A Key Sentences Based Convolution Neural Network for Text Sentiment Classification

Zhang Mohan\textsuperscript{1,a} and Xiang Yang\textsuperscript{2,b}

\textsuperscript{1}Department of Computer Science and Technology, Tongji University, Shanghai, China 201800; \\
\textsuperscript{2}Department of Computer Science and Technology, Tongji University, Shanghai, China 201800

Email: \textsuperscript{a} zhangmohan@tongji.edu.cn; \textsuperscript{b} shxiangyang@tongji.edu.cn

Abstract. Existing research treated all sentences in the text on an equal basis during the training process and did not consider that key sentences tend to have a stronger influence. We propose a Convolutional Neural Network text sentiment classification model based on the key sentences enhancement. The proposed model can identify key sentences in the text and generate text representation based on these key sentences to reduce noise and improve the accuracy of the model sentiment classification. The experiment results show that the proposed model improves the accuracy of the text sentiment classification compared with other classic text sentiment classification models.

1. Introduction
With the continuous development of the Internet, a large amount of text data has been generated, making it possible to apply a deep learning method, which requires a large amount of training data, to the field of text classification. The text classification method based on deep learning has made great progress and has gradually become the mainstream direction of text classification research [1]. At present, the text emotion classification method based on deep learning treats each sentence in the text to be classified equally, and regards it as a whole to perform feature extraction and emotion classification [2]. Besides, this achieves good results and the accuracy rate is basically about 80\%. However, there are only a few key sentences in the text that contain the core ideas. These key sentences are of great significance to the results of sentiment classification. The remaining non-key sentences have little meaning for the emotional classification results. When using the deep learning method to extract the features of the text, the existence of non-key sentence noise increases the difficulty of model classification, and may even play a negative role. Therefore, this paper attempts to improve the classification accuracy by extracting the key sentences in the text and reducing the non-critical noise.

2. Related Work
In 2014, Kim first proposed a text classification model based on Convolution Neural Network (CNN) [3], which opened up the research of text classification based on deep learning. After Kim proposed the basic text classification model based on convolutional neural network, many researchers have carried out further research on this basis and proposed a good model. For example, in order to extract the relationship between words-words and different distances in text, Kalchbrenner et al. proposed a
text classification model (DCNN) based on dynamic pooling convolutional neural network. Mingbo Ma et al. proposed text classification models (DCNNs) based on syntactic dependency trees, which can better extract the dependence between words and words in sentences [4]. Xiang Zhang et al. proposed a convolutional nerve at the character level Network-based text-classification model (character-level ConvNets), which models feature extraction based on more fine-grained character-level semantics. In addition to the text classification model based on convolutional neural networks, Pengfei Liu et al. proposed a text classification model based on Recurrent Neural Networks (RNN) which can better process text sequences [5]. Tang et al. used the long-term short-term memory neural network (LSTM) for text classification to solve the problem of gradient disappearance in the RNN model [6]. In addition, Chunting Zhou et al. combined CNN and LSTM and proposed a long-term and short-term memory convolutional neural network model (C-LSTM) for text classification, while using CNN and LSTM networks to improve the ability of the model to extract text features [7-9].

The above-mentioned classic deep learn based text models have achieved good results, but these models do not extract and identify the key sentences in the text [10-11]. The key sentences in the text contain the core ideas which will influence results of the text classification, especially in the text sentiment prediction classification task. Therefore, the deep learning model still has room for improvement in the text sentiment classification task [12].

In this paper, the text sentiment classification based on the key sentence enhancement convolutional neural network model is studied. The model is based on the key sentences in the text for emotional classification, which enhances the ability of the model to extract features to a certain extent and the accuracy rate of the text sentiment classification. In addition, this paper also improves the traditional convolutional neural network model and enhances it by means of multi-type pooling.

3. CNN Model for Text Sentiment Classification

3.1. Key sentence recognition

We use a CNN model to recognize whether a sentence is a key sentence in the document. We divide sentences into three categories: when a key sentence appears in a positive document, it is a positive key sentence; when a key sentence appears in a negative document, it is a negative key sentence; All other sentences belong to a third, neutral class: these are non-key-sentences.

