LETTER *Special Section on Recent Advances in Machine Learning for Spoken Language Processing*

# Short Text Classification Based on Distributional Representations of Words

**Cheng long MA**[†a)], ***Nonmember***, **Qingwei ZHAO**[†], ***Member***, **Jielin PAN**[†], ***and*** **Yonghong YAN**[†,††], ***Nonmembers***

**SUMMARY** Short texts usually encounter the problem of data sparseness, as they do not provide sufficient term co-occurrence information. In this paper, we show how to mitigate the problem in short text classification through word embeddings. We assume that a short text document is a specific sample of one distribution in a Gaussian-Bayesian framework. Furthermore, a fast clustering algorithm is utilized to expand and enrich the context of short text in embedding space. This approach is compared with those based on the classical bag-of-words approaches and neural network based methods. Experimental results validate the effectiveness of the proposed method.

***key words:*** *short text classification, word embedding, gaussian model*

## 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 important to understand short text in order to efficiently detect the topic which users focus on. Unlike long documents, it is hard to measure similarities among these short texts since they do not share much in common [1]. 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 classify it to the predefined class label (or target theme $T$) according to its semantic information. To tackle the problem of data sparseness, we exploit word embeddings. Word embedding $W{:}words \rightarrow R^n$ is a distributed representation for a word which is usually learned from a large unlabeled corpus through unsupervised method. Many researches have found that the learned word vectors capture linguistic regularities and collapse similar words into groups [2]. Word embeddings can be used as features or as initializations of other neural networks [3]. Many existing works have shown that these can improve many semantic analysis tasks and yield state-of-the-art results.

In this work, we apply an information theoretic approach which assumes that the short text is generated from a predefined parametric model, which optimal parameters are estimated from training data. We apply Gaussian models to represent the distribution of word embeddings since it can describe continuous distributions in common practice. Then, we classify new short texts using the Bayesian rule to get the posterior probability [4]. Inspired by the idea of using external semantically related words [5], we discover the similar words by a fast clustering algorithm based on density peaks searching [6] in the embedding space. Then, the expanded short texts are fed into the proposed model. Evaluated on the benchmark of Web snippets [1], our proposed method achieves better performance than the existing state of the art methods.

## 2. Related Work

Learning to identify the theme of a short text document has been extensively studied during the past decade. Because of the short length, the data do not provide enough contextual information and the feature sparsity problem is an outstanding issue [1]. Thus, conventional text classification methods based on bag-of-words (BoW) cannot be directly applied to short texts [7]. Several approaches have been explored to overcome the data sparseness in order to get better performance.

One way is to select more useful contextual information to expand and enrich the original text, e.g. using large unlabeled corpora, such as Wikipedia [8] and WordNet [9]. Another way is to integrate the context data with a set of hidden topics discovered from related corpora. E.g., Phan et al. [1] and Chen et al. [10] manually built a large and rich universal dataset, and derived a set of hidden topics using Latent Dirichlet Allocation (LDA) [11]. These approaches have achieved satisfactory results, but they require manual collection of the corpora which are difficult to prepare in some cases.

With the recent revival of interest in deep neural networks, many researchers have concentrated on using deep learning to learn a real-valued vector representation in a continuous space [12]. Such methods for training word embeddings [13]–[16] are primarily based on the same distributional hypothesis: words that occur in similar contexts tend to have similar meanings [3]. One of the most widely used models for training word embeddings is the Skip-gram or continuous bag-of-words (CBOW) model [17]

Manuscript received January 30, 2016.
Manuscript revised April 8, 2016.
Manuscript publicized July 19, 2016.

[†]The authors are with the Key Laboratory of Speech Acoustics and Content Understanding, Institute of Acoustics, Chinese Academy of Sciences, China.
[††]The author is with the Xinjiang Laboratory of Minority Speech and Language Information Processing, Chinese Academy of Sciences, China.
a) E-mail: machenglong@hccl.ioa.ac.cn

DOI: 10.1587/transinf.2016SLL0006

because of their efficiency and simplicity. Furthermore, Le and Mikolov [18] presented the paragraph vector algorithm to learn continuous distributed vector representations for variable-length pieces of texts. Kim [19] applied convolutional neural networks (CNN) based on pre-trained and task-specific word vectors by having multiple channels to sentence-level classification tasks. Kalchbrenner et al. [20] described the Dynamic Convolutional Neural Network (DCNN) for modeling sentences. This network used Dynamic k-Max Pooling and induced a feature graph over the sentence that was capable of explicitly capturing short and long-range relations. Although the methods discussed above can capture complex features, they cannot guarantee the classification performance for very short texts. Unlike Wang et al. [5] that applied multi-scale semantic units to CNN, we propose a Gaussian method to tackle the short text classification.

## 3. Proposed Method

This section describes the proposed Gaussian-Bayesian framework that uses the learned word embeddings to model a classifier for the task of short text classification.

