Association for Information Systems
AIS Electronic Library (AISeL)

PACIS 2018 Proceedings

Pacific Asia Conference on Information Systems (PACIS)

6-26-2018

Short Text Classification Research Based on TW-CNN

Kefeng Pei
Nanjing University of Aeronautics and Astronautics, peikefeng@163.com

Yongzhou Chen
Nanjing University of Aeronautics and Astronautics, yzchen@nuaa.edu.cn

Jing Ma
Nanjing University of Aeronautics and Astronautics, majing5525@126.com

Weimin Nie
Nanjing University of Aeronautics and Astronautics, mingzhu61@163.com

Follow this and additional works at: https://aisel.aisnet.org/pacis2018

Recommended Citation
Pei, Kefeng; Chen, Yongzhou; Ma, Jing; and Nie, Weimin, "Short Text Classification Research Based on TW-CNN" (2018). *PACIS 2018 Proceedings*. 41.
https://aisel.aisnet.org/pacis2018/41

This material is brought to you by the Pacific Asia Conference on Information Systems (PACIS) at AIS Electronic Library (AISeL). It has been accepted for inclusion in PACIS 2018 Proceedings by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact elibrary@aisnet.org.

Short Text Classification Research Based on TW-CNN*

Submission Type: Completed Research Paper

Kefeng Pei

College of Economics and Management, Nanjing University of Aeronautics and Astronautics, Nanjing, China

peikefeng@163.com

Yongzhou Chen

College of Economics and Management, Nanjing University of Aeronautics and Astronautics, Nanjing, China

yzchen@nuaa.edu.cn

Jing Ma

College of Economics and Management, Nanjing University of Aeronautics and Astronautics, Nanjing, China

majing5525@126.com

Weimin Nie

College of Economics and Management, Nanjing University of Aeronautics and Astronautics, Nanjing, China

nieweimin@nuaa.edu.cn

Abstract

Short texts are characterized by short length and sparse features. The study is less effective in the classification of short texts. Motivated by this, this paper seeks to extract features from the "topic" and "word" levels with proposing a convolutional neural network (CNN) based on topic and word, which is named TW-CNN. It uses the Latent Dirichlet Allocation (LDA), a topic model, and word2vec to obtain two distinct word vector matrices, which are then respectively taken as the inputs of two CNNs. After the process of convolution and pooling of the CNNs, there are two different vector representations of the text. And the vector representations are connected with the text-topic vector obtained by LDA, forming the final representation vector of the text. In the end, softmax text classification is conducted. And experiments based on short news texts show that the TW-CNN model has an improvement over the traditional CNNs.

Keywords: Short texts classification, Word2vec, LDA topic model, CNN

* This paper is the National Natural Science Fund Project " Research on Internet public opinion adaptive Topic Tracking Method basing on evolutionary Ontology" (project number: 71373123); Key Projects of Philosophy and Social Science Research in Jiangsu Colleges and Universities" Research on the Evolution Model and Application of Multi-Opinion of public in Jiangsu Education basing on hyper net"(Item number: 2015ZDIXM007).

Introduction

With the rapid development of mobile Internet, represented by tweets, comments and short news, the short texts have become a significant form of information. Compared with long ones, short texts are characterized by short length, sparse features and strong context dependency. So there is a problem to extract useful features from lots of short texts and classify them automatically.

Text classification is one of the main research fields of natural language processing (NLP). And short text classification is currently the hot spot. Confronted with strong context dependency, short length and sparse features, recent study of short text classification is mainly seeking to improve the ability of text vector to represent semantic meaning of the text, which can be advanced by two basic approaches. The first one is extending the given short text to a longer one based on the semantic meaning, resulting in a better representing ability of the text vector. In order to extend the text, Li, X. D. added the key words associated with specific topics, probabilities of which were above a given threshold, to the short texts and got a better experimental result (Li, X. D. 2015). Zhan, Y. et al. proposed a short text categorization based on theme ontology feature extended (Zhan, Y. et al. 2014). Hu, Y. J. et al. put forward a high-frequency words expansion method based on LDA, using LDA to derive latent topics from the corpus, extending the topic words into the short-text (Hu, Y. J. et al. 2013). However, the first method can also change the intrinsic semantic information and structure. The second method is improving the representing vectors of texts by extracting more information from the word level of short texts. Meng, X. et al. proposed a method of short text expansion and classification based on word embedding (Meng, X. et al. 2017). Gao, J. Y. et al considered the defects of applying TF-IDF to text classification and optimized TF-IDF by iteration (Gao, J. Y. et al. 2011). This kind of method improves text vector from the aspect of semantic meaning of word mainly, but overlooks other aspects of semantic features.

