Frontiers in Computing and Intelligent Systems
ISSN: 2832-6024 | Vol. 2, No. 2, 2022

Research on news text classification based on RCNN

Xudong Wang

Asterfusion Data Technologies Co., Ltd., China

Abstract: One of the popular directions of natural language processing is text processing, which is an effective method used to manage data, and it has developed rapidly after the rise of artificial intelligence. However, text classification based on traditional machine learning algorithms encountered bottlenecks in the development process, and after deep learning, a multi-level neural network, was introduced, it was soon widely used in natural language processing problems, making many branches of the field develop further, and now deep learning is the mainstream model in the field of text classification. In this paper, we first introduce the general process of text classification, point out the current problems of text classification based on traditional machine learning, and then describe the superiority of deep learning in this aspect of feature extraction. After that, three different deep learning models are experimented, and the differences between the trained models are compared and the relative optimal models are derived. Finally, the future trends and research directions of deep learning in this field are discussed.

Keywords: Text classification; Deep learning; Neural networks.

1. Background

Machine learning is an interdisciplinary discipline, mainly applied to natural language processing and image processing. Text classification is a branch of natural language processing, and its purpose is to organize and classify text data. In traditional machine learning-based text classification, the main algorithms are support vector machines and K-nearest neighbor algorithms, etc. However, machine learning-based text classification is a very important part of natural language processing. However, the machine learning-based text classification algorithms ignore the relationship between word vectors and sentences, and have poor processing and generalization ability for high-dimensional data, while this limitation is solved accordingly after the introduction of deep learning models. The essence of deep learning is multilayer neural network, and using multilayer neural network and combining with traditional machine learning algorithm, feature extraction of large amount of data can be done automatically by computer. Throughout the training process, model learning is automated, the design cost of the problem is reduced, and the data information analysis and extraction capability is enhanced. The mainstream deep learning models are convolutional neural networks (CNN), recurrent neural networks (RNN), attention mechanisms and other models. These models have achieved good experimental results in the field of text classification. In this paper, we will first give a detailed introduction to the traditional text classification process and show the problems of traditional methods, then describe three commonly used deep learning models and conduct experiments with them, and finally compare the optimal model among the three models by experimental data.

2. Methods and Models

There are two main types of text classification methods: traditional text classification; deep learning text classification.

For the former, traditional text classification should first select a dataset, and then manually construct feature engineering, which contains several major steps, including text pre-processing, feature extraction, and feature representation. Finally, a suitable machine learning algorithm is selected to classify the text after the feature engineering process. This process is described in detail below.

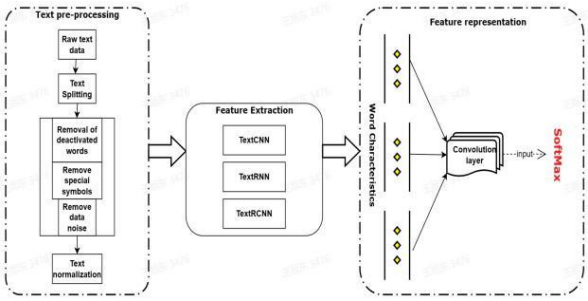

Figure 1. Structure of news text classification algorithm

2.1. Text pre-processing

The core steps of text pre-processing are text splitting and text cleaning. Firstly, the original text data collection is divided into a single chapter-based collection, and then the text is divided into words according to certain rules for extracting the feature values of the text through the text splitting process. The dictionary is constructed first, and then the text is divided into words using a dictionary algorithm. The current popular method for constructing dictionaries is dictionary trees, and the word separation algorithms can be divided into three categories: string matching-based word separation algorithms, such as forward maximal matching algorithms; statistical-based word separation algorithms such as CRF and N-gram (N-gram); and understanding-based word separation methods, i.e., artificial intelligence methods. Finally, text cleaning, i.e., removing noise from the data, such as discontinued words, special symbols, and those words or characters that do not characterize this text. The idea of the forward maximum matching algorithm is as follows: Initialize the word string S1 to be cut, output the word string S2, maximum length M, then:

while(S1 is not empty){
    Fetch the candidate string W whose length is less than M.
    Query whether W is in the dictionary.
    Remove the rightmost word of W.
if(W is a single word){

$S2 = S2 + W$
$S1 = S1 - W$
}
}

2.2. Feature Extraction

The purpose of feature extraction is to reduce the number of words to be processed by the model as much as possible without damaging the core information of the text, so as to reduce the vector space dimension and thus simplify the operation.

