Distributional Representations of Words for Short Text Classification

Chenglong Ma, Weiqun Xu, Peijia Li, Yonghong Yan
The Key Laboratory of Speech Acoustics and Content Understanding
Institute of Acoustics, Chinese Academy of Sciences
No. 21 North 4th Ring West Road, Haidian District, 100190 Beijing, China
{machenglong,xuweiqun,lipeijia,yanyonghong}@hccl.ioa.ac.cn

Abstract

Traditional supervised learning approaches to common NLP tasks depend heavily on manual annotation, which is labor intensive and time consuming, and often suffer from data sparseness. In this paper we show how to mitigate the problems in short text classification (STC) through word embeddings – distributional representations of words learned from large unlabeled data. The word embeddings are trained from the entire English Wikipedia text. We assume that a short text document is a specific sample of one distribution in a Bayesian framework. A Gaussian process approach is used to model the distribution of words. The task of classification becomes a simple problem of selecting the most probable Gaussian distribution. This approach is compared with those based on the classical maximum entropy (MaxEnt) model and the Latent Dirichlet Allocation (LDA) approach. Our approach achieved better performance and also showed advantages in dealing with unseen words.

1 Introduction

With the boom of e-commerce and social media, short texts, such as instant messages, microblogs and product reviews, become more available in diverse forms than before. These short forms of documents have become convenient presentations of information. It is becoming more and more important to understand those short text documents and to efficiently detect what users are interested in. Unlike long documents such as news articles and blogs, it is hard to measure similarities among these short texts since they do not share much in common (Phan et al., 2008). This poses a great challenge to short text classification (STC).

The task of short text classification can be described as follows: given a short text *S*, the aim is to identify its target theme *T*. Several supervised learning approaches have been proposed for short text classification. They have been shown to be effective and yielded good performance. These approaches are effective because they leverage a large body of linguistic knowledge and related corpora. However, the supervised learning approaches depend heavily on manual annotation, which is labor intensive and time consuming, and often suffer from data sparseness.

To tackle the above problems, we exploit word embeddings. A word embedding $W{:}words \rightarrow R^n$ is a distributed representation for a word which is usually learned from a large corpus. Many researches have found that the learned word vectors capture linguistic regularities and collapse similar words into groups (Mikolov et al., 2013b).

In this paper, we apply an information theoretic approach which assumes that the short text is generated from a predefined parametric model, and estimate its optimal parameters from training data. We use Gaussian models to describe the distribution of words embeddings since it can describe any continuous distribution in common practice. Then, we classify new short texts using the Bayesian rule to get the posterior probability (Baker and McCallum, 1998).

The paper is organized as follows. Some related work is presented in Section 2. The word embedding

Proceedings of NAACL-HLT 2015, pages 33–38,
Denver, Colorado, May 31 – June 5, 2015. ©2015 Association for Computational Linguistics

based approach to short text classification is presented in Section 3. The dataset and evaluation metrics are described in Section 4. Experimental results on short text classification are given in Section 5. Some conclusions are drawn in Section 6.

2 Related Work

Learning to identify the theme of a short text document has been extensively studied during the past decade. Because the text length is short, data sparseness is an outstanding issue. Several approaches have been explored to overcome the data sparseness in order to get better performance.

Some try to calculate the similarity between short texts. E.g., (Zelikovitz and Hirsh, 2000) utilizes a corpus of unlabeled longer documents to compute the similarity between the test sample and the training one. To avoid collecting the specific longer documents, Web search engines (e.g. Google) are used to measure the similarity score (Bollegala et al., 2007; Yih and Meek, 2007) . But the efficiency of those approaches is a severe problem because they repeatedly queried search engines.

Some try to select more useful contextual information to expand and enrich the original text, e.g. using large unlabeled corpora, such as Wikipedia (Banerjee et al., 2007) and WordNet (Hu et al., 2009). A disadvantage of these approaches is that their adaptability would be an issue for certain languages because some of those external resources may be unavailable. Another approach is to integrate the context data with a set of hidden topics discovered from related corpora. E.g., (Phan et al., 2008; Chen et al., 2011) manually built a large and rich universal dataset, and derived a set of hidden topics through topic models such as Latent Dirichlet Allocation (LDA) (Blei et al., 2003) from these corpora. This approach has achieved satisfactory results, but it requires manual collection of the corpora. These researches have shown good improvement, but they rely too much on external resources which are difficult to get in some cases.

