International Core Journal of Engineering Volume 7 Issue 11, 2021
ISSN: 2414-1895 DOI: 10.6919/ICJE.202111_7(11).0080

Sentiment classification of English movie reviews based on deep learning

Wenjie Zhao

Jiangsu University of Technology/Troy University, Zhenjiang, Jiangsu, China

1446568118@qq.com

Abstract

With the development of online social network, an increasing number of people choose to share their views on films and television works on the Internet, which provides a more convenient way for film and television investors to understand the audience's feedback on the film. The film reviews on Doban (a social networking platform), for example, include a large number of users' sentimental views, which reflects audiences' positive or negative attitude towards films and TV works. Therefore, analysing the emotional tendencies of Douban film reviews can help investors make wiser decisions and improve the quality of works. As a large amount of data analysis can be done with the help of computer technology, this paper uses sentiment classification to deal with the data. Sentiment classification is a technique of natural language processing, which is often used to analyse and judge the type of sentiment described in the text. Therefore, it is also called sentiment analysis. In order to conduct the sentiment analysis based on deep learning, this paper adapts the convolutional neural network (CNN) model to handle the data. Finally, the result of F1 score = 0.9017 is obtained, which is satisfactory.

Keywords

deep learning, CNN, Sentiment classification.

1. Introduction

Sentiment classification, also known as sentiment orientation analysis, is a process of analysing, processing, summarising and reasoning subjective texts with emotional color. For example, it can identify the audience's praise and criticism of movies from various reviews, conclude the user's evaluation of products, and even it can predict the stock trend.

Sentiment classification aims to assign labels to sentences or documents. It is one of the fundamental task in terms of Natural Language Processing. The task on which this paper focuses is sentiment classification, which is an important direction of text classification [1]. Sentiment classification refers to building a model which can tell whether a text demonstrates positive or negative sentiment of the writer.

The main purpose of sentiment classification is to identify people's opinions and attitudes towards others or products. Currently sentiment classification is mainly used to analyse the user's evaluation of products on e-commerce shopping websites. Generally, buyers' attitude towards goods is relatively fixed, while the audience's evaluations of film works include the evaluation of actors, directors and screenwriters. Those are the evaluations of people. Compared with users' evaluation of objects, audiences' evaluation of people is relatively more complex and variable. Therefore, it is more difficult to analyse the emotion of film reviews. Studies by many researchers have also confirmed this point, that is, the accuracy of using sentiment classification to identity movie reviews is lower than using the same model to analyse other types of data.

ISSN: 2414-1895 DOI: 10.6919/ICJE.202111_7(11).0080

Natural language processing includes techniques such as automatic question answering, machine translation, information extraction, etc.. Sentiment classification is one branch of natural language processing. Natural language processing can be

carried out from different levels of text, including character level, word level and sentence level. The proposal of word embedding vector enables words to obtain the semantic expression of words through learning, thereby providing a more effective method for emotion classification based on positive or negative semantics. Specially, this paper solves the problem of sentiment classification for English movie reviews.

During the past few years, deep learning methods such as convolutional neural networks (CNN) [2] and recurrent neural networks (RNN) [3] become more and more popular for sentiment classification [4] [5] . To solve the problem of long text classification, some RNN-based methods are developed, like Long-Short Term Memory [6] and Gated Recurrent Unit (GRU) [7] , and have achieved promising scores on most of public datasets. Here this paper uses the bag-of-words model to represent semantics, and studies the feature extraction of convolutional neural network (CNN) model.

2. Relate Work

Text classification is a topic with a long history for Natural Language Processing, in which models need to assign predefined categories to free- text texts. The range of text classification research goes from selecting the best features from the data space to choose the best possible machine learning classiher [8] . Early machine learning methods depend on sentiment dictionary which provides sentiment information of each word in a text [9] . A sentiment classification method based on machine learning typically consists of two steps. At training step, we need to train the model on labeled training data set by machine learning algorithms such as k Nearest Neighbors (KNN) [10] and Support Vector Machine (SVM) [11], by which we get the classification model. At test step, the trained classification model is called for sentiment classification. Machine learning based methods often use sentiment dictionary as an external tool to get the emotional tendency of each word. Ultimately, these models determine the sentiment polarity of the target text by analyzing modihers, distances between texts, negative words, etc..