2.2.1. Machine learning based feature extraction

Feature extraction includes feature selection and feature weight calculation. The basic idea of feature selection is to rank the original feature items according to a certain evaluation index, select the highest scoring feature items, and filter the rest. For feature weighting, the TF-IDF algorithm is generally chosen to evaluate the importance of a word to a document set. When the word frequency and inverse document frequency are available, these two words are multiplied together to obtain the value of TF-IDF for a word, and the larger this value is, the greater the probability that the word will become a keyword. This calculation can effectively avoid the influence of common words on keywords and improve the relevance between keywords and articles.

$$tf_{ij} = \frac{n_{ij}}{\sum_k n_{kj}} \quad (1)$$

$$idf_i = \log(\frac{|D|}{1+|D_i|}) \quad (2)$$

$$tf * idf(i,j) = tf_{ij} * idf_i = \frac{n_{ij}}{\sum_k n_{kj}} * \log\left(\frac{|D|}{1+|D_i|}\right) \quad (3)$$

2.2.2. Deep learning-based feature extraction

The difference between machine learning and deep learning is the interpretability of features. In machine learning, each feature is required to have a certain meaning as much as possible, which is equivalent to manually extracting features from the original input, and then modeling with the extracted features. In contrast, deep learning learns features automatically from the data rather than selecting them manually. For all layers before the output layer, it can be viewed as a feature extraction process. After abstracting features by passing the original features through a multilayer neural network, the extracted features are then fed to the final layer for the corresponding operations. In this way, deep learning feature extractors such as TextCNN, TextRNN, etc. can be directly utilized instead of the above complex traditional machine learning feature extraction process.

TextCNN:

The word vector is first represented by e(wi).

Then a bidirectional GRU is used to obtain the context vector representation cl(wi),cr(wi) for each word, and the original word vector and the context representation vector are aggregated through a fully connected network to form a new word vector representation.

After that, maximum pooling is used to obtain the final representation of the sentences.

Finally, softmax is used to do the classification work.

TextRNN:

Firstly, the word vector is represented by e(wi).

Firstly, the word vector of each word in the sentence is input to the bidirectional two-layer LSTM in turn.

Then the hidden layers at the last valid position in each of the two directions are stitched into a vector as a representation of the text.

Finally, softmax is used to do the classification work.

TextRCNN:

Firstly, the word vector is represented by e(wi).

Next, the word vector is passed through the bidirectional RNN to get cl(wi) and cr(wi) vectors.

Then the cl(wi), e(wi) and cr(wi) are spliced to obtain the new vector, which is fed to the fully connected network for integration.

The output of the fully connected network is then subjected to MaxPooling pooling operation.

Finally, softmax is used to do the classification work.

2.3. Feature representation

Since the length of text is not fixed, the text of indefinite length should be converted into a fixed length space, which needs to be represented after a word embedding process. If the words are represented as discrete vectors by the One-hot method, each word in a sentence of n words is converted into an n-dimensional sparse vector, so that a sentence is converted into an n × n sparse matrix.

As can be seen from the above process, the main problem of traditional classification is that the final text representation obtained is highly sparse and high latitude, which makes the model feature representation weak. In addition, feature engineering often requires manual creation, and the whole process is tedious. And one of the important reasons why deep learning has been so successful in the image field is that the original data of images are continuous and dense with local relevance. It is envisioned that if deep learning is applied to text classification with high local relevance the problem of large-scale text representation can be solved.

3. Experimental results

Test set: 200,000 news headlines from THUCNews with text lengths between 20 and 30, 100,000 categories in total, 20,000 items per category.