### 3.1 Word Representation

In the word representation component, each input word token is transformed into a vector by looking up word embeddings [12]. Each dimension of the vectors corresponds to a feature and might even have a semantic interpretation [21]. While many sophisticated approaches for building word vectors have been proposed, a comparison of the available word embeddings is beyond the scope of this paper. We directly utilize *Word2Vec* tool (which contains Skip-gram and CBOW model) provided by Mikolov et al. [17] to train word embeddings.

### 3.2 Short Text Expansion

Because of the length of short texts, they do not contain enough words to present the full semantic meaning. The simple method is to expand the coverage of the original short text. This is extremely useful to deal with future data that usually contain a lot of previously unseen features. In embedding spaces, words with similar meanings tend to have similar word embeddings. However, the vocabulary size of word embeddings is usually large. It is difficult to detect the similar words efficiently. Inspired by the idea of Wang et al. [5], we used a fast algorithm based on searching density peaks [6] to perform word embeddings clustering. In order to recognize precise semantic units [22], the semantic clique is firstly searched in the clustering groups and then the nearest words can be detected. Finally, we utilized each word in experimental data to detect its similar semantic results and expand them to the original short text. There was an average of 0.033 words which were added to each short text.

### 3.3 Our Approach

As mentioned in Sect. 3.1, all of the words are represented as word vectors. We assume that a short text $d_j$ is generated by theme $t_k$ according to the domain prior $p(t_k)$. 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,

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

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 Eq. (1) as

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

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

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

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

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,

$$p(t_k|d_j) = \frac{p(t_k) \prod_{i=1}^{|d_j|} p(w_j^i|t_k)}{p(d_j)} \tag{5}$$

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

$$\sum_{i=1}^{|d_j|} \log(p(w_j^i|t_k)) \tag{6}$$

## 4. Experiments

We conducted two sets of experiments. In the first set of experiments, we compared the performance of our approach with the previous studies. The second was to test the capability of our approach in dealing with the unseen words

**Table 1** Statistics of the web Snippets 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 2** Short text classification performance

| Method | Acc (%) |
|---|---|
| TF*IDF+MaxEnt [1] | 65.75 |
| TF*IDF+SVM | 66.1 |
| LDA+MaxEnt [1] | 82.18 |
| Multi-Topics+MaxEnt [10] | 84.17 |
| Paragraph Vector+SVMs [22] | 61.9 |
| LSTM [22] | 63.0 |
| Semantic-CNN [5] | 85.1 |
| CCNN [22] | 85.5 |
| **Proposed without expansion** | **85.48** |
| **Proposed** | **85.8** |

using different size of training data.

## 4.1 Dataset

To evaluate the performance of the proposed approach, we conducted experiments on the Web snippets dataset, as shown in Table 1. Each snippet has an average of 18 words.

We downloaded the English Wikipedia dump of October 8, 2014,[†] 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* were the same as that in Mikolov et al. [17]. To compare our results with the previous studies, we adopted accuracy as the performance metric, which was the proportion of the true results in the test output.

## 4.2 Comparison with Previous Work

**Baseline methods**. We compared our method with the following classification algorithms:

(1) TF*IDF+MaxEnt/SVM: We used the method of BoW to represent word and term frequency (TF) and inverse document frequency (IDF) to calculate the feature value. The classifiers of maximum entropy (MaxEnt) and support vector machines (SVM) were adopted.

(2) LDA+MaxEnt: In order to overcome the data sparseness, Phan et al. [1] used LDA to discover hidden topics from external Wikipedia corpus. Then, the expanded topics were integrated with the original short text and fed into MaxEnt classifier.

(3) Multi-Topics+MaxEnt: Based on the work of Phan et al. [1], Chen et al. [10] found that topics of certain granularity were usually not sufficient to set up effective feature spaces. Thus, topics at multiple granularity were used to expand the original short text.

(4) Paragraph Vector+SVMs: Inspired by *word2vec* [13], Le and Mikolov [18] represented a variable length of document by a fixed-length vector which was trained to predict words in the document. This paragraph vector was thought of as another general word and shared across all contexts generated from the same document.

(5) LSTM: Wang et al. [22] proposed a variation of the standard long short-term memory (LSTM) to model sentences. In their work, the model contained a single LSTM layer followed by an average pooling and a logistic regression layer. They changed the activation of cells output gate which did not depend on the memory cells state for efficient computation.

(6) Semantic-CNN: This method first discovered multi-scale semantic units by a fast clustering algorithm. Instead of the original short text, these meaningful units were combined and fed into convolutional layer, followed by max-pooling operation [5].

(7) CCNN: Based on the work of Semantic-CNN, Wang et al. [22] expanded short texts based on word embeddings clustering and used them to combine with the original words embeddings matrix. Finally, the expanded matrices were fed into the network of CNN.

With the same setup of experimental data, the comparisons of our method against the 7 baselines are given in Table 2. Baseline (1) is relatively weak because they ignore semantics of the words. Baselines (2) (3) are the traditional methods which take the short text document as a bag of words and use hidden topics to expand the original data. Semantic-CNN and CCNN perform better because they utilize word clustering method to enrich the representation of the short text. Unlike normal documents, short texts are usually noisier, less topic-focused, and much shorter. Through baselines (4) (5) are the state-of-the-art methods which have been shown effective in many semantic analysis tasks, the small length of short text without external information still heavily affects the classification performance. In general, our methods achieve the highest accuracy which show the effectiveness of the proposed model.