During the recent years, with the development of computational power and ton of data resources from the Internet, a lot of works turn to deep learning methods which exploit the deep meaning of each text and have obtained noticeable improvement. Mikolov etc. proposed Word2vec [12] to obtain vector representation for each possible word in text. By using vectors to represent words, deep learning models are enabled to better understand the semantic relations between words. TextCNN uses convolutional neural networks to extract feature vector of sentence meaning [2]. It utilizes layers of convolutional hlters that are applied to local features, and thus obtain multiple feature maps and extract features for multiple classification tasks. Besides, Long-Short Term Memory network is proved eVective for text classification tasks [6] [13]. Bidirectional Long-Short Term Memory (BiLSTM) extends the capability of LSTM to tackle the long-distance dependencies problem which RNNs cannot handle [14]. For every word in a given sentence, BiLSTM is able to obtain complete, sequential information about all words before and after it. Although it solves the problem of sentiment analysis for to a certain extent. As attention

mechanism becomes popular and is proved eVective for Computer Vision problems, Transformer applies an attention-based approach that allows the model to pay more attention to important information in sentences [15]. Besides, a part of recent works focus on leveraging knowledge into model to help the model make decisions [16] [17].

The techniques of English text classification has also made progress during the past years. Long-Short Term Memory (LSTM) is one of the early breakthrough in terms of text classification, which uses cell states to memorize long term information of a sentence [6]. It is extended by BiLSTM to catch context information of each word both forward and backward. Gated Recurrent Unit (GRU) is a variant of LSTM, which reduces several parts from LSTM and obtains comparable results with less

ISSN: 2414-1895 DOI: 10.6919/ICJE.202111_7(11).0080

computational costs [7] [18]. Hierarchical Recurrent Neural Network makes another big step for text classification [19]. It separates long texts into several sentences and obtains sentence representations before calculates the overall text representation. The most recent work in this domain is Transformer and its variant models. By leveraging positional embedding instead of processing text in sequence, these models are able to exploit global context information of each word in a text [15][20] . Many recent works extend the capability of BERT and get even better performance [21][22][23][24].

3. Model

3.1 Problem Definition

In the problem of sentiment classification, the dataset can be denote as a set $\mathcal{D}^{\text{train}} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{n}$, where n is the size of the dataset. Each piece of data (xi, yi) is a pair of text text and its sentiment label. A text xi is denoted as a sequence of word token {xi1, xi2 . . . , xili }, where li is the length of the i-th text. The label of a text is either "positive" or "negative" . The target of a sentiment classification model is to predict the label of given texts from the test set.

3.2 Overall Structure

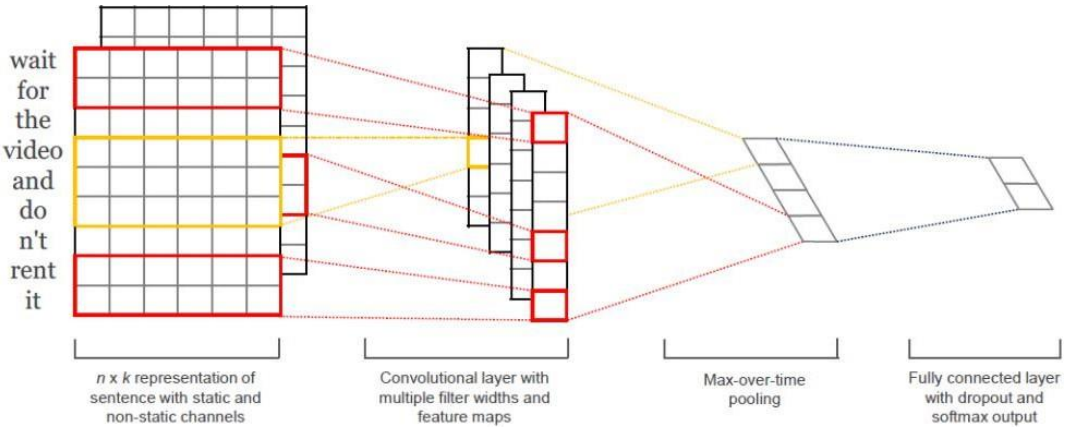

Fig. 1 Overall Structure

Suppose that we have some sentences that need to be classified. Each word in a sentence is composed of n-dimensional word vectors, which means that the input matrix size is m*n, where m is the sentence length. Then we use the CNN model to convolute the input samples. For text data, the filter no longer slides horizontally, but only moves downwards, which is similar to the local correlation between words extracted by N-grams. There are three types of step strategies in the above graph, which are 2, 3, and 4. Each step has two filters (there will be many filters in actual training). Then we get six convoluted vectors by applying different filters on different word windows. After each vector is maximized and pooled, and each pooled value is spliced, the feature representation of the sentence can be obtained. The sentence vector is thrown to the classifier for classification. So far, the whole process is completed.

3.2.1 Embedding Layer

Through the bag-of-words model, the semantic vector is extracted and spliced to form an embedding matrix.

3.2.2 Convolution Layer

When processing image data, the width and height of the convolution kernel used by CNN are the same. However, in text-CNN, the width of the convolution kernel is consistent with the dimension of the word vector. This is because each line vector we input represents a word, and in the process of feature extraction, the word is regarded as the smallest granularity of the text. Therefore, when we use convolution kernel for convolution, we consider not only word meaning but also word order and the context.