With the recent revival of interest in deep neural networks, many researchers have concentrated on learning a real-valued vector representation in a continuous space, where similar words are likely to have similar vectors. This is called word embedding (Turian et al., 2010). In fact, the learned word vectors capture linguistic regularities in a very simple way. In the embedding space, the vector offsets can measure specific relationship, such as the offset between *vec ("King")* and *vec ("Man")* is very close to that between *vec ("Woman")* and *vec ("Queen")* (Mikolov et al., 2013b).

3 Methodology

This section describes the proposed Gaussian classification approaches that use the learned word embeddings to model a classifier for the task of short text classification.

3.1 Word Representation

To get word representation, each input word token is transformed into a vector by looking up word embeddings learned from language model (Zeng et al., 2014). Distributed representations of words in word embedding space are shown to explicitly encode many syntactic and semantic regularities. Word embeddings have been used to help to achieve better performance in several NLP tasks (Collobert et al., 2011). There are some free tools for training word embeddings (Turian et al., 2010). We directly utilize *Word2Vec* tool provided by Mikolov et al. (Mikolov et al., 2013a) to train word embeddings on the Wikipedia corpus.

3.2 Our Approach

As mentioned in Section 3.1, all of the words are represented as word vectors. Word embeddings can be taken as an observation from an unsupervised generative model. We assume that a short text d_j is generated by theme t_k (parameterized by λ_k) according to the domain prior $p(t_k|\lambda_k)$. Similar to language modeling, we assume that a word embedding w_j^i for the i-th word in short text d_j depends only on the preceding words. Under this assumption, the probability of a document given theme t_k is,

$$p(d_j|t_k;\lambda_k) = \prod_{i=1}^{|d_j|} p(w_j^i|t_k;\lambda_k;w_j^m, m<i) \quad (1)$$

Next we assume that each word in a document is independent of its context, which is the same as that

for uni-gram language model. Then we rewrite equation 1 as

$$p(d_j|t_k;\lambda_k) = \prod_{i=1}^{|d_j|} p(w_j^i|t_k;\lambda_k) \quad (2)$$

Gaussian model is used to describe the distribution. We use the training data to estimate the parameters $\lambda_k = \{\mu_k, \Sigma_k\}$, where μ_k and Σ_k denote the mean vector and covariance matrix. We also assume that the covariance matrix of Gaussian is diagonal. λ_k can be estimated through Maximum Likelihood (ML) estimation as $\hat{\lambda}_k$:

$$\hat{\mu}_k = \frac{1}{|w_k|}\sum_{i=1}^{|w_k|} w_k^i \quad (3)$$

$$\hat{\Sigma}_k = \frac{1}{|w_k|}\sum_{i=1}^{|w_k|}(w_k^i - \hat{\mu}_k)(w_k^i - \hat{\mu}_k)^T \quad (4)$$

where $|w_k|$ is the total number of words in theme t_k on the training set, w_k^i is the i-th word.

Given estimates of the model parameters, new test data can be classified using the Bayesian theorem. A new short test text can be assigned the most likely theme as follows,

$$p(t_k|d_j;\hat{\lambda}_k) = \frac{p(t_k|\hat{\lambda}_k)\prod_{i=1}^{|d_j|} p(w_j^i|t_k;\hat{\lambda}_k)}{p(d_j|\hat{\lambda}_k)} \quad (5)$$

A uniform prior is used to choose the most probable theme which minimizes cross entropy on the test document. In equation 5, we drop the denominator (which is the same constant across all domains), and take the log of the entire expression. This results in

$$\sum_{i=1}^{|d_j|} \log(p(w_j^i|t_k;\hat{\lambda}_k)) \quad (6)$$

4 Dataset and Evaluation Metrics

To evaluate the performance of the above approach, we use the Web snippet dataset used in (Phan et al., 2008; Chen et al., 2011; Sun, 2012). The dataset contains 10,060 training and 2,280 test snippets of 8 domains, as shown in Table 1. The snippets of search results are (Phan et al., 2008), who collected various phrases belonging to different domains to query the web search engine (Google) and selected the top 20 or 30 snippets from the search results. Different phrases for the training and test data were used to make sure that test data were difficult to classify (Phan et al., 2008). The dataset has an average of 18 words in each snippet. Column 2 of Table 2 shows that the test data include about 4,378 words (about 43.62%) which do not appear in the training data. Column 3 shows the sizes of unseen words after Porter stemming (Sparck Jones, 1997). This table shows that there are more than 40% unseen words in the test data.

