A News Headlines Classification Method Based on the Fusion of Related Words

Yongguan Wang, Binjie Meng, Pengyuan Liu(✉), and Erhong Yang

School of Information Science, Beijing Language and Culture University, Beijing, China
yongguanl992@163.com, mllrose@126.com, liupengyuan@pku.edu.cn, yerhong@blcu.edu.cn

Abstract. Short text classification is a challenging work as a result of several words, usually fewer than 20 words, in each text which brings about a problem of feature sparsity. In this paper, we propose a method of extending short text to cope with the problem of data sparsity. Additionally, we combine extension of short text, which forms a new representation with the word vector of each word in the short text trained by word2vec model on large-scale corpus. Furthermore, the new representation works as input for neural bag-of-words (NBOW) model. We evaluate this method on NLPCC 2017 Evaluation Task 2. The experimental results show that extension of short text extension with NBOW model outperforms baselines and can achieve excellent performance on the news headline classification task.

Keywords: Short text classification · NBOW · Text extension

1 Introduction

Short text classification is the task of automatically labeling short documents, such as news headlines and weibo blogs, which has numerous applications including topic categorization, sentiment analysis, or question answering. Until now, general machine learning methods, such as support vector machine (SVM), Naive Bayes Classifier (short text) and k-nearest neighbors (KNN), can be applied in text categorization and achieve desired performance. However, these methods based on BOW representation usually get unsatisfactory performance when processing short text features.

Short text, such as Chinese news headlines, usually consists of dozens of words, which provides limited contextual information and thus results in high sparsity in the text presentation. To solve this problem, the researchers put attention to text representation and classifier optimization. The existing methods in this field can be roughly classified into two groups. The first one is to expand short text features by utilizing external repositories (e.g. Wikipedia and web search-engine based). For example, Sahami and Heilman [1] calculated the short text document similarity and incorporated the extended short text, which is based on the web search results. Bollegala et al. [2] also added web search engine information as extended short text. Besides, web page ranking and lexical and syntax analysis was taken into consideration. Phan et al. [3] proposed a classification algorithm to extract useful information from Wikipedia. In his

X. Huang et al. (Eds.): NLPCC 2017, LNAI 10619, pp. 818–827, 2018.
https://doi.org/10.1007/978-3-319-73618-1_71

model, LDA is also used to obtain topic information from short text. Then he merged these feature words and topic information as expanded short text. However, the performance of these methods highly depends on whether the extended knowledge matches the short text.

Another group of methods consist of the deep neural networks based on language model, such as CNNs and RNNs. With the rapid development of deep learning, neural language models and their variations have achieved remarkable results on various classification tasks such as sentiment analysis and question-type classification. Kim [4] applied a simple CNN with a single convolution layer on top of pre-trained word vectors obtained from an unsupervised neural language model for sentence level classification. Despite little tuning of hypermeters, the simple model achieves excellent results on classification tasks. Kalchbrenner et al. [5] introduced a Dynamic Convolution neural networks (DCNN) model, which employed dynamic k-max pooling to keep sequential information. RNN [6] is able to deal with sequences of any length and capture long-term dependencies, and LSTM [7] can avoid the problem of gradient exploding or vanishing that occur in standard RNN. These two models and their variations [8–10] are often applied in classification tasks. In addition, Sentence representation is the basic work of text categorization. Sentence modeling [5, 11–13] is also related to our task.

In this paper, we introduce a model based on related filtering and extension of short text for short text classification. In order to make full use of semantic and syntactic information in the short text, we propose a method of short text extension. First of all, we find related words about each word in short text, which can easily be retrieved from the word vectors trained by word2vec [13] model on large-scale corpus. We treat the related words as an extension of each word, and the vector representation of each word and it's extension are merged into a new representation. The new representation is thought as the sentence feature vector, which is the input of the classification model. In this paper, NBOW model, CNN model and LSTM model are treated as three basic classification models. In addition, there are two different ways to filter the related words. One is to filter with the TFIDF values of the words, and the other is to calculate the word similarity based on word2vec model. These two methods are performed respectively in this paper, and the results manifest that combining TFIDF value with word similarity as a filter of the related words is the best one compared with other filters. And adding NBOW model as the classification model can get the best performance compared with other baseline models, such as CNN, LSTM and NBOW models in the headline classification task. Besides, the word vectors in three baseline models are initialized randomly.