ISSN: 2414-1895 DOI: 10.6919/ICJE.202111_7(11).0080

3.2.3 Pooling Layer

Due to the convolution kernels of different heights we used in the convolution layer, the vector dimensions we get after passing the convolutional layer will be inconsistent. So in the pooling layer, we use 1-Max-pooling to pool each feature vector into a value. In other words, we extract the maximum value of each feature vector to represent its feature, and define that the maximum value represents its most important feature. When we do 1-max-pooling for all eigenvectors, we need to splice each value together. After we conduct 1-Max-Pooling on all feature vectors, we also need to splice each value together. Then we get the final feature vector of the pooling layer. Dropout can be added before the pooling layer to the fully connected layer to prevent over fitting.

3.2.4 Fully connected layer

The fully connected layer is the same as other models. Assuming there are two fully connected layers, the first layer can add 'relu' as the activation function, and the

second layer uses the softmax activation function to get the probability of each class.

3.3 Classification

To obtain the sentiment label from the overall feature vector of a text, a traditional fully-connected network is applied to obtain the likelihood of the text belonging to each class. Formally, we calculate the 3-dimensional logit vector

$$\mathbf{l} = W_c \mathbf{f} + \mathbf{b}_c$$

where Wc is a 3 × k matrix and bc is a 3-length vector. Wc and bc are both trainable parameters. The result l1, l2, l3 means the likelihood that the text is positive, neutral and negative, respectively. A softmax function is then applied to convert the likelihood vector into a probability vector:

$$p_i = \frac{e^{l_i}}{\sum_i^3 e^{l_i}}$$

pi stands for the probability of the text belonging to the i-th class.

3.4 Optimization

The total model is trained to optimize parameters to minimize the cross-entropy loss function

$$\min_{W_{conv}, b_{conv}, W_c, b_c} \mathcal{L} = -\sum_i^3 y_i \log p_i$$

As is adopted in most deep learning methods, gradient descent approach is used for minimize the target loss function. That is, at every step, each parameter W is updated by

$$W' = W - \frac{\partial \mathcal{L}}{\partial W}$$

After obtains the optimized parameters on the training set, the model is then evaluated on the test set.

4. Experiments

4.1 Dataset

The dataset we use is an movie dataset. The dataset contains 25000 samples from movies reviews annotated either as positive or negative. Among them, 12500 have positive tendency and the remaining 12500 have negative tendency. The statistics of Weibo dataset is shown in Table 1.

4.2 Data pre-processing

Firstly, we need to denoise the text. In this step we use beautifulsoup to remove the noise such as url and only keep the text. Then we adapt nltk to remove the stop words. After removing the stop words, we utilise nltk's lemmatizer to restore the part of speech. After these steps, we get the cleaned text.

ISSN: 2414-1895 DOI: 10.6919/ICJE.202111_7(11).0080

Next, we use the BOW algorithm of sklearn to convert the semantic features into semantic vectors, and finally save them into a matrix that conforms to the network dimension.

Table 1. Statistics of Weibo Dataset

	Positive	Negative
Training set	919	223
Development set	306	73
Test set	306	73
In Total	1,531	369

4.3 Evaluation

All the models experimented in this paper are evaluated in a unihed process. First the training set is used to train the parameters of the model. The validation set is used to observe whether the model weights are adjusted optimally. When the metrics of the model on the validation set has not changed for a several epochs or the specihed number of iterations has been reached, training is stopped. The test set is used as a hnal test. All the following experimental results are results on the test set.

4.3.1 Precision

Precision measures to what extent a model can avoid making wrong decisions. For each class, we calculate the fraction of right predictions among all the samples predicted as the class. The formula is as follows.

$$\text{Precision}_i = \frac{|\{\text{texts in class } i\} \cap \{\text{texts predicted as class } i\}|}{|\{\text{texts predicted as class } i\}|}$$

4.3.2 Recall

Recall measures to what extent a model can retrieve as many as possible samples of a class. For each class, we calculate the fraction of right predictions among all the samples in the class. The formula is as follows.

$$\text{Recall}_i = \frac{|\{\text{texts in class } \mathbf{i}\} \cap \{\text{texts predicted as class } \mathbf{i}\}|}{|\{\text{texts in class } \mathbf{i}\}|}$$

4.3.3 F1-score

F1-score is defined as the harmonic mean of precision and recall. The formula is as follows.

$$F1_i = \frac{2*\text{Recall}_i \ \text{Precision}_i}{\text{Recall}_i + \text{Precision}_i}$$

After obtain the three metrics above of all the three classes, we get the hnal metrics by calculate the mean value among all the classes.

4.4 Experimental results

The experimental results on dataset are show in Table 2.