The network structure of the key sentence recognition model based on CNN is shown in Figure 1, which includes an input layer, a convolution layer, a pooling layer and a softmax layer. The model takes an n×d matrix as input. Each row of the matrix is a vector of a word. The dimension of the word vector is d, which is trained by the word2vec method. The number of words contained in the sentence is n, as shown in Figure 2. We perform feature extraction operations on the convolutional layer and the pooling layer to obtain the feature vector of the sentence as input to the softmax layer. We use the softmax regression model to calculate the probability of the sentence $P_{sen}$. $P_{sen}$ is the probability of the positive key sentence, negative key sentence and non-key-sentence for each sentence. The equation is:

$$P_{sen}(y_{sen}^{ij} = k; E, C, W_{sen}) = \frac{\exp(W_{sen}^{(k)} x_{sen}^{ij})}{\sum_{k=1}^{3} \exp(W_{sen}^{(k)} x_{sen}^{ij})}$$

(1)

Where $x_{sen}^{ij}$ is the vector of the j-th sentence in the i-th document, $y_{sen}^{ij}$ denotes the label for sentence j in document i, E denotes the word embedding matrix, C denotes the convolution layer parameters, and $W_{sen}$ is a matrix of weights.
3.2. Key Sentence Based Document Embedding
To get a better a document embedding, we use a weighted sum of its constituent sentence vectors to represent this document. Each sentence weight is set to the estimated probability of the key sentences. The equation is:

\[
\mathbf{x}^i_{doc} = \sum_{j=1}^{N_i} \mathbf{x}^j_{sen} \cdot \max\{p^i_{pos}, p^i_{neg}\}
\]  

Where \(\mathbf{x}^i_{doc}\) is vector of i-th document, \(\mathbf{x}^j_{sen}\) is the vector of the j-th sentence in the i-th document, \(p^i_{pos}\) denotes the probability of sentence \(\mathbf{x}^j_{sen}\) is positive key sentence, \(p^i_{neg}\) denotes the probability of sentence \(\mathbf{x}^j_{sen}\) is negative key sentence.

3.3. KSt-CNN
We use another CNN model to perform feature extraction and sentiment classification on the document. The document representation vector obtained based on the method described in section 2.2
is used as the input to the CNN model. A convolution operation is performed using a plurality of convolution kernels and a feature extraction operation is performed using an activation function to obtain a plurality of feature vectors. Then, the pooling method is used to perform quadratic feature extraction, and the fault tolerance of the model is improved, and the final feature vector of the text is obtained. Finally, we use softmax regression method to calculate the probability $P_{doc}$ of the document belongs to each class, as shown in Equation 3:

$$P_{doc}(y_{doc}^j = k; E, C, W_{doc}) = \frac{\exp(W_{doc}^{(k)} x_{doc}^i)}{\sum_{k=1}^{K} \exp(W_{doc}^{(k)} x_{doc}^i)}$$  \hspace{1cm} (3)

The whole network structure of the key sentence based convolution neural network (KSt-CNN) is shown in Figure 3.

![Figure 3 KSt-CNN Network Structure](image)

3.4. Multi Type Pooling
Pooling is an important part in the CNN model. The pooling operation is generally performed after the convolution operation. The main function is to perform secondary extraction of features, and also to reduce the dimension of the feature map obtained by the convolutional layer [13]. The pooling operation selects or calculates a feature value as a representative of the sub-vector in the sub-vector from which the convolution operation obtains the feature vector, and traverses the entire feature vector in a certain step size. The pooling operation mainly consists of two kinds of max-pooling and average-pooling. Max-pooling refers to selecting the largest eigenvalue as the representative among the sub-feature vectors; average-pooling refers to calculating the average value of all eigenvalues of the sub-vector as a representative.
The classical CNN model only uses a pooling method, and in order to extract more features, this paper proposes a multi-type pooling CNN text sentiment classification model based on KSt-CNN, we call it KSt-MpCNN. Multi type pooling is shown in figure 4.