Moreover, thanks to the development of deep learning, some models of deep learning such as CNN and RNN are also introduced into NLP. Yoon Kim firstly applied CNN to English text classification, with modeling CNN based on texts, using static vectors and training CNN (Yoon Kim 2014). Richard Socher et al. put forward semi-supervised recursive autoencoders to predict sentiment distributions (Richard Socher et al. 2011). The Chinese researchers also attempt to apply deep learning to Chinese short text classification. Yin, Y. B. et al. proposed a short text classification algorithm based on convolutional neural network and KNN (Yin, Y. B. et al. 2016). And Yu, B. G. et al. put forward a multi-input convolutional neural network model CP-CNN, which used pinyin sequences to characterize the feature at the character level, thus to build double input matrix at the character and phrase level (Yu, B. G. et al. 2018). The CNN has been applied to text classification, and some reasonable results have been obtained. However, the extracted text information by the CNN mainly stays at the two levels of "word" and “character”, without considering the other semantic features, such as the topic feature.

All in all, the paper seeks to extract semantic meaning of short texts fully from “topic” level and “word” level by LDA topic model and word2vec respectively, and proposes a topic and word convolutional neural networks, named TW-CNN.

Word2vec Model and LDA Topic Model

An Introduction to Word2vec Language Model

Based on the effort of Bengio and Hintonm, word2vec was created by a team led by Tomas Mikolov at Google, which attempted to produce a vector space and assigned each unique word a corresponding vector in the space.

The word2vec is a group of three-layer neural network models and has two model architectures: continuous bag-of-words Model (CBOW) and continuous Skip-Gram. The architectures of two kinds of models are shown as follow:

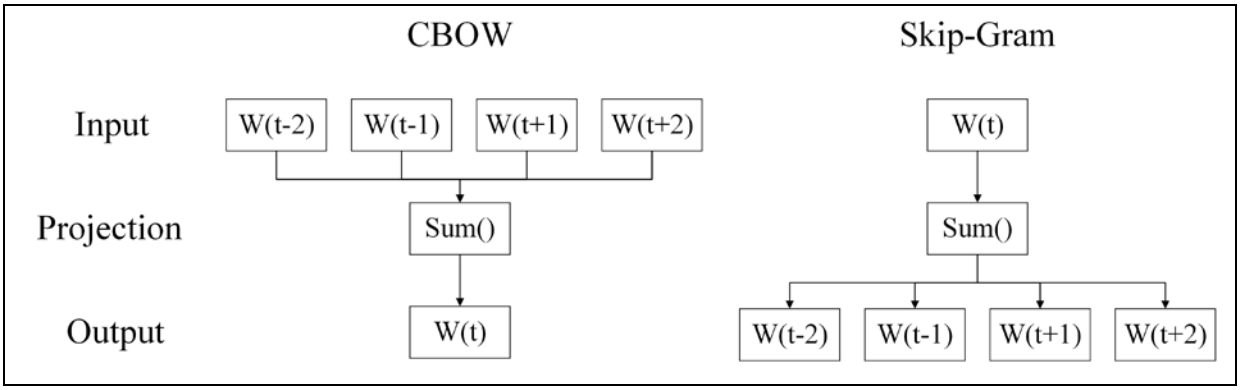

Figure 1. Two Forms of Word2vec

As is shown, CBOW and Skip-Gram are both based on semantic relation of context. And CBOW predicts the current word from a window of surrounding context words, while skip-gram uses the current word to predict the surrounding window of context words. CBOW and Skip-Gram can be trained similarly, so the paper chooses CBOW to demonstrate the details of word2vec.

CBOW takes as its input the semantic information of surrounding context words, which is initially represented by randomly-generated word vectors with same dimension. The projection layer adds up all word vectors to get the projection of the semantic information. And the output layer is actually a Huffman tree, the nodes of which are the words in the given corpus.

An Introduction to LDA Model

LDA model is a type of topic model proposed by Blei, which is an unsupervised generative statistical model. It assumes that each document can be viewed as a mixture of various topics. LDA is generated through a process of sampling. It describes how to generate words in a document under a hidden topic.

