Research on Short Text Classification Based on TextCNN

Tianyu Zhang*, Fucheng You
College of Information Engineering, Beijing Institute of Graphic Communication, No. 1, Xinghua Street (Section 2), Daxing District, Beijing, China

*807696182@qq.com

Abstract. The TextCNN model is widely used in text classification tasks. It has become a comparative advantage model due to its small number of parameters, low calculation, and fast training speed. However, training a convolutional neural network requires a large amount of sample data. In many cases, there are not enough data sets as training samples. Therefore, this paper proposes a Chinese short text classification model based on TextCNN, which uses back translation to achieve data augment and compensates for the lack of training data. The experimental data shows that our proposed model has achieved good results.

Keywords: Text Categorization, TextCNN, Natural language processing, Data augment.

1. Introduction
Classification refers to automatically labeling data. People divide categories by experience in daily life. However, it is impossible to manually classify every page on the Internet according to some rules. Therefore, computer-based efficient automatic classification technology has become an urgent need for people to solve Internet application problems. Similar to classification, technology is clustering. Clustering does not match data to a pre-defined set of tags but automatically aggregates into one or more categories through implicit structures related to other data. Text classification is an important research direction in the field of data mining and machine learning.

Classification is a topic that has been studied in the field of information retrieval for many years. On the one hand, it aims to improve the effectiveness and efficiency in some cases with the application of search; on the other hand, classification is also a classic machine learning technology. In the field of machine learning, classification is performed under a pre-defined category system with annotations.

Text Classification (Text Classification or Text Categorization, TC), or Automatic Text Categorization (Automatic Text Categorization), refers to the computer mapping a text containing information to a predetermined category or several categories of topics process. Text classification also belongs to the field of natural language processing. In this article, Text and Document are indistinguishable and have the same meaning.

2. Related Work
Text classification and clustering technology have a wide range of applications in intelligent information processing services. For example, most online news portals (such as Sina, Sohu, Tencent, etc.) generate many news articles every day. If the manual sorting of this news is very time-consuming and labor-intensive, and the automatic classification or clustering of this news will be News
Classification and follow-up personalized recommendations provide great help. The Internet also has a large number of text data such as webpages, papers, patents, and e-books. The classification and clustering of the text content is an important basis for fast browsing and retrieval of this content. In addition, many natural language analysis tasks, such as opinion mining, spam detection, etc., can also be regarded as specific applications of text classification or clustering techniques.

With the continuous improvement of research methods such as machine learning and deep learning, the solution path of text classification problems has gradually shifted from the previous vector space model (VSM) to the combination of machine learning and deep learning [1]. In the deep learning network, the convolutional neural network CNN can identify predictive n-grams in the text; the convolution structure supports n-grams with similar components to share their prediction behaviors, even if they have not logged in during the prediction process. The specific n-grams that have been passed are also possible, while each layer of the hierarchical CNN focuses on the longer n-grams in the sentence so that the model can be more sensitive to non-continuous n-grams. It can have a significant impact on the effect of text classification [2–3].

What this article realizes is the classification of news headlines. The main feature of news headlines is to summarize rich information in a concise language as possible. According to statistics, 95% of news headlines do not exceed 20 Chinese characters in length. Therefore, the existing research summarizes headline classification as short text classification[4]. Classified categories include finance, real estate, stocks, education, technology, society, current affairs, sports, games, and entertainment.

However, when the number of training data sets is insufficient, the training effect will be poor. At this time, the use of text data enhancement methods can achieve the purpose of improving the training effect. Data enhancement refers to the process of transforming (limited) training data into new data. Moreover, text data enhancement is to operate on text data. In short, it is the use of data augmentation to expand the scale of data.

3. Model structure

3.1. Text preprocessing

As the basis of text vectorization, text preprocessing is an indispensable step to achieve classification. Through word segmentation, the text can be cut into a collection of single words, and a collection of keywords can be extracted. At present, more mature Chinese word segmentation tools such as Zieba word segmentation and ICTCLAS word segmentation of the Chinese Academy of Sciences have achieved good results in word segmentation through the iteration of developers. [6]

The first task of text classification is to transform the text into a clean word sequence suitable for presentation and classification. The preprocessing includes the following contents.

Word segmentation. Chinese differs from English in that there is no explicit separation between words, so Chinese word segmentation technology needs to be used to separate words.

Stemming. Convert the singular and plural, tense, and other deformed words in English into prototypes.

Delete stop words. Such words do not contain any information, such as "的" and "了" in Chinese.

Remove low-frequency words. Some words have only appeared in a few texts and have no practical meaning to most texts, so they need to be removed. [7]

3.2. TextCNN model

When it comes to CNN, it is usually considered to belong to CV and is used for computer vision work [8]. However, in 2014, Yoon Kim made some changes to the input layer of CNN and proposed the text classification model textCNN [9].