![Multi Type Pooling Structure](image)

**Figure 4.** Multi Type Pooling Structure

### 3.5. Loss Function Regularization

One of the most commonly used loss functions in neural networks is the cross-entropy loss function, especially the neural network used for classification [14]. For a neural network with n classification, the output of the last layer is a vector of the dimension, and the dimension of the vector represents the probability value of the sample, and the sum of each dimension of the vector is 1. The equation is as follows:

$$H(y, \bar{y}) = - \sum y_i \ln(\bar{y}_i)$$ \hspace{1cm} (4)

In the ordinary cross entropy loss function, the loss value of each category is equal weight when it is backpropagated, but in the actual situation, because the proportion of samples in each category is inconsistent, the loss value of each category should also be emphasized. Thus, we introduce a custom loss function, known as weighted cross entropy in which the cross-entropy loss is multiplied by a corresponding penalty weight specified in a penalty matrix, as shown in the table 1.

| predicted/excepted | positive | negative | neutral |
|--------------------|----------|----------|---------|
| positive           | 1        | 4        | 3       |
| negative           | 4        | 1        | 3       |
| neutral            | 2        | 2        | 1       |

Table 1. The penalty matrix

In the sentiment classification task, if the true value of a sample is positive and predicted to be neutral, the loss value of the cross entropy will be regularized on the original basis, multiplied by the corresponding penalty coefficient. For example, if $y$ is [0,1,0] (negative), $\bar{y} = [0.2,0.3,0.5]$ (neutral), then the default cross entropy will result in 1.204, while the result of weighted cross entropy is 2.408.

### 4. Experiment

#### 4.1. Dataset

The two datasets used in this experiment are from a movie film review data set in the IMDB. This paper selects 2000 film reviews as positive emotional reviews, 1500 film reviews as negative emotional reviews, and 500 neutral emotional reviews. The other is a dataset from Yelp. This paper selected 3,500 reviews as positive emotions, 2000 reviews as negative emotions, and 1,000 as neutral emotions.
4.2. Experiment setting
The corpus used in this experiment is in English, so the word vector used is the public word vector that Google uses the word2vec method to train on a large number of Google news corpora. The word vector dimension of each word is 300 dimensions [15].

The convolution kernels of the key sentence extraction part of the KSt-CNN and KSt-MpCNN models and the prediction part of the text emotion are set to 128, and the activation function is ReLU function. The maximum length of each sentence is set to 25 words. The maximum length of the text is set to 200 sentences.

4.3. Experimental results and analysis
The experiment uses a variety of classical text sentiment prediction methods as benchmarks, and experiments are carried out on two different data sets. The experimental results are as follows.

**Table 2. Experimental results based on IMDB dataset**

| Method        | Recall  | Precision | F     |
|---------------|---------|-----------|-------|
| NBSVM         | 79.49%  | 76.40%    | 77.91%|
| CNN           | 83.50%  | 81.50%    | 82.49%|
| Doc-CNN       | 86.04%  | 83.14%    | 84.57%|
| KSt-CNN       | 87.08%  | 85.69%    | 86.38%|
| KSt-MpCNN     | 88.11%  | 86.23%    | 87.16%|
| KSt-MpCNN-reg | 89.03%  | 87.01%    | 88.01%|

**Table 3. Experimental results on Yelp datasets**

| Method        | Recall  | Precision | F     |
|---------------|---------|-----------|-------|
| NBSVM         | 79.45%  | 75.49%    | 77.42%|
| CNN           | 88.32%  | 85.46%    | 86.87%|
| Doc-CNN       | 89.04%  | 87.14%    | 88.08%|
| KSt-CNN       | 91.07%  | 90.18%    | 90.62%|
| KSt-MpCNN     | 87.17%  | 93.55%    | 90.25%|
| KSt-MpCNN-reg | 90.33%  | 95.26%    | 92.73%|

NBSVM is the Naive Bayes Support Vector Machine (NBSVM) classification method proposed by Wang and Manning in 2012; CNN is the basic convolutional neural network based classification method proposed by Kim in 2014; Doc-CNN is a CNN-based document classification model proposed by Misha Denil et al. KSt-CNN and KSt-MpCNN are the text sentiment classification model of the convolutional neural network based on key sentence enhancement and the text sentiment classification model of multi-type pooled convolutional neural network based on key sentence enhancement, respectively. Kst-MpCNN-Reg is a multi-type reddening convolutional neural network based on key sentence enhancement using the loss function regularization.