A Text Classification Model Based on CNN and LDA Topic Model

Description of Method, Process and Architecture

The application of deep neural networks to NLP is inspired by the rapid development of deep learning. The traditional CNN takes word vector input matrix obtained by word2vec as its input. And there are several convolutional kernels which are designed to extract features. Then the pooling layer continues extracting features. In the end, it joins the extracted features together to form the text vector, which then serves as the input of softmax layer for classification. The traditional model is shown as follows:

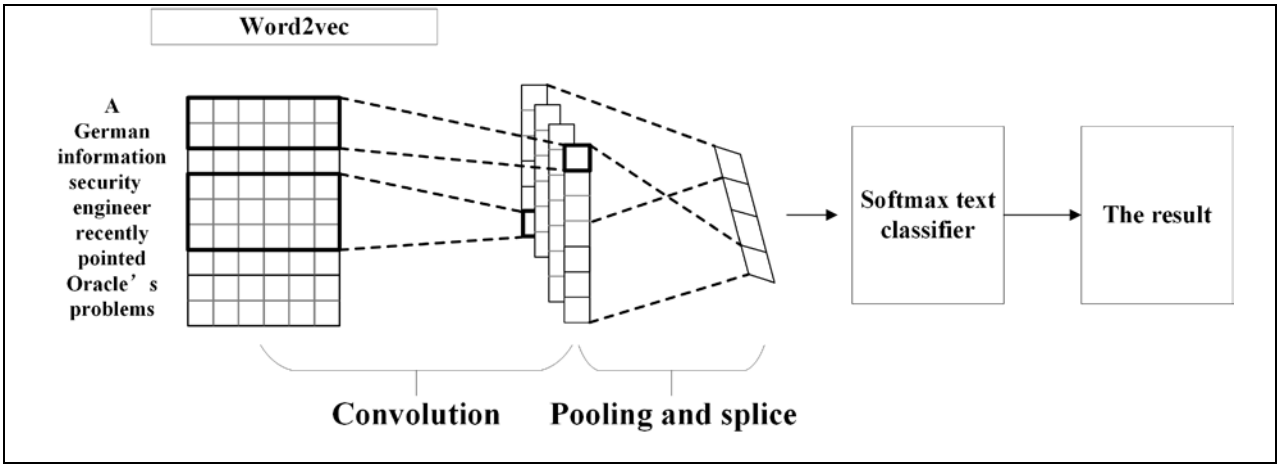

Figure 2. The Classification Process of Traditional CNN

The above-mentioned CNN takes advantage of the word vector of every single word to generate a text vector, which implies that this kind of CNN extracts information on "word", without consideration of

information on "topic". So it's intended that the traditional CNN can be improved by information on "topic" generated by LDA topic model, forming a new model named TW-CNN. Firstly, it gets a text-topic matrix and a topic-word matrix by LDA, and then transposes the topic-word matrix to get a word-topic matrix, which shows the ability of every word representing different topics. And it gets rid of the broad words and converts the given text into a word vector matrix which is taken as the input of a CNN. At the same time, it gets the set of word vectors by means of word2vec and forms a word vector matrix, which is also taken as input of another CNN. And then construct two distinct CNNs based on the above word vector matrices respectively. So there are two text vectors obtained. And it splices the two text vectors and the text-topic vector generated by LDA into the final text vector. In the end, the softmax classification is performed to get the prediction. The model is shown below:

