capturing the similarity/regularity between words. This idea has been extended to compute embeddings that capture the semantics of word sequences (e.g., phrases, sentences, paragraphs and documents). These representations are built upon various types of compositions of word vectors, ranging from simple averaging to sophisticated architectures. [3] show that the simple n-gram and maxpooling strategies can capture the order information in document. Further, they suggest that simple models are efficient and interpretable, and have the potential to outperform sophisticated deep neural. LEAM introduces label information as an attention mechanism based on SWEM and gets better results.

2. Relate work

Developing an expressive but computationally efficient combination function is a fundamental goal of NLP, which captures the linguistic structure of natural language sequences. Recently, some studies have shown that in some NLP applications, simpler word-embedding-based architectures exhibit comparable or even higher performance than more complex models that use recurrence or

convolutions [3][4]. Although complex compositional functions are avoided in these models, additional modules, such as attention layers, are employed on top of the word embedding layer. Label embedding has been shown to be effective in various domains and tasks, there has been a research on leveraging label embeddings to design efficient attention models [4]. In addition, recent studies have proven by experience that the advantages of different component functions depend to a large extent on specific tasks.

We need a text representation that captures the importance of words in document classification task and maintaining low computational cost. $LEAM-$bdc is similar to propose the Label-Embedding Attentive Model (LEAM) (Guoyin et al.， 2018).LEAM embed both words and labels a of document into a joint space i.e., $\Delta^D \rightarrow R^p$ and $\Upsilon \rightarrow R^p$. The label embeddings are $C = [c_1, \dots, c_k]$, where k is the number of classes. LEAM uses a s cosine similarity to measure the compatibility of one label and one word pairs :

$$G = (C^T V) \oslash G^{\wedge}$$

where $G^{\wedge}$ is a normalization matrix of size $K \times L$. Each element obtained by the multiplication between $l2\ norms$ of the $k-th$ label embedding and $l-th$ word embedding: $g_{kl} = \|c_k\|\|v_l\|$.

Then we get the weight of the word $\beta = \{\beta_1, \dots \beta_L\}$ through G. The document representation z_weight can be simply obtained by averaging the word embeddings, weighted by label-based attention scores:

$$\text{z\_weight} = \sum_{i=1}^{L} \beta_i * v_i \tag{1}$$

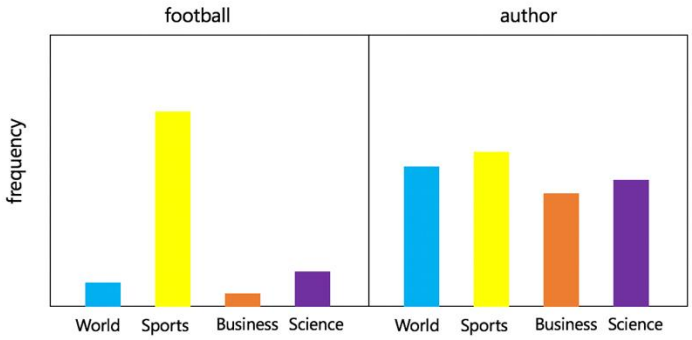

Figure 2

3.1 Label embedding

When calculating the matrix G, LEAM use 300-dimensional GloVe word embeddings Pennington et al. (2014) as initialization for word embeddings and label embeddings. The initialization of the label embedding has some impact on the matrix G, which in turn affects the weight of the word. Sometimes the words of the label are out-of-vocabulary or desensitized. $LEAM-$bdc propose a general method to initialize the label embedding.

The method of label initialization is to find the words that best represents the attribute of the category in the training set. For example, in the Agnews dataset, we will select the discriminative words in the training set of the 'sports' category, such as: 'football', 'baseball' etc. The label embedding initialization can be simply obtained by averaging the discriminative words embeddings. Figures 3 shows the word ‘football’ and the word ‘author’ frequencies in 4 categories

respectively. We see that the word 'football' mainly concentrates in a single category. Obviously, the word 'football’ has more discriminating power than the word ‘author’. Based on this observation, We believe that the word ‘football’ has a strong discrimination for the ‘sports’ category and can represent the label of ‘sports’ .$WOEM-weight$ can use the method to initialize the label and make our model achieve similar or even better results than the words of labels in the pre-training. $WOEM-weight$ use the term frequency-Balanced Distributional Concentration (tf-bdc)[5] to select the most distinguishing words.