Table 2. Experimental Results of Movie Dataset

Model Name	Precision	Recall	F1-score
CNN	90.57%	0.8977	0.9017

Based on the above results, we can see that the CNN model successfully extracts the semantic features and the BOW algorithm also represent the semantics of the text. Each index has achieved satisfactory results.

5. Feature Work

In the future, we are going to do more ablation experiments to perfect our conclusion, such as using Word2vec or Glove or other embedding methods. Also we will try more models, such as LSTM, GRU, DPCNN and so on. If time allows, we will also adapt Bert, Xlnet or ERINE and other pre-training models for experiments.

ISSN: 2414-1895 DOI: 10.6919/ICJE.202111_7(11).0080

References

[1] Liu B. Sentiment Analysis and Opinion Mining. Synthesis Lectures on Human Language TechnologiesMorgan & Claypool Publishers. 2012.

[2] Kim Y. Convolutional Neural Networks for Sentence Classification. Eprint Arxiv 2014: 1746–1751.

[3] Elman JL. Finding structure in time. Cognitive science 1990; 14(2): 179–211.

[4] Kim Y. Convolutional Neural Networks for Sentence Classification. In: ; 2014: 1746–1751.

[5] Zhang X, Zhao J, LeCun Y. Character-level convolutional networks for text classification. In: ; 2015: 649–657.

[6] Hochreiter S, Schmidhuber J. Long short-term memory. Neural computation 1997; 9(8): 1735–1780.

[7] Chung J, Gulcehre C, Cho K, Bengio Y. Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv preprint arXiv:1412.3555 2014

[8] Zhang X, Zhao J, LeCun Y. Character-level convolutional networks for text classification. Advances in neural information processing systems

[9] 2015; 28: 649–657.

[10] Tsai ACR, Wu CE, Tsai RTH, Hsu JYj. Building a concept-level sentiment dictionary based on commonsense knowledge. IEEE Intelligent Systems. 2013; 28(2): 22–30.

[11] Altman NS. An introduction to kernel and nearest-neighbor nonparametric regression. The American Statistician 1992; 46(3): 175–185.

[12] Cortes C, Vapnik V. Support-vector networks. Machine learning 1995; 20(3): 273–297.

[13] Mikolov T, Sutskever I, Chen K, Corrado GS, Dean J. Distributed Representations of Words and Phrases and their Compositionality. Advances in neural information processing systems 2013: 3111–3119.

[14] Chen Y, Yuan J, You Q, Luo J. Twitter sentiment analysis via bi-sense emoji embedding and attention-based LSTM. In: ; 2018: 117–125.

[15] Zhang S, Zheng D, Hu X, Yang M. Bidirectional Long Short-Term Memory Networks for Relation classification. In: ; 2015; Shanghai, China: 73–78.

[16] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need. In: ; 2017: 5998–6008.

[17] Tian H, Gao C, Xiao X, et al. SKEP: Sentiment Knowledge Enhanced Pre-training for Sentiment Analysis. In: Jurafsky D, Chai J, Schluter N, Tetreault JR. , eds. Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020Association for Computational Linguistics; 2020:4067–4076

[18] Wang K, Shen W, Yang Y, Quan X, Wang R. Relational Graph Attention Network for Aspect-based Sentiment Analysis. In: Jurafsky D, Chai J, Schluter N, Tetreault JR., eds. Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020Association for Computational Linguistics; 2020: 3229–3238

[19] Tang D, Qin B, Liu T. Document Modeling with Gated Recurrent Neural Network for Sentiment classification. In: ; 2015: 1422–1432.

[20] Yang Z, Yang D, Dyer C, He X, Smola A, Hovy E. Hierarchical attention networks for document classification. In: ; 2016: 1480–1489.

[21] Devlin J, Chang M, Lee K, Toutanova K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In: ; 2019: 4171– 4186.

[22] Liu Y, Ott M, Goyal N, et al. RoBERTa: A Robustly Optimized BERT Pretraining Approach. CoRR 2019; abs/1907.11692.

[23] Clark K, Luong M, Le QV, Manning CD. ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators. In: OpenReview.net; 2020.

[24] Yang Z, Dai Z, Yang Y, Carbonell JG, Salakhutdinov R, Le QV. XLNet: Generalized Autoregressive Pretraining for Language Understanding. In: Wallach HM, Larochelle H, Beygelzimer A, d'Alché-Buc F, Fox EB, Garnett R., eds. Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada; 2019:5754–5764.

ISSN: 2414-1895 DOI: 10.6919/ICJE.202111_7(11).0080

[25] Zhang Z, Han X, Liu Z, Jiang X, Sun M, Liu Q. ERNIE: Enhanced Language Representation with Informative Entities. In: Korhonen A, Traum DR, M àrquez L., eds. Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long PapersAssociation. for Computational Linguistics; 2019: 1441–1451