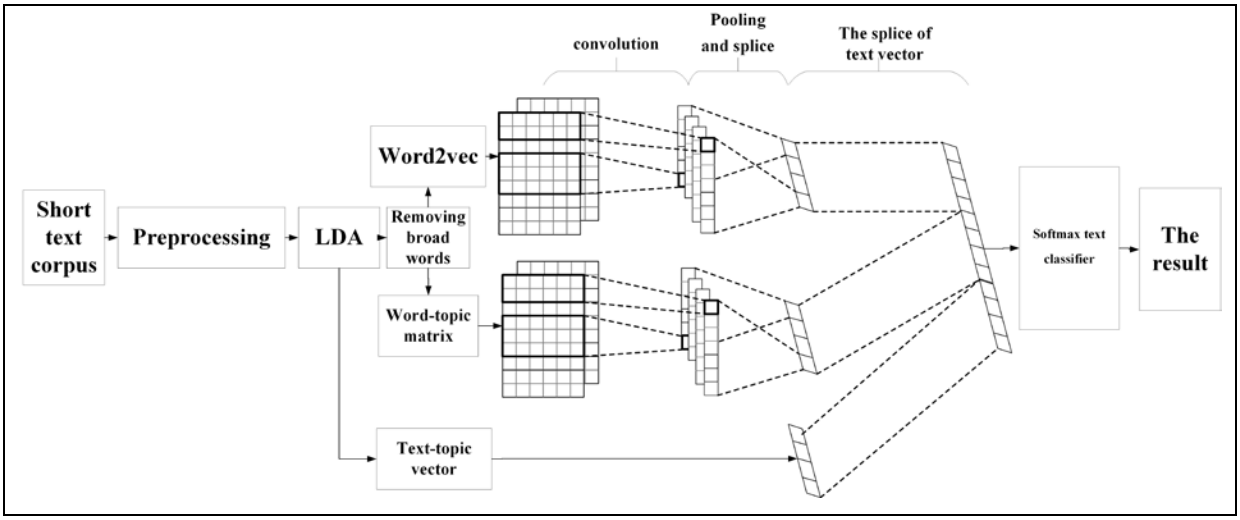

Figure 3. The Classification Process of TW-CNN

Modeling LDA and Removing the Broad Words

LDA topic model is one of the wildly-studied topic models. It can generate the text-topic matrix and the topic-word matrix, which represent the distribution of the specified k topics in the text and the distribution of the word under each topic, respectively. And the former one represents topic distribution for the text. And a text-topic matrix θ and a topic-word matrix ϕ can be generated by modeling short text corpus using LDA.

Moreover, the broad words, which refer to the words that can appear in too many topics can influence the semantic understanding of short texts and even make the result of word2vec worse. So, it's necessary to get rid of the broad words after removing the stop words. The word-topic matrix, the transpose of the topic-word matrix produced by LDA, can demonstrate the ability of a specific word to represent different topics. For a specific word, the more reasonable it represents a topic, the bigger numerical value it has with the topic in topic-word matrix. If a word is a broad one, the distribution of presenting abilities for each topic is more evenly distributed. So, the possibility that a word is a broad one can be evaluated by the variance of the topic-representation vector of the word (actually the column vector of the topic-word matrix). These broad words can be removed from the texts according to the variance. The variance computation equation is shown as follows:

$$s_v^2 = \frac{\sum_{k=1}^{K} (g_{v,k} - \overline{g}_v)^2}{K} \quad (1)$$

in which s_v^2 is the variance of the word *v*, and $g_{v,k}$ denotes the distribution value of the word *v* on the *k*-th topic, and $\overline{g}_v$ represents the mean of the word *v* distributed over the various topics.

Input Representation of Short Texts

Since a CNN takes a vector matrix as its input, it is necessary to convert the unstructured text to a vector matrix. To fully extract the semantic information of a short text, this paper converts a short text to two vector matrices from the levels of "word" and "topic", forming multi-input convolutional neural network.

CBOW, a type of word2vec, can generate a vector matrix to represent the text from the level of "word". It can convert all the words in the text to vectors of the same dimension. And LDA topic model can produce a vector matrix to represent the text from the perspective of "topic". The word vectors of all the words in a short text are spliced together according to the order of the words in the text to form the word vector matrix for the text $x_{1:n} = [x_1, x_2, ..., x_n] \in R^{n*t}$, where *n* represents the number of the short text words, and *t* represents the dimension of the extracted word vector, and x_i represents the word vector of the *i*-th word.

Short text input representation at the "topic" level is achieved by LDA topic model. After LDA modeling, a topic-word matrix representing the distribution of words in each topic is obtained. And then the matrix is transposed into the word-topic matrix, each row in the matrix representing the ability of topics to represent the word. The word-topic matrix can also be regarded as the word vectors with the same dimension as the number of the topics. Similarly, the word vectors obtained are spliced together according to the order of the words in the text to form the word vector matrix of the text $y_{1:n} = [y_1, y_2, ..., y_n] \in R^{n*k}$, where *n* is the number of short text words, and *k* is the number of topics, and y_i is the word vector of the *i*-th word.

As the short texts in corpus have different number of words, the length of input vector matrices *n* is set to the maximal word number of short text in corpus taking the empty vector in the same dimension as a supplement.

The Convolutional Layer and Pooling Layer

The convolutional layer and the pooling layer are the core building blocks of a CNN. And the former one is designed to extract a new feature from the local feature of the input. And the latter one can reduce the dimension of the feature obtained by the convolutional layer.