2 NBOW Model of Fusion Related Words

In this section we will describe our model in detail. The model we proposed in this paper consists of two parts. The first part is the baseline model NBOW, whose main function is to learn text representation from input short texts and classify the texts. The other part is short text expansion, which is to obtain extension of the short text. In this part, we firstly train the word vectors by word2vec model, from which we calculate

distance (similarity) between each target word and other words. The related words are the nearest several words to the target words, which are then selected by the filter of word TFIDF value and similarity. After this, the related words after being selected form the extension of the short text, which is fused into the NBOW model to make classification. The model structure is shown in Fig. 1.

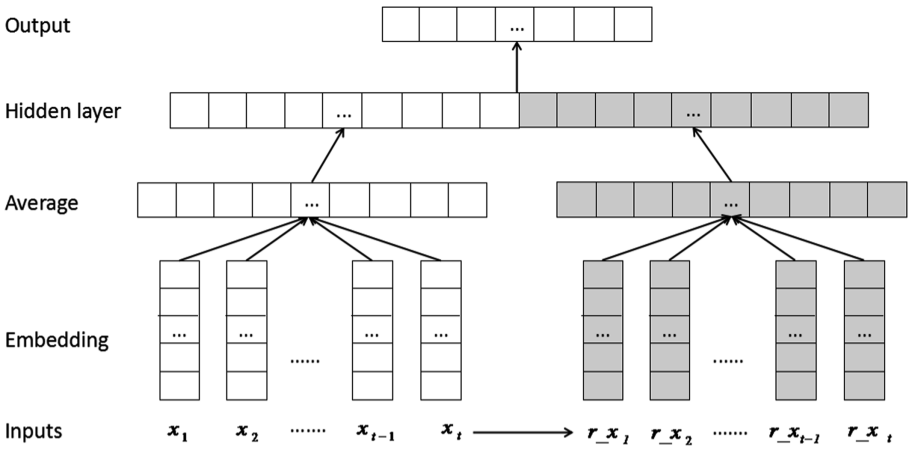

Fig. 1. Model architecture

2.1 Short Text Representation

A news title sentence of length t (padded where necessary) is represented as $\boldsymbol{x} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_t]$, where $\boldsymbol{x}_i$ stands for the d-dimensional word vectors of the i_{th} word in the sentence. In our model, we firstly input the t d-dimensional word vectors for the average operation, and get a d-dimensional word vector $\boldsymbol{z}_x$, which is the representation of the sentence. Then, the representation $\boldsymbol{z}_x$ of sentence is used as input of hidden layer, and the ReLU (Rectified Linear Units) function is the activation function which is to calculate the hidden layer output $\boldsymbol{h}_x$.

$$ReLU(x) = \begin{cases} x, x > 0 \\ 0, x \leq 0 \end{cases} \tag{1}$$

$$\boldsymbol{h}_x = ReLU(\boldsymbol{w} \cdot \boldsymbol{z}_x + \boldsymbol{b}) \tag{2}$$

2.2 Short Text Extensions

News headlines are short texts, which contain less semantic information and less features. When using the NBOW model for classification, its' performance is not as good as long text, so extending short text is a method to improve performance.

We propose a method of expanding the short text, which is treated as the input for classification model. In this paper, word2vec model is used to train the distributed

vector representation of words, and then the relevant words of the words in sentence are determined by the distance between the word and its target word. Given a news headline sentence $\boldsymbol{x}$, where $\boldsymbol{x_i}$ represents the i_{th} word. Find the m related words $\boldsymbol{r\_x_i} = \left[\boldsymbol{r\_x_i^{(1)}}, \boldsymbol{r\_x_i^{(2)}}, \ldots, \boldsymbol{r\_x_i^{(m)}}\right]$ of $\boldsymbol{x_i}$ through the pre-trained word2vec model, and calculate the similarity $\boldsymbol{sim\_x_i} = \left[\boldsymbol{sim\_x_i^{(1)}}, \boldsymbol{sim\_x_i^{(2)}}, \ldots, \boldsymbol{sim\_x_i^{(m)}}\right]$ of words pairs, where $\boldsymbol{sim\_x_i^{(j)}}$ stands for the cosine distance between the word $\boldsymbol{x_i}$ and the word $\boldsymbol{r\_x_i^{(j)}}$. Thus, we extend a sentence $\boldsymbol{x}$ whose length is t to obtain $t \times m$ related words $[\boldsymbol{r\_x}_1, \boldsymbol{r\_x}_2, \ldots, \boldsymbol{r\_x}_t]$.