Operating system: Windows 10

Compiler: PyCharm Community Edition 2020.2.3 x64

Deep learning environment: Pytorch 1.6.0

Due to the large number of iterations, only the first three iterations are taken here for reference, and the status of each model iteration is as follows:

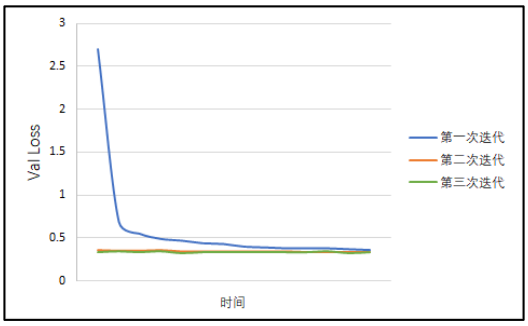

Figure 2. TextCNN loss descent graph: blue for the first iteration, yellow for the second, green for the third

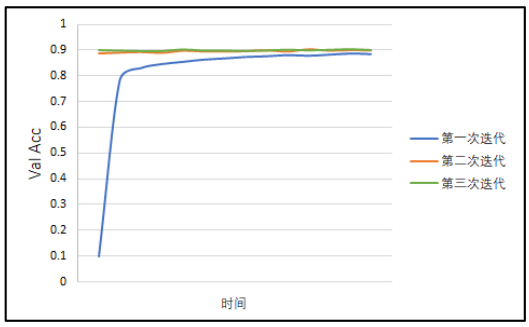

Figure 3. TextCNN accuracy transformation graph: blue for the first iteration, yellow for the second, and green for the third

The following results are obtained by averaging the first three iterations using the TextCNN model:

Table 1. CNN Mean

	Loss	Acc
First	0.586	81.39%
Second	0.3414	89.47%
Third	0.3307	89.89%

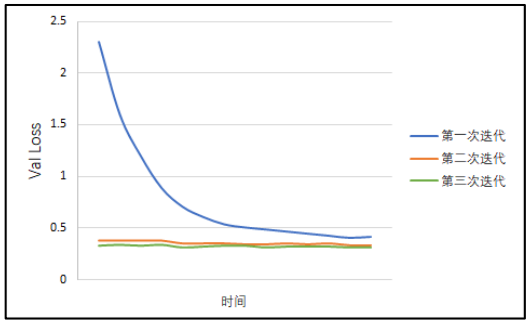

Figure 4. TextRNN accuracy transformation graph: blue for the first iteration, yellow for the second, green for the third

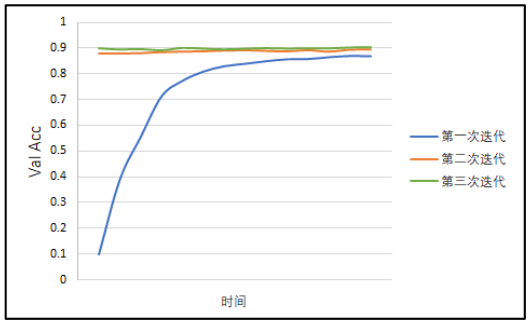

Figure 5. TextRNN accuracy transformation graph: blue for the first iteration, yellow for the second, green for the third

The following results were obtained by averaging the first three iterations using the TextRNN model.

Table 2. RNN average

	Loss	Acc
First time	0.7613	73.55%
Second time	0.3536	88.75%
Third time	0.3229	89.75%

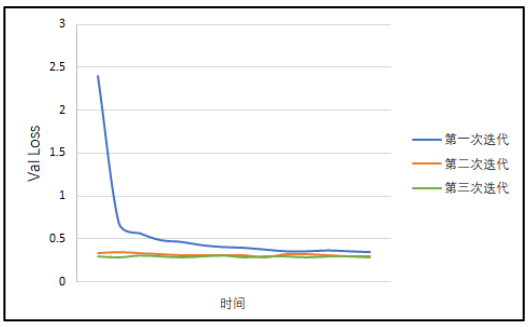

Figure 6. TextRCNN loss descent graph: blue for the first iteration, yellow for the second, green for the third

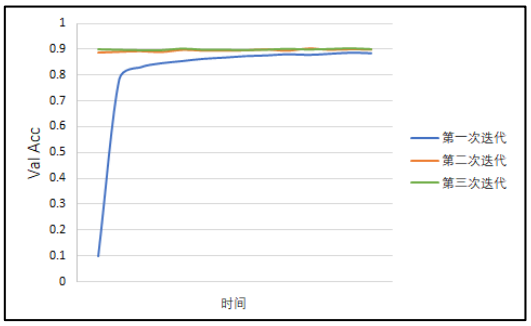

Figure 7. TextRCNN accuracy transformation graph: blue for the first iteration, yellow for the second, green for the third

The average of the first three iterations using the TextRCNN model gives the following results:

Table 3. RCNN average

	Loss	Acc
First	0.5507	81.72%
Second	0.3143	89.94%
Third	0.2979	90.59%

A Confusion Matrix is a situation analysis table that summarizes the prediction results of classification models in data science, data analysis and machine learning. It summarizes the records in a data set in matrix form according to two criteria: the true categories and the classification judgments made by the classification models. The confusion matrix for each model is as follows.