In detail, it's supposed that the size of receptive field of the kernel is *h*. And a conventional operation is performed on the receptive field of the input vector matrix, denoted by $x_{i:i+h-1}$, and the convolution operation formula is shown as follows:

$$c_i = f(w * x_{i:i+h-1} + b) \tag{2}$$

in which $f()$ is the activation function of neural networks, such as Relu and tanh, and *w* represents the weight with b representing the bias.

The kernel is convolved across the width and height of the input to generate the corresponding feature map $c = [c_1, c_2, ..., c_{n-h+1}]$ to represent the feature of the text. Moreover, there are a group of words to make up a sense group, so it's necessary to construct multiple convolution kernels to extract the features of different granularity of the text.

After the convolution process, there are several feature maps. The pooling layer is used to extract the most representative feature. This paper applies the maxpooling function to choose the maximal value to represent the feature map, shown as follows:

$$m = \max\{c\} \tag{3}$$

And the text vector representation is generated by splicing the maxpooling results of the several feature maps.

Text Vector Splicing

After the convolution and pooling process above, there are two text representation vectors μ and γ, representing the word feature and the topic feature respectively. What's more, the LDA can generate the text-topic matrix θ. The topic distribution of the text, donated by η, can complement the topic feature. The final text vector representation can be obtained by splicing the three text vector features, denoted by $v_m = [\mu_m{}^T, \gamma_m{}^T, \eta_m{}^T]^T$.

Dropout and Softmax Classification

To reduce overfitting, dropout is introduced into the fully connected layer, which means at each training stage, individual nodes are either "dropped out" of the net with probability $1-p$ or kept with probability p. Apply the softmax classification on the processed text vector and then the class of the given text is obtained. The softmax classifier is a common method used for multi-category problems, which is calculated as follows:

$$p(y_k) = \frac{\exp(s_k * v_m + b_k)}{\sum_{i=1}^{n} \exp(s_i * v_m + b_i)} \tag{4}$$

where $p(y_k)$ represents the probability that the text belongs to y_k, and s_k represents the weight, and b_k represents the bias, and v_m represents the final text representation vector of the *m*-th text.

The Experiment and the Analysis of the Result

The Data and the Preprocessing

The paper takes SogouCS of Sogou Labs, version 2012 as the corpus and chooses the news of which the number of Chinese character is less than 500. This corpus contains 10 distinct classes, each of which consists of 500 news. And the paper applies 10 fold Cross-Validation.

Text preprocessing is one of the key steps in the field of text analysis. Firstly, this paper calls jieba, one of practical Chinese words segmentation utilities with several dictionaries added, to segment the chosen short news, and then removes the stop words, which can influence the experiment. In order to remove stop words as many as possible, this paper collects several lists of stop words.

The Experimental Set-up

The Evaluation Indices

The paper adopts accuracy, precision, recall and F1 to evaluate different approaches.

And the accuracy is calculated as follows:

$$\text{Accuracy} = \frac{\sum_{i=1}^{N} \left| y_i = \hat{y}_i \right|}{N} \tag{5}$$

where *N* is the number of the chosen texts, and y_i is the actual category of the *i*-th short text, and $\hat{y}_i$ is the category predicted by the approach applied.

And the precision, recall and F1 are calculated as follows:

$$Precision = \frac{TP}{TP + FP} \tag{6}$$

$$\mathrm{R}ecall = \frac{TP}{TP + FN} \tag{7}$$

$$\frac{1}{F1} = \frac{1}{2} * (\frac{1}{\mathrm{Pr}ecision} + \frac{1}{\mathrm{Re}call}) \tag{8}$$

where TP, FP and FN are shown in the following confusion matrix.

Table 1. Confusion Matrix

The Reality	The Prediction	
	Positive	Negative
Positive	True Positive(TP)	False Negative(FN)
Negative	False Positive(FP)	True Negative(TN)

In order to get more objective results, this paper uses the average accuracy, precision, recall and F1 of multiple iterations as the final result.

The Parameters Setting

For LDA topic model, let $\alpha = 50/K$, $\beta = 0.1$ and K=50 , where α and β are the priori hyperparameters of text-topic distribution and topic-word distribution, respectively. And K is the number of topics. And for word2vec, the word vector dimension is set to 100 dimensions.