2.3 Related Word Filtering

In order to reduce the noise in the extended text, the relative words of short text will be filtered. Firstly, the corpus data of training set is statistically calculated to calculate the TFIDF value of all words. For each input sample of sentence $\boldsymbol{x}$, a threshold τ is set for the TFIDF value. If the TFIDF value of some word in the sentence is smaller than τ, all the related words of this word are discarded. Then the word similarity is used to filter all the related words. The first way is to select top m related words by the similarity of word pairs for each word, and the other is to set the similarity threshold φ. If the similarity of one word is less than φ, the word will be filtered out. In our experiments, only the first is used after all these operations, we obtain the final related words $\boldsymbol{r\_x}$ of one sentence.

2.4 Fusion Related Words

For an input sentence $\boldsymbol{x}$, all related words $\boldsymbol{r\_x}$ are used as input, where $\boldsymbol{r\_x}$ is actually the word vector of all related words, obtained from the pre-trained word2vec model. First, the vector of all the related words are made an average operation. After that, we get a d-dimensional word vector $\boldsymbol{r\_z_x}$ which is related information of the sentence $\boldsymbol{x}$, and then we can get the output $\boldsymbol{r\_h_x}$ after going through one hidden layer. Then we connect the sentence representation $\boldsymbol{h_x}$ and $\boldsymbol{r\_h_x}$, and obtain a new sentence representation $\boldsymbol{h_x^{'}}$, which is a 2d-dimensional vector.

$$\boldsymbol{r\_h_x} = Relu(\boldsymbol{w} \cdot \boldsymbol{r\_z_x} + \boldsymbol{b}) \tag{3}$$

$$\boldsymbol{h_x^{'}} = \boldsymbol{h_x} + \boldsymbol{r\_h_x} \tag{4}$$

2.5 Classification

At the output layer, a full connection layer is applied to classify sentences after which output the classification results. The sentence representation vectors $\boldsymbol{h_x^{'}}$ is used as input and calculate the output $\boldsymbol{y}$. According to the real classification labels of training data, the model parameters are updated by back propagation algorithm.

$$\boldsymbol{y} = \text{softmax}\left(\boldsymbol{w} \cdot \boldsymbol{h}_{\boldsymbol{x}}^{'} + \boldsymbol{b}\right) \tag{5}$$

3 Experiments

3.1 Dataset

We conduct our experiment with the dataset of NLPCC 2017 Evaluation Task 2–Chinese News Headline Categorization, whose data released by NLPCC 2017 is collected from several Chinese news websites, such as toutiao, sina, and so on. In this task, most title sentence word numbers are less than 20, with a mean of 12.07. All the sentences are segmented by the python Chinese segmentation tool jieba, and the number of all categories in the dataset is 18. The complete dataset consists of training set, validation set and test set. The detailed information of each category and sub dataset is shown in Table 1. (The information of categories and sub datasets).

Table 1. The information of categories

Category	Train	Dev	Test
Entertainment	10000	2000	2000
Sports	10000	2000	2000
Car	10000	2000	2000
Society	10000	2000	2000
Tech	10000	2000	2000
World	10000	2000	2000
Finance	10000	2000	2000
Game	10000	2000	2000
Travel	10000	2000	2000
Military	10000	2000	2000
History	10000	2000	2000
Baby	10000	2000	2000
Fashion	10000	2000	2000
Food	10000	2000	2000
Discovery	4000	2000	2000
Story	4000	2000	2000
Regimen	4000	2000	2000
Essay	4000	2000	2000