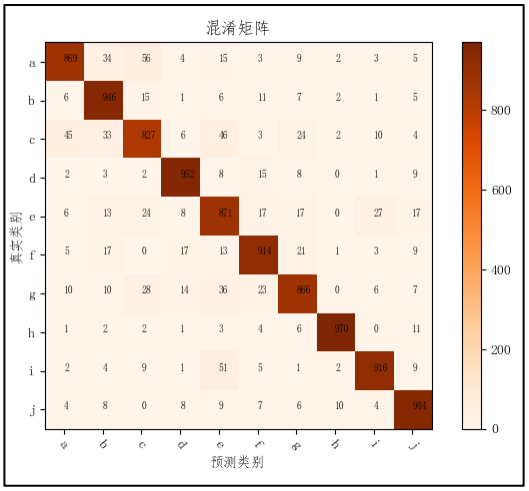

Figure 8. Confusion matrix of TextCNN

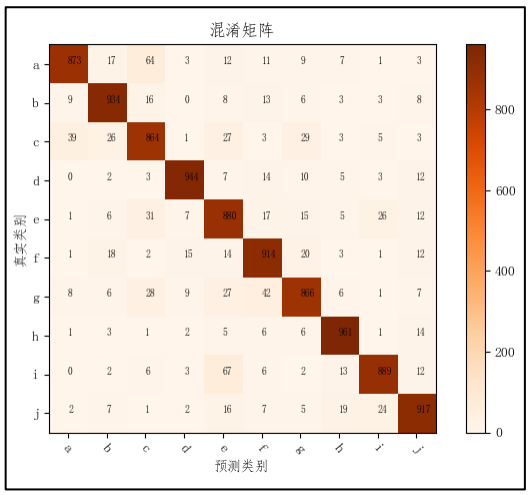

Figure 9. Confusion matrix of TextRNN

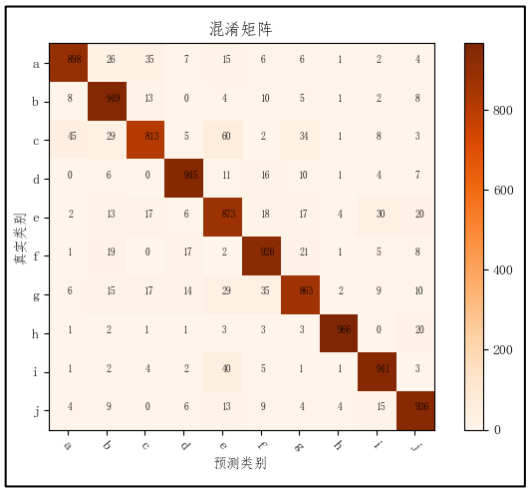

Figure 10. Confusion matrix of TextRNN

The confusion matrix shows that the percentage of correct predictions is higher in the total sample, with the TextRCNN model performing slightly better.

Cross Entropy Loss is often used when solving classification problems using neural networks, and it is often used to determine how close the actual output is to the desired output. the formula for implementing Cross Entropy Loss in PyTorch is:

$$LogSoftmax(x_i) = \log\frac{\exp(x_i)}{\sum_j \exp(x_j)}$$

The accuracy rate is calculated as:

$$Acc = \frac{TP + TN}{TP + FP + TN + FN}$$

Analyzing the results of the first three iterations of each of the three models, we can see that the Loss and Acc values (accuracy refers to the percentage of correct predictions over the total sample) are not very good after the first iteration, but all models have a significant improvement after the second iteration, with lower loss function values than the first one and a corresponding increase in accuracy. The difference between the second and third iterations was not significant, but the improvement was generally steady. Finally, after completing all iterations, the results of the three models TextCNN, TextRNN, and TextRCNN were obtained with loss values of 0.3, 0.29, and 0.28, respectively, and accuracy rates of 90.42%, 90.75%, and 91.10%, respectively. The experimental results are then compared to obtain the following results:

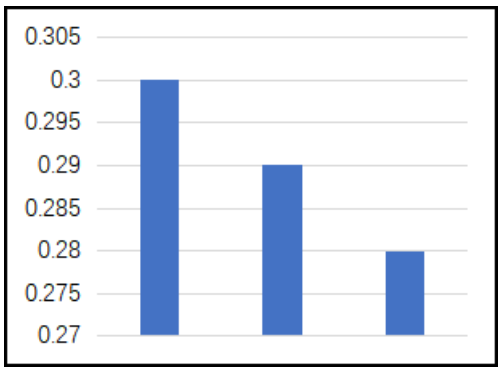

Figure 11. Three models Loss values

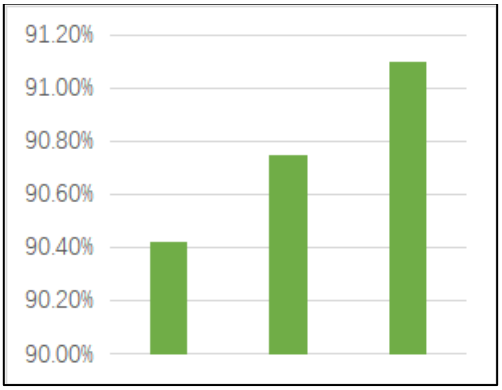

Figure 12. Final prediction accuracy of the three models, left to right for TextCNN,TextRNN,TextRCNN

4. Summary and Outlook

Through the above confusion matrix and experimental charts, the Loss value is lower than CNN and RNN 0.02, 0.01, and the accuracy rate is higher than CNN and RNN 0.68%, 0.35%, indicating that the TextRCNN model performs better than RNN and CNN, is easier to train, and the applicable range should be broader. However, the model should not be selected blindly in the actual training, but should be analyzed for specific problems before choosing a suitable model.

Traditional machine learning requires a lot of pre-processing work in text classification, and when text features are extracted, more shallow features are obtained. Deep learning models are generally better than machine learning for text classification, but there are differences between different models and they all have their own specialties, and the above experiments are also concluded by comparison. In the real problem, how to choose the right model is also an issue that needs attention. The significant feature of deep learning model is to extract more deep text feature information, reduce the loss of document information and improve the accuracy of text classification, but gold is not perfect, deep learning model is far from perfect, the main problem is still the need for massive data scale, and how to design a new deep neural network that can complement the advantages of the model after getting considerable data may be A research direction that needs to be paid attention to. There is no doubt that the Internet will continue to flourish in the future, and this is the driving force behind the development of new technologies, so it can be expected that more perfect new technologies will be applied to the field of artificial intelligence in the future.

References

[1] Minaee S , Kalchbrenner N , Cambria E , et al. Deep Learning Based Text Classification: A Comprehensive Review[J]. 2020.

[2] LightRNN: Memory and Computation-Efficient Recurrent Neural Networks[J]. 2016.

[3] Kadlec R , Schmid M , Ba Jgar O , et al. Text Understanding with the Attention Sum Reader Network[J]. 2016.

[4] Tu Z , Liu Y , Shang L , et al. Neural Machine Translation with Reconstruction[J]. 2016.

[5] Wan, J. Shan, Wu, Y. C.. A review of research on text classification methods based on deep learning[J]. Journal of Tianjin University of Technology,2021,37(02):41-47.

[6] Wu, Longfeng. Research on news text classification based on deep learning[D]. Anhui University of Technology,2020.

[7] Clark K , Luong M T , Le Q V , et al. ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators[J]. 2020.

[8] Jia, Pengtao, Sun, Wei. A review of deep learning-based text classification[J]. Computers and Modernization,2021(07):29-37.

[9] Wang Guihong. Research on Chinese text classification based on deep learning[D]. Xi'an University of Technology,2020.

[10] Minaee S , Kalchbrenner N , Cambria E , et al. Deep Learning Based Text Classification: A Comprehensive Review[J]. 2020.

[11] Wang Yibin. Deep learning-based news text classification and application[D]. Chongqing Normal University,2020.