In addition, the TW-CNN is based on TensorFlow. And the local information extracted by the CNN varies as the size of the receptive field changes. Therefore, this paper considers three sizes of the receptive field, respectively 3, 4, 5. The other important parameters are shown as follows:

Table 2. The Important Parameters of the CNN

Parameters	Values
The Number of Filters	20
Batch	50
Dropout	0.25
The Number of Iterations	2000

Note that all the parameters of LDA topic model and TW-CNN are set based on experience.

The Analysis of the Experimental Result

The Influence of Removing Broad Words

As is shown above, before converting the short texts to vectors, this paper takes advantage of the topic-word matrix, one of the output of LDA, to remove the broad words. To verify the significance of removing broad words, a contrast experiment is conducted. Here is the result:

Table 3. Broad Word Comparison Experiment

Classifiers	Accuracy	Precision	Recall	F1
Word2vec-CNN(without removing broad words)	0.9446	0.9461	0.9442	0.9444
Word2vec-CNN(removing broad words)	0.9537	0.9554	0.9539	0.9540
Wt-CNN(without removing broad words)	0.9691	0.9694	0.9689	0.9687
Wt-CNN(removing broad words, and k=100)	0.9759	0.9761	0.9760	0.9760

As is shown in the table, the classifiers after removing broad words perform better than the ones without removing broad words, which implies that it's useful and necessary to remove broad words with the help of LDA topic model.

Comparison with the Traditional Models

To compare TW-CNN with the traditional models, this paper performs the experiment of text classification with Logistic Regression (LR), Support Vector Machine (SVM), k-NearestNeighbor (KNN) and Random forest (RF). The parameters of the traditional models are all adjusted to make the results as good as possible. And the result is shown as follows:

Table 4. Comparison of the Classification Results of Each Classifier

Classifiers	Accuracy	Precision	Recall	F1
SVM	0.9180	0.9199	0.9189	0.9184
LR	0.9000	0.9034	0.9014	0.9008
RF	0.8278	0.8329	0.8303	0.8286
KNN	0.7980	0.8245	0.8016	0.7997
TW-CNN	0.9817	0.9817	0.9814	0.9813

Among the traditional classifiers, SVM and LR outperform the others. In particular, SVM is the best one, considering all the evaluation indices. Correspondingly, the performances of KNN and RF are not ideal. And the performance of KNN is the worst. The result above shows that for the same text corpus, there is a big difference in classification performance among different classification algorithms. And it can be observed that the TW-CNN text classification model proposed in this paper is the best among all the classifiers in terms of classification accuracy, precision, recall and F1, implying that compared with the traditional ones, the TW-CNN has a great advantage over the traditional models in extracting short text features.

Comparison with the Other CNNs

To verify the effectiveness of adding the features from the "topic" level to the traditional CNN, this paper compares the accuracy of several kinds of CNNs with various inputs, listed as follows:

- rand-CNN, without extracting text feature, taking the word vectors matrix obtained by splicing randomly-generated word vectors as its input.
- word2vec-CNN, taking the word vectors matrix obtained by splicing word vectors generated by word2vec as its input.
- word2vec-CNN-tt, with the input same as the word2vec-CNN and the output text vector combined with the text-topic vector of LDA.
- wt-CNN, taking the word vectors matrix obtained by word vectors from word-topic matrix produced by LDA as its input.
- wt-CNN-tt, with the input same as the wt-CNN and the output text vector combined with the text-topic vector of LDA.

- TW-CNN, taking the input of word2vec-CNN and the input of wt-CNN as multi-input, and the output vector combined with the text-topic vector of LDA.

Let the number of topics of LDA modeling, which is denoted by K, be 100, and the result of the experiment is shown as follows:

Table 5. Comparison of the Classification Results of Each Classifier

Classifiers(after removing the broad words)	Accuracy	Precision	Recall	F1
rand-CNN	0.8371	0.8689	0.8349	0.8420
word2vec-CNN	0.9537	0.9554	0.9539	0.9540
word2vec-CNN-tt	0.9549	0.9562	0.9548	0.9545
wt-CNN	0.9759	0.9761	0.9760	0.9760
wt-CNN-tt	0.9765	0.9768	0.9762	0.9762
TW-CNN	0.9817	0.9817	0.9814	0.9813

Please note that all experiments above have already removed the broad words.