3.2 Settings

We implemented our model based on Tensorflow—an open source software library for numerical computation using data flow graph. We initialized the word vectors with the dimension of 200, which was pre-trained by word2vec [11] on several of corpus, such

as Wikipedia documents, sougou news corpus and People's Daily corpus. The embedding layers in our model requires fixed-length input, we define 40 to represent the maximum length of sentence in the datasets, and pad zero at the end of the last word in the sentence whose length is shorter than max length. For all neural network models, batch size is initialized to a value of 64. Besides, the number of filter in CNN model, which is also called window, is initialized with 128. And to capture different syntactic and semantic information, the size of filter is 1, 2 and 3. As for LSTM model, we set 300 as hidden size.

3.3 Model Variations

In the experiment, we tested our approach based on the three baseline models, and compared the results of different models. Explanations for different model representations are as follows (The representation of other models is similar to that of CNN).

- CNN-w2v: Using CNN model for classification and using pre-trained vectors from word2vec.
- CNN-w2v-extension: Using CNN model for classification and using pre-trained vectors from word2vec, extending input sentences (News headlines) by related words. For each word, we set the number of extensions m to 10.
- CNN-w2v-extension-filters: Using CNN model for classification and using pre-trained vectors from word2vec, extending input sentences (News headlines) by related words. In addition, using the TFIDF value to filter the related words.

3.4 Results and Discussion

By comparing the baseline model (without pre-trained word vector) and baseline+ word2vec model, we can find that the accuracy of using pre-trained word vector as the original input for news headlines classification is higher than that of the original input using the random initialization of the word vector. The reason is that the word vectors trained by the word2vec tool contain contextual semantic information, so they can get better results in text classification. Then, comparing NBOW-w2v-extension model and NBOW-w2v model, the results show that it is effective to classify short texts after expansion, and can obviously improve classification performance. However, the use of TFIDF filtering related words has a slight increase in NBOW models, but other models do not improve performance. We argue that this is because the expansion scale of news heading sentences in our experiments is not large. Therefore, the noise introduced by this cannot have a strong impact on classification. In addition, most neural network models have the ability to automatically extract features from inputs, which are in fact similar to filter functions. So, in our experiments, adding filters does not significantly improve classification performance (Tables 2 and 3).

Table 4 shows the results of misclassification of the NBOW-w2v-extension-filter model. We can find by analysis that the category information contained in some news headlines is ambiguous. For example, the word "营养价值" in sentence 3 means that the sentence may belong to the category of regimen, and another word "吃" in the same sentence means that it should belong to the category of food. So we infer that without

Table 2. Accuracy of different models

Group	Model	Accuracy%
CNN	CNN	59.54
	CNN-w2v	69.13
	CNN-w2v-extension	80.62
	CNN-w2v-extension-filter	78.46
LSTM	LSTM	74.70
	LSTM-w2v	79.20
	LSTM-w2v-extension	80.13
	LSTM-w2v-extension-filter	80.06
NBOW	NBOW	78.30
	NBOW-w2v	80.09
	NBOW-w2v-extension	81.10
	NBOW-w2v-extension-filter	**81.34**

Table 3. The test result of NBOW-w2v-extension-filter model

Tag	Precision	Recall
<unk>	0.0000	0.0000
History	0.8219	0.8535
Military	0.8377	0.8565
Baby	0.8250	0.8865
World	0.7131	0.7095
Tech	0.8094	0.8240
Game	0.8962	0.8760
Society	0.5727	0.6205
Sports	0.8931	0.8895
Travel	0.7334	0.8170
Car	0.8941	0.8825
Food	0.8200	0.8790
Entertainment	0.7375	0.7825
Finance	0.8090	0.8195
Fashion	0.8037	0.8150
Discovery	0.9372	0.8575
Story	0.8585	0.7435
Regimen	0.8909	0.7515
Essay	0.8553	0.7775
Overall	0.8134	0.8134

news text, a news headline sentence can belong to several categories at once. This led to the difficulty of classifying them correctly. Table 5 shows the extension of a news headlines. For example, in the absence of extensions, the sentence "特朗普上台，美国会减弱针对中俄军事压力？" classification results are world,

This is a wrong result. After the sentence was extended, we got some words, such as "军队","反恐" and so on. These words can be used to expand the features of military categories. However, when extending a sentence, it may extend to some words related to another category. Therefore, when a sentence contains different categories of information, how to choose the right word is a problem to be solved.