Domain	Training data	Test data
business	1,200	300
computer	1,200	300
cul.-arts-ent.	1,880	330
engineering	220	150
health	880	300
politics-soc.	1,200	300
sports	1,120	300
edu.-sci.	2,360	300
Total	10,060	2,280

Table 1: Statistics of the Web Snippets data

	Original	After stemming
Training Vocabulary	26,265	21,596
Test Vocabulary	10,037	8,200
Unseen Words	4,378	3,677
Difference (%)	43.62	44.84

Table 2: The number of unseen words

We downloaded the English Wikipedia dump of October 8, 2014, [1] which was used for training word embeddings. After removing all the non-roman characters and MediaWiki markups, we had 14,941,377 articles. The hyper-parameters used in *Word2Vec* are the same as that in (Mikolov et al., 2013a). To compare our results with the previous studies, we adopt accuracy as the performance metric, which is the proportion of the true results in the test output.

[1] Available at http://download.wikipedia.com/enwiki/.

Method	Feature	Classifier	Acc (%)
1	words (TF*IDF)	MaxEnt	65.75
2	words (TF*IDF) +topics	MaxEnt	82.18
Proposed	words (word embeddings, 400 dimensions)	Our Method	85.48

Table 3: Short Text Classification Performance

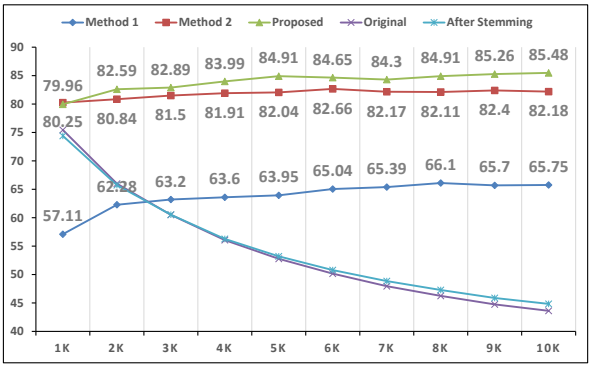

Figure 1: Evaluation with Different Sizes of Training Data.

5 Experiments

We conducted three sets of experiments. In the first set of experiments, we compare the performance of our approach with the previous studies. The second is to test the capability of our approach in dealing with the unseen words using different size of training data. The third is to investigate the effect of the word representation dimension on STC.

5.1 Comparison with Previous Work

For comparison, we select two approaches from (Phan et al., 2008) and the results are given in Table 3. The first method took the short text document as a bag of words (Salton, 1989) and used classical TF/IDF to represent the contribution of each term to its theme. In the second method, topic models are estimated from related corpus using LDA, then topics of the short text are inferred from those models. Thus, the features in method 2 contain topic distributions and bag-of-word vectors. The two approaches employ MaxEnt classifiers.

Table 3 illustrates the results for the three approaches. The best result is obtained from our proposed method with an absolute gain of 3.3 percent. It is clear that using word embeddings which were trained from universal dataset mitigated the problem of unseen words. Unlike the simple representations based on word frequencies (with some simplifications) (Clinchant and Perronnin, 2013) used in the previous studies, an important advantage is that our approach makes better use of the semantics from all the words in the short text document.

5.2 Dealing with Unseen Words

To validate the importance and influence of the size of training data in our approach, we increase the size of training data from 1,000 to 10,000 and measure the performance on the same test set. Since less training data will lead to more unseen words in the test phase, this experiment shows the capability in coping with unseen words, as shown the lines of Original and After Stemming in Figure 1. We directly cited the results of (Phan et al., 2008) because we could not crawl the related corpora which contained 3.5GB Wikipedia documents to re-implement their work.

The results of this experiment are shown in Figure 1. It can be seen that our approach based on the Gaussian process with word embeddings achieved good performance using relatively small data and reduced the cost of collecting and annotating training data.

5.3 The Effect of Word Representation Dimensions on STC

In our method, there is a free parameter in building word embeddings, i.e., the dimension of word representations. We empirically show the effect on the test data.