It can be seen from Table 2 and Table 3 that the KSt-CNN and KSt-MpCNN models proposed in this paper perform better on the text sentiment classification task than the classical text sentiment prediction method, which proves the feasibility of reducing the text information redundancy and enhancing model classification performance by identifying the key sentences in the text. At the same time, it can be seen that the accuracy of the KSt-MpCNN model is slightly higher than that of the KSt-CNN model, indicating that the multi-type pooling method can improve the effect of CNN classification to a certain extent. And the model Kst-MpCNN-reg with the loss function regularization on both data is better than the Kst-MpCNN using the traditional loss function.
5. Conclusion

We propose a convolutional neural network text sentiment classification model based on key sentence enhancement (KSt-CNN), which can identify key sentences in text, reduce the influence of non-key sentence noise on model classification, and improve the accuracy of convolutional neural networks on text sentiment classification task. Based on KSt-CNN, the classic CNN model is optimized and improved. A multi-type pooled convolutional neural network text sentiment classification model (KSt-MpCNN) based on key sentence enhancement is proposed. KSt-MpCNN can use the multi-type pooling method to better perform the quadratic feature extraction operation and further optimize the text sentiment classification effect of the model. The experimental results verify the feasibility and effectiveness of the KSt-CNN and KSt-MpCNN models. At the same time, the experimental result of regularization of the loss function is carried out. The experimental results show that the regularized cross entropy loss function is superior to the traditional loss function.

Acknowledgement

The authors would like to thank the reviewers for their comments on this article. This work was supported in part by the National Natural Science Foundation of China under Grant No.71571136 and in part by the Project of Science and Technology Commission of Shanghai Municipality under Grant No.16JC1403000.

References

[1] Zhiheng Huang, Marcus Thint, Zengchang Qin. Question classification using head words and their hypernyms. [C]// In Proceedings of the Conference on Empirical Methods in Natural Language Processing, EMNLP, 2008:927–936.

[2] Shen Y, He X, Gao J, et al. Learning semantic representations using convolutional neural networks for web search. [C]// Proceedings of the 23rd International Conference on World Wide Web. ACM, 2014: 373-374.

[3] Yoon Kim. Convolutional neural networks for sentence classification. [J]// arXiv preprint arXiv:1408.5882.

[4] Ma M, Huang L, Xiang B, et al. Dependency-based convolutional neural networks for sentence embedding. [J]// arXiv preprint arXiv:1507.01839.

[5] Cho K, Van Merriënboer B, Gulcehre C, et al. Learning phrase representations using RNN encoder-decoder for statistical machine translation. [J]// arXiv preprint arXiv:1406.1078.

[6] Duyu Tang, Bing Qin, and Ting Liu. Document modeling with gated recurrent neural network for sentiment classification. [C]// EMNLP2015: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pages 1422–1432.

[7] Ye Zhang and Byron C. Wallace. A sensitivity analysis of (and practitioners’ guide to) convolutional neural networks for sentence classification. [J]// arXiv preprint arXiv:1510.03820.

[8] Tai K S, Socher R, Manning C D. Improved semantic representations from tree-structured long short-term memory networks. [J]// arXiv preprint arXiv:1503.00075.

[9] Mnih A, Hinton G. Three new graphical models for statistical language modelling. [C]// Proceedings of the 24th international conference on Machine learning. ACM,2007: 641-648.

[10] Yoav Goldberg. A primer on neural network models for natural language processing. [C]// Proceedings of the 25th international conference on Machine learning. ACM,2008: 160-167.

[11] Quoc V Le and Tomas Mikolov. Distributed representations of sentences and documents. [J]// arXiv preprint arXiv:1405.4053.

[12] Collobert R, Weston J. A unified architecture for natural language processing: Deep neural networks with multitask learning. [C]// Proceedings of the 25th international conference on Machine learning. ACM, 2008: 160-167

[13] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. [J]// arXiv preprint arXiv:1409.0473.

[14] Marc Baroni,Roberto Zamparelli. Nouns are vectors, adjectives are matrices: Representing adjective-noun constructions in semantic space [C]// ACL EMNLP, 2010:1183–1193.

[15] Mikolov T, Chen K, Corrado G, et al. Efficient estimation of word representations in vector space. [J]// arXiv preprint arXiv:1301.3781.