Table 4. Examples of misclassification of NBOW-w2v-extension-filter

	News headline	Classification result	Correct result
1	你是我心头的痛——记《安生与七月》	essay	story
2	你本来就拥有的5大奢侈品	fashion	baby
3	它的营养价值很高，但有很多人却不敢吃	food	regimen

Table 5. The comparison of classification results between NBOW models

Input text	Extended words	Classification result (no extension)	Classification result (extension)	Correct result
不只是地球，其他星球也下雪！还有粉红色的雪呢！	…月球 火星 星球 行星 冥王星 宇宙 地球.表面 太阳系 外太空 人类系 国度 银河系 月球 下雨 有雪 大雪 刮风 落雪…	travel	discovery	discovery
特朗普上台，美国会减弱针对中俄军事压力？	罗姆尼 希拉里 麦凯恩 奥巴马 共和党 杜特蒂 萨科齐 美国共和党 国防 政治 军队 反恐 军事力量 外交 网络战 武装力量 防务 军备…	world	military	military

4 Conclusion

In this paper, we propose a model based on neural bag of words (NBOW) on top of word2vec, which merges with related words and TFIDF filters for news headlines classification. We have described a series of experiments with convolution neural networks (CNN), long-short-term memory network (LSTM) and neural bag of words (NBOW) on top of word2vec. And we have demonstrated that our model performs best among all the other models. In particular, the related words are a good supplement of syntactic and semantic information for classification, which also solve the problem of data sparsity to some degree.

Our model can be applied to sentence modeling as well as other natural language processing tasks. Future works can modify our model to complete long text classification tasks.

Acknowledgment. This research project is supported by Science Foundation of Beijing Language and Culture University (supported by "the Fundamental Research Funds for the Central Universities") (17PT05); Supported by Major Project of the National Language Committee of the 12th Five-Year Research Plan in 2015 (No. ZDI125-55)

References

1. Sahami, M., Heilman, T.D.: A web-based kernel function for measuring the similarity of short text snippets. In: Proceedings of the 15th International Conference on World Wide Web, pp. 377–386. ACM (2006)
2. Bollegala, D., Matsuo, Y., Ishizuka, M.: Measuring semantic similarity between words using web search engines. WWW **7**, 757–766 (2007)
3. Phan, X.H., Nguyen, L.M., Horiguchi, S.: 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. ACM, New York, pp. 91–100 (2008)
4. Kim, Y.: Convolutional neural networks for sentence classification. In: EMNLP, pp. 1746–1751 (2014)
5. Kalchbrenner, N., Grefenstette, E., Blunsom, P.A.: Convolutional neural network for modelling sentences. arXiv preprint arXiv:1404.2188 (2014)
6. Tang, D., Qin, B., Liu, T.: Document modeling with gated recurrent neural network for sentiment classification. In: EMNLP, pp. 1422–1432 (2015)
7. Tai, K.S., Socher, R., Manning, C.D.: Improved semantic representations from tree-structured long short-term memory networks, pp. 1556–1566. ACL (2015)
8. Liang, D., Zhang, Y.: AC-BLSTM: asymmetric convolutional bidirectional LSTM networks for text classification (2016)
9. Yogatama, D., Dyer, C., Ling, W., et al.: Generative and discriminative text classification with recurrent neural networks. arXiv preprint arXiv:1703.01898 (2017)
10. Mou, L., Peng, H., Li, G., et al.: Discriminative neural sentence modeling by tree-based convolution. arXiv preprint arXiv:1504.01106 (2015)
11. Chen, X., Qiu, X., Zhu, C., et al.: Sentence modeling with gated recursive neural network. In: EMNLP, pp. 793–798 (2015)

12. Zhang, Y., Wallace, B.: A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification. arXiv preprint arXiv:1510.03820 (2015)
13. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Iformation Processing Systems 2013. Proceedings of a Meeting held 5–8 December, 2013, Lake Tahoe, Nevada, USA, pp. 3111–3119 (2013)