Figure 2 presents the short text classification performance obtained with different dimensions of word embeddings. In this section, we used all the training data as our experimental data. The best performance is about 85.83% when the size of word embedding space is 550 dimensions. The system achieves 7.23% absolute improvement when the di-

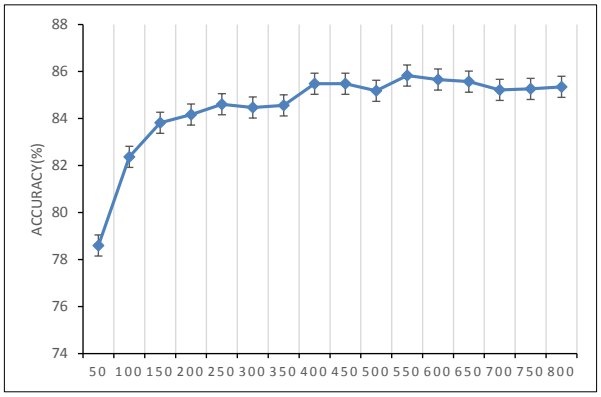

Figure 2: The effect of word embedding dimensions on STC performance.

mension of word embeddings increases from 50 to 550.

6 Conclusion

In this paper, we proposed to use Gaussian process with continuous word embeddings for short text classification. The experimental results show that our approach is effective and that the word embeddings capture syntactic and semantic relationships between words can make good contributions to handle unseen data. For future work, we would like to investigate how continuous word embeddings will work on other genres of short texts like microblogs or on conventional (long) texts, in topic and sentiment classification.

Acknowledgments

This work is partially supported by the National Natural Science Foundation of China (Nos. 11161140319, 91120001, 61271426), the Strategic Priority Research Program of the Chinese Academy of Sciences (Grant Nos. XDA06030100, XDA06030500), the National 863 Program (No. 2012AA012503) and the CAS Priority Deployment Project (No. KGZD-EW-103-2).

References

L Douglas Baker and Andrew Kachites McCallum. 1998. Distributional clustering of words for text classification. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 96–103. ACM.

Somnath Banerjee, Krishnan Ramanathan, and Ajay Gupta. 2007. Clustering short texts using wikipedia. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 787–788. ACM.

David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *the Journal of machine Learning research*, 3:993–1022.

Danushka Bollegala, Yutaka Matsuo, and Mitsuru Ishizuka. 2007. Measuring semantic similarity between words using web search engines. In *Proceedings of the 16th International Conference on World Wide Web*, WWW '07, pages 757–766, New York, NY, USA. ACM.

Mengen Chen, Xiaoming Jin, and Dou Shen. 2011. Short text classification improved by learning multi-granularity topics. In *Proc. of IJCAI 2011*, pages 1776–1781.

Stephane Clinchant and Florent Perronnin. 2013. Aggregating continuous word embeddings for information retrieval. In *Proceedings of the Workshop on Continuous Vector Space Models and their Compositionality*, pages 100–109, Sofia, Bulgaria, August. Association for Computational Linguistics.

Ronan Collobert, Jason Weston, Lon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *The Journal of Machine Learning Research*, 12:2493–2537.

Xia Hu, Nan Sun, Chao Zhang, and Tat-Seng Chua. 2009. Exploiting internal and external semantics for the clustering of short texts using world knowledge. In *Proceedings of the 18th ACM conference on Information and knowledge management*, pages 919–928. ACM.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013a. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, pages 3111–3119.

Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013b. Linguistic regularities in continuous space word representations. In *Proceedings of NAACL-HLT*, pages 746–751.

Xuan-Hieu Phan, Le-Minh Nguyen, and Susumu Horiguchi. 2008. Learning to classify short and sparse text & web with hidden topics from large-scale data collections. In *Proceedings of the 17th international conference on World Wide Web*, pages 91–100. ACM.

Gerard Salton. 1989. *Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer*. Addison-Wesley.

Karen Sparck Jones, editor. 1997. *Readings in information retrieval*. Morgan Kaufmann.

Aixin Sun. 2012. Short text classification using very few words. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*, pages 1145–1146. ACM.

Joseph Turian, Lev Ratinov, and Yoshua Bengio. 2010. Word representations: a simple and general method for semi-supervised learning. In *Proceedings of the 48th annual meeting of the association for computational linguistics*, pages 384–394. Association for Computational Linguistics.

Wen-Tau Yih and Christopher Meek. 2007. Improving similarity measures for short segments of text. In *Proceedings of the 22Nd National Conference on Artificial Intelligence - Volume 2*, AAAI'07. AAAI Press.

Sarah Zelikovitz and Haym Hirsh. 2000. Improving short text classification using unlabeled background knowledge to assess document similarity. In *Proceedings of the Seventeenth International Conference on Machine Learning*, volume 2000, pages 1183–1190.

Daojian Zeng, Kang Liu, Siwei Lai, Guangyou Zhou, and Jun Zhao. 2014. Relation classification via convolutional deep neural network. In *Proceedings of COLING*, pages 2335–2344.