The structure of textCNN is the same as that of CNN. The TextcNN model is a variant of the CNN model. It can give full play to the parallel computing capabilities of CNN, and the training speed is faster. In addition to retaining the characteristics of the original CNN, it also adds the ability to extract text features. TextcNN uses one-dimensional convolution to Obtain the n-gram feature representation.
of the sentence, which has a strong ability to extract shallow text features. The model can identify the linguistic n-gram in the task. When it encounters a specific n-gram that has not been registered during the prediction process, its convolution structure can also allow n-grams with similar elements to share the predicted behavior, and each layer of the hierarchical CNN pays attention to the longer n-grams in the sentence, so that the model can be more responsive to non-continuous n-grams Sensitive. By adjusting the height of the convolution kernel, TextCNN can flexibly process various timing information of the comprehensive vocabulary, which improves the model's ability to interpret the text.

Compared with the traditional image CNN network, textCNN has no changes in the network structure (or even simpler). In fact, textCNN has only one layer of convolution, one layer of max-pooling, and finally, the output is externally connected to softmax for n classification. As shown in the following Fig.1:

![TextCNN model](image)

3.2.1. Input layer
The text vector generation first needs to generate word vectors. Using the n-gram language model and word embedding method, the patent text is expressed as a space vector for operation. The core idea is similar to building a word bag model to pack all words into a bag, and the text vector is expressed by the sum of word vectors, regardless of its morphology and word order. The difference is that this
article uses the n-gram language model to generate the sub-words of the words in the text, which helps to solve the expression problems of unregistered words and low-frequency words. It can also capture the word order of words to a certain extent.

First, a sentence can be regarded as a sequence of words; the length of the sequence is n, each word is represented by a vector \( X_i \), and the dimension of each word embedding is k. So the sentence is expressed as follows:

\[
X_{i:n} = X_1 \oplus X_2 \oplus \cdots \oplus X_n
\]  

\( X_{i:i+j} \) is an interval with left closed and right closed.

The input here has two channels; in fact, we can regard it as one because one of the two channels is static, and the other is non-static. Static: The word vector is pre-trained and will not change during training. Non-static: The word vector changes with model training. The advantage of this is that the word vector can be adjusted according to the data set. When the data set is relatively small, it is easy to overfit.

The input layer is to splice the word vectors of all words in a sentence into a matrix, each row represents the word vector of a word, and all sentences are padding into a length: seq_len

3.2.2. Convolutional layer
Since the convolution used by TextCNN is one-dimensional convolution, the width of the convolution kernel is consistent with the dimension of word embedding. The height h of the convolution kernel represents the number of words taken in each window. So the convolution kernel \( \omega \in \mathbb{R}^{h \times k} \). For each sliding window result \( C_i \) (scalar), the result of the convolution operation is

\[
C_i = f(\omega \cdot X_{i:i+h-1}) + b
\]

Where \( b \in \mathbb{R} \) and \( f \) is a nonlinear function.

Since the convolution operation is an operation in which the corresponding elements are multiplied and then added, the dimensions of \( \omega \) and \( X_{i:i+j} \) are the same. Since the dimension of \( \omega \) is \( h \times k \) and the dimension of \( X_{i:i+j} \) is also \( h \times k \), the dimension of \( X \) is \( (n - h + 1) \times h \times k \) (can be obtained by comparing with the dimension of \( c \)).

Since the length of the sentence sequence is \( n \) and the height of the convolution kernel is \( h \), there are a total of \( n-h+1 \) sliding windows. So the convolution summary result is \( c = [c_1, c_2, \ldots, c_{n-h+1}] \)

The size of each convolution kernel is filter_size*embedding_size, and filter_size is equivalent to the size of \( n \) in n-gram, generally [3-5], indicating that there is a word order relationship between several adjacent words. embedding_size represents the size of the word vector. After the calculation of each filter is completed, a column vector is obtained, which represents the features extracted from the sentence by the filter, and how many features can be extracted as many convolution kernels.

3.2.3. Pooling layer
The pooling operation is to extract the maximum value of the vector obtained by the convolution so that after the pooling operation, we get a num_filter-dimensional row vector, which is to connect the maximum value of each convolution kernel. Another advantage of this is that if we did not pad the sentence before, the sentence length is different, and the column vector dimensions obtained after convolution are also different. Pooling can be used to eliminate the difference in length between sentences.

3.2.4. Fully connected layer
In order to convert the output vector of the pooling layer into the prediction result we want, a softmax layer is added. You can use dropout and L2 regularization to prevent overfitting.

3.3. Data enhancement

Text data enhancement is different from data enhancement in the image domain because the text is discrete, and the image belongs to a continuous space. For example, for two pictures, another picture can be constructed by linear interpolation, rotation, or SMOTE; however, for text data, suppose that \( x_1, x_2, x_1', x_2' \) represents a sentence, and pass. The linearly interpolated sentence may not exist at all, or the existence of the constructed sentence may be satisfied. However, it may be that a small disturbance affects the semantic information of the entire sentence, so the text data enhancement is difficult.

This article uses back translation for data enhancement. In this method, by calling the Baidu API method, the data set is translated into English one by one and then translated back to Chinese and stored in the txt text of the data set to quickly generate some inaccurate translations. As a result, the purpose of data enhancement is achieved.