## 4.3 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 line of “unseen words” in Fig. 1.

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

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

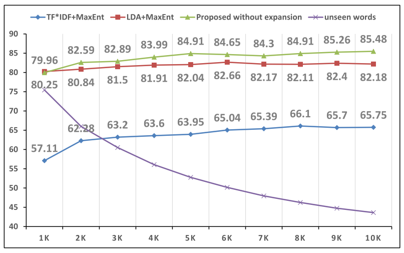

**Fig. 1** Evaluation with different sizes of training data.

## 5. Conclusion

In this paper, we proposed to use Gaussian-Bayesian model with continuous word embeddings for short text classification. Experimental results showed the effectiveness of the proposed approach. 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. 11461141004, 61271426, 11504406, 11590770, 11590771, 11590772, 11590773, 11590774), the Strategic Priority Research Program of the Chinese Academy of Sciences (Nos. XDA06030100, XDA06030500, XDA06040603), National 863 Program (No. 2015AA016306), National 973 Program (No. 2013CB329302) and the Key Science and Technology Project of the Xinjiang Uygur Autonomous Region (No. 201230118-3).

## References

[1] X.-H. Phan, L.-M. Nguyen, and S. Horiguchi, "Learning to classify short and sparse text & web with hidden topics from large-scale data collections," Proceedings of the 17th international conference on World Wide Web, pp.91–100, ACM, 2008.

[2] T. Mikolov, W.t. Yih, and G. Zweig, "Linguistic regularities in continuous space word representations," Proceedings of NAACL-HLT, pp.746–751, 2013.

[3] S. Lai, K. Liu, L. Xu, and J. Zhao, "How to generate a good word embedding?," arXiv preprint arXiv:1507.05523, 2015.

[4] L.D. Baker and A.K. McCallum, "Distributional clustering of words for text classification," Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval, pp.96–103, ACM, 1998.

[5] P. Wang, J. Xu, B. Xu, C.-L. Liu, H. Zhang, F. Wang, and H. Hao, "Semantic clustering and convolutional neural network for short text categorization," Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, pp.352–357, 2015.

[6] A. Rodriguez and A. Laio, "Clustering by fast search and find of density peaks," Science, vol.344, no.6191, pp.1492–1496, 2014.

[7] B. Sriram, D. Fuhry, E. Demir, H. Ferhatosmanoglu, and M. Demirbas, "Short text classification in twitter to improve information filtering," Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval, pp.841–842, ACM, 2010.

[8] S. Banerjee, K. Ramanathan, and A. Gupta, "Clustering short texts using wikipedia," Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval, pp.787–788, ACM, 2007.

[9] X. Hu, N. Sun, C. Zhang, and T.-S. Chua, "Exploiting internal and external semantics for the clustering of short texts using world knowledge," Proceedings of the 18th ACM conference on Information and knowledge management, pp.919–928, ACM, 2009.

[10] M. Chen, X. Jin, and D. Shen, "Short text classification improved by learning multi-granularity topics," Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence, pp.1776–1781, 2011.

[11] D.M. Blei, A.Y. Ng, and M.I. Jordan, "Latent dirichlet allocation," J. Machine Learning Research, vol.3, pp.993–1022, 2003.

[12] D. Zeng, K. Liu, S. Lai, G. Zhou, and J. Zhao, "Relation classification via convolutional deep neural network," Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers, pp.2335–2344, 2014.

[13] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," arXiv preprint arXiv:1301.3781, 2013.

[14] Y. Bengio, R. Ducharme, P. Vincent, and C. Janvin, "A neural probabilistic language model," J. Machine Learning Research, vol.3, pp.1137–1155, 2003.

[15] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, "Natural language processing (almost) from scratch," J. Machine Learning Research, vol.12, pp.2493–2537, 2011.

[16] A. Mnih and G. Hinton, "Three new graphical models for statistical language modelling," Proceedings of the 24th international conference on Machine learning, pp.641–648, ACM, 2007.

[17] T. Mikolov, I. Sutskever, K. Chen, G.S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," Advances in Neural Information Processing Systems, pp.3111–3119, 2013.

[18] Q. Le and T. Mikolov, "Distributed representations of sentences and documents," Proceedings of the 31st International Conference on Machine Learning, 2014.

[19] Y. Kim, "Convolutional neural networks for sentence classification," Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp.1746–1751, 2014.

[20] N. Kalchbrenner, E. Grefenstette, and P. Blunsom, "A convolutional neural network for modelling sentences," Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, pp.655–665, 2014.

[21] J. Turian, L. Ratinov, and Y. Bengio, "Word representations: a simple and general method for semi-supervised learning," Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, pp.384–394, 2010.

[22] P. Wang, B. Xu, J. Xu, G. Tian, C.-L. Liu, and H. Hao, "Semantic expansion using word embedding clustering and convolutional neural network for improving short text classification," Neurocomputing, vol.174, pp.806–814, 2016.