Firstly, the rand-CNN performs worst, which implies that it's necessary to extract features from texts when generating word vectors. What's more, the performances of the wt-CNN and wt-CNN-tt are both better than the word2vec-CNN and the word2vec-CNN-tt, respectively. As is known to us, the LDA topic model can extract the topic feature of a text, and the word2vec can extract the word feature. So it's concluded that for the short texts, the topic feature of a short text can represent the semantic information of the text better than the word feature.

Secondly, adding the text-topic vector to the fully connected layers can improve the classification, which is shown by the comparison between the performance of word2vec-CNN and word2vec-CNN-tt, wt-CNN and wt-CNN-tt. And it's observed that the wt-CNN-tt improves the performance a little compared with the wt-CNN. The reason is that the wt-CNN already extracts the topic feature, so adding the text-topic vector improves little.

All in all, it's obvious that both taking the word-topic matrix as the input of the CNN and adding the text-topic vector to the fully connected layers can improve the classification performance. And the TW-CNN based on the improvements above has the best performance.

Conclusions

Aiming at the problems such as the difficulty of extracting effective text information from short texts and the strong dependence of contexts, this paper proposes a TW-CNN model, which combines the LDA topic model and the word2vec to obtain the semantic features of short texts from the "topic" and "word" levels, then extracts the text features through CNN, and then combines the text-topic vector to generate a short text vector representation, and finally short text classification is performed.

Experiments show that compared with the traditional machine learning text classification methods, the classification accuracy of TW-CNN model proposed in this paper has been improved by about 8%. In addition, all the improvements proposed in this paper can improve the classification accuracy of short texts. In general, the TW-CNN model has a classification accuracy improvement of about 5% over the original word2vec-based CNN text classification methods.

Although this paper introduces the "topic" level of short text to improve the efficiency of short text classification, it does not consider the contextual feature information in the text. Therefore, it is the focus of further study to combine the contextual semantic information extracted by the recurrent neural network (RNN) with other text features. What's more, this paper applies LDA topic model to extract topic-level information, which is proved to be an effective method to improve the classification by the result of experiment. However, as is known to us, LDA suffers a serious

performance reduction when dealing with short texts. So it's another focus to seek a more suitable topic model.

References

Gao, J. Y., Xu, C. J., and Feng, Y. J. 2011. "Application of TF-IDF Based on Iteration in Short Text Categorization," *Information studies: Theory & Application* (34:06), pp. 120-122.

Guo, S. H., and Fan, X. H. 2010. "An Improved Short Text Classification Algorithm Based on Bayesian Network," Journal of Guangxi Normal University. Natural Science Edition (28:03), pp. 140-143.

Hu, Y. J., Jiang, J. X., and Chang, H. Y. 2013. "A New Method of Keywords Extraction for Chinese Short －Text Classification," New Technology of Library and Information Service:06), pp. 42-48.

Kim, Y. 2014. "Convolutional Neural Networks for Sentence Classification," in: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing. Doha, Qatar.

Li, X. D., Cao, H., Ding, C., and Huang, L. 2015. "Short-Text Classification Based on Hownet and Domain Keyword Set Extension," New Technology of Library and Information Service:02), pp. 31-38.

Liu, X. S. 2007. "Modeling Based on Improved SVM Text Classification," Information studies: Theory & Application), pp. 841-843.

Meng, X., and Zuo, W. L. 2017. "Short Text Expansion and Classification Based on Word Embedding," Journal of Chinese Computer Systems (38:08), pp. 1712-1717.

Socher, R., Pennington, J., Huang, E. H., Ng, A. Y., and Manning, C. D. 2011. "Semi-Supervised Recursive Autoencoders for Predicting Sentiment Distributions," in: Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing. Edinburgh, Scotland, UK.

Yin, Y. B., Yang, W. Z., Yang, H. T., and Xu, C. Y. 2016. "Research on Short Text Classification Algorithm Based on Convolutional Neural Network and KNN," Computer Engineering), pp. 1-6.

Yu, B. G., and Zhang, L. B. 2018. "Chinese Short Text Classification Based on CP-CNN," Application Research of Computers:04).

Zhan, Y., and Chen, H. 2014. "Short Text Categorization Based on Theme Ontology Feature Extended," Journal of Hebei University (Natural Science Edition) (34:03), pp. 307-311.

Zhou, Q. P., Tan, C. G., Wang, H. J., and Zhan, M. X. 2016. "Improved KNN Text Classification Algorithm Based on Clustering," Application Research of Computers (33:11), pp. 3374-3377.