3.4. Evaluation index

The evaluation criteria used in this article are Recall, Precision, F1-measure.

The statistics of the classification and labeling results of the text classification for the category \( c_i \) are shown in Tab. 1.

| Relationship | Classification judgment | belong | Does not belong |
|--------------|--------------------------|--------|-----------------|
| Mark as "yes"| a                        | b      |                 |
| Mark as "no" | c                        | d      |                 |

**Tab.1** Text classification for classification and labeling results statistics for category \( c_i \)

The meanings of the symbols in the above figure and table are as follows:
1) \( a \) represents the number of texts that correctly label the test set text as category \( c_i \);
2) \( b \) represents the number of texts that incorrectly label the test set text as category \( c_i \);
3) \( c \) means the number of texts that have been excluded from the category \( c_i \) in the test set by mistake;
4) \( d \) represents the number of texts that correctly exclude the test set text outside the category \( c_i \)

The recall rate (also called recall rate) of the classifier in category \( c_i \) is defined as:

\[
\text{recall}_i = \frac{a}{a + c} \times 100\%
\]

(3)

The accuracy of the classifier in the category \( c_i \) (also called precision) is defined as follows:

\[
\text{precision}_i = \frac{a}{a + b} \times 100\%
\]

(4)

The F1 value of the classifier in category \( c_i \) is defined as follows:

\[
F_{1i} = \frac{2 \times \text{precision}_i \times \text{recall}_i}{\text{precision}_i + \text{recall}_i}
\]

(5)

4. Experimental Results And Analysis
4.1. Experimental platform and data set
Experimental environment: Windows operating system, Intel(R) Xeon(R) CPU E5-16200@ 3.60GHz 3.60 GHz processor, 12GB memory, no GPU acceleration. 30,000 news headlines were extracted from THUCNews, and the text length was between 20 and 30. Simulate insufficient data sets. There are a total of 10 categories, each with 3000 items.

Data set division: training set 10,000, validation set 10,000, test set 10,000.

Enter the model in units of words, using pre-trained word vectors: Sogou News Word+Character 300d.

Categories: finance, real estate, stocks, education, technology, society, current affairs, sports, games, entertainment.

4.2. Analysis of experimental results
The experiment uses textCNN text classification to train and verify the data set using back translation for data enhancement, which proves that a better training effect can be achieved when the number of data sets is small. The results are shown in Tab.2:

| Category   | precision | recall   | f1-score |
|------------|-----------|----------|----------|
| finance    | 0.8947    | 0.6800   | 0.7727   |
| realty     | 0.8913    | 0.8283   | 0.8586   |
| stocks     | 0.6015    | 0.8000   | 0.6867   |
| education  | 0.9381    | 0.9192   | 0.9286   |
| science    | 0.6827    | 0.7100   | 0.6961   |
| society    | 0.8817    | 0.8200   | 0.8497   |
| politics   | 0.6783    | 0.7800   | 0.7256   |
| sports     | 0.9524    | 0.6000   | 0.7362   |
| game       | 0.7500    | 0.8700   | 0.8056   |
| entertainment | 0.7593 | 0.8283   | 0.7923   |

Tab.2 Results after data enhancement

5. Conclusion
This article is based on the text classification function implemented by the convolutional neural network TextCNN. It uses the back translation in the data enhancement to translate the data set into English and then translates back to Chinese. The data set is doubled in number, and the sample size is simulated. In this case, the use of data augmentation can expand the training set. Let the model achieve a better training effect so that it can show higher accuracy in the test set.

Acknowledgments
This work was supported by Hunan Provincial Education Department Foundation under grant No.18C1311.

BIGC Project (Ec202007).

References
[1] Salton G, Wong A and Yang CS 1975 A vector space model for automatic indexing Communications of the ACM 18(11) pp 613-620(doi: 10.1145/361219.361220)
[2] Joulin A Grave E and Bojanowski P et al 2017 Bag of tricks for efficient text classification. Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics (Valencia, Spain) pp 427-431
[3] Xueliang H, Xin L and Yuanping C 2020 A short text classification model based on a mixture of multiple neural networks Computer System Applications 29(10) pp 9-19
[4] Xiaozheng D, Rui S, Hongyu and et al 2018 News headline classification based on multiple models J. Journal of Chinese Information Processing 32(10) p 69
[5] Xingyu L and Py torch 2017 M. First edition Electronic Industry Press
[6] Dingpeng D, Yajian Z, Junhui C and et al 2020 Overview of Short Text Classification Technology Research J. Software 41(02) pp 141-144
[7] Jing W, Lang L and Deqiang W 2018 Research on Chinese short text classification based on word2vec J. Computer system applications 7(05) pp 211-217
[8] Jian L and Qian Y 2018 Overview of Convolutional Neural Networks J. Computer Times 317(11) pp 19-23
[9] Kim Y 2014 Convolutional neural networks for sentence classification Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (Doha, Qatar: EMNLP) pp 1746-1751
[10] Zhijie L, Chaoyang G and Peng S 2020 Study on Short Text Classification of LSTM-TextCNN Joint Model J. Journal of Xi’an Technological University 40(3) pp 299-304