The Balanced Distributional Concentration (bdc) method is based on the entropy principle and reflects the distribution of words in each category of the training set. We think that the more concentrated the terms appear in a certain category, such as 'football' in sports, the better the terms can represent the category and distinguish between the category and other categories. We consider the frequency of words, because some words happen to be concentrated only a few times in a certain category of the training set, so the bdc values is very big, but the word does not represent the category well. The term frequency (tf) is the number of occurrences of a word divided by the total number of words in the document. We calculate and rank the tf-bdc values of all words in the training set, and then select the words for each category. The tf-bdc is:

$$tf - bdc = tf * bdc$$

$$\mathrm{bdc}(t) = 1 - \frac{BH(t)}{\log(|C|)} = 1 + \frac{\sum_{i=1}^{|C|} \frac{p(t/c_i)}{\sum_{i=1}^{|C|} p(t/c_i)} log \frac{p(t/c_i)}{\sum_{i=1}^{|C|} p(t/c_i)}}{\log(|C|)}$$

$$p(t/c_i) = \frac{f(t, c_i)}{f(c_i)}$$

We select the top k (k = 1, 2, 3...) words with larger values of tf-bdc in each category, which can better distinguish the c_k category and other categories. Therefore, we believe that these k words can represent the category c_k, and the initialization of the label embedding of the category c_k is represented by the average of the word-embedding of the k words:

$$c_k \text{ initialization} = \frac{1}{k}\sum_{j=1}^{k} v_j$$

4.Experiment

4.1 Data sets

We evaluate the effectiveness of our model on five large scale document classification data sets. These data sets can be categorized into two types of document classification tasks: sentiment estimation and topic classification. The statistics of the data sets are summarized in Table 1. The content specified below:

sentiment estimation

Yelp Review Full: The dataset is obtained from the Yelp Dataset Challenge in 2015, the task is sentiment classification of polarity star labels ranging from 1 to 5.

Yelp Review Polarity: The same set of text reviews from Yelp Dataset Challenge in 2015, except that a coarser sentiment definition is considered: 1 and 2 are negative, and 4 and 5 as positive.

topic classification

AGNews: Topic classification over four categories of Internet news articles (Del Corso et al., 2005) composed of titles plus description classified into: World, Entertainment, Sports and Business.

DBPedia: Ontology classification over fourteen non-overlapping classes picked from DBpedia 2014 (Wikipedia).

We use 300-dimensional GloVe word embeddings [6]as initialization for word embeddings and label embeddings in our model. The out-Of-Vocabulary (OOV) words in document are initialized from a uniform distribution with range [−0.01, 0.01]. The out-Of-Vocabulary (OOV) words in labels are replaced by the words in document. The final classifier is implemented as an MLP layer followed by a sigmoid or softmax function depending on specific task. Adam is used to optimize all models, with an initial learning rate of 0.0002.Dropout regularizationis employed on the final MLP layer, with dropout rate 0.6. The batch size is 100. The model is implemented using Tensorflow and is trained on GPU 1080ti.

Tabel 2

World	Sports	Business	Science
government	team	company	microsoft
minister	win	prices	internet
president	game	stocks	internet

Table 3

	AGNews	DBPedia	Yelp F.	Yelp P.
K=3	92.67	99.005	63.64	95.347
K=4	92.61	99.023	63.65	95. 653
K=5	92.63	99.009	63.67	95.484
K=6	92.71	99.010	63.78	95.472
K=8	92.78	99.020	63.65	95.570
K=10	92.82	99.000	63.73	95.565
Random	92.48	98.900	63.67	95.474

Sometimes the words of the label are out-of-vocabulary or desensitized. We select the top k (k = 3,…,10) words with larger values of term frequency-Balanced Distributional Concentration (tf-bdc) in each category. These words can better distinguish between the certain category and other categories. As shown in Table 3, We have selected three words with the highest tf-bdc value for each category. We can observe that these words are similar to the meaning of labels.

We select the k range 3 from 10 , and then average the selected word-embeddings. The experimental results are shown in Table3. When the value of k is larger, the result of the corresponding experiment is better. The experimental results of our method are better than the results of random initialization of label.As the value of k is larger, the more words that represent the label are selected. The richer the information of the label. So the model can show better results.

References

[1] Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2016. Bag of tricks for efficient text classification. EACL.

[2] John Wieting, Mohit Bansal, Kevin Gimpel, and Karen Livescu. 2016. Towards universal paraphrastic sentence embeddings. ICLR.

[3] Dinghan Shen, Guoyin Wang, Wenlin Wang, Martin Renqiang Min, Qinliang Su, Yizhe Zhang, Chun-yuan Li, Ricardo Henao, and Lawrence Carin. 2018a. Baseline needs more love: On simple word-embedding-based models and associated pooling mechanisms. In ACL.

[4] Guoyin Wang, Chunyuan Li∗, Wenlin Wang, Yizhe Zhang Dinghan Shen, Xinyuan Zhang, Ricardo Henao, Lawrence Carin. 2018b. Joint Embedding of Words and Labels for Text Classification. In ACL.

[5] Tao Wang, Yi Cai, Ho-fung Leung, Zhiwei Cai and Huaqing Min.Entropy-based Term Weighting Schemes for Text Categorization in VSM .2015 IEEE 27th International Conference on Tools with Artificial Intelligence

[6] Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In Proceedings of the 2014 confer- ence on empirical methods in natural language pro- cessing (EMNLP), pages 1532–1543.