

## A Method Of Emotional Analysis Of Movie Based On Convolution Neural Network And Bi-directional LSTM RNN

Shudong Li\*

<sup>1</sup>*College of Mathematics and Information Science  
Shandong Technology and Business University*

*Yantai, Shandong 264005, China;*

<sup>2</sup>*Institute of Intelligent Software, Guangzhou University,  
Guangzhou, Guangdong 510006, China;*

<sup>3</sup>*College of Computer National University of Defense Technology,  
Changsha, Hunan 410073, China.  
Email: leeshudong79@163.com*

Zhou Yan

*College of Computer National  
University of Defense Technology,  
Changsha, Hunan, China;  
Email: candle531@126.com*

Xiaobo Wu

*School of software engineering  
Yantai Vocational College,  
Shandong Yantai, 264670 China;  
Email: happywxb@163.com*

Aiping Li

*College of Computer National  
University of Defense Technology,  
Changsha, Hunan, China;  
Email: apli1974@gmail.com*

Bin Zhou

*College of Computer National  
University of Defense Technology,  
Changsha, Hunan, China;  
Email: binzhou@nudt.edu.cn*

**Abstract**—The movie recommendation system provides an important way for users to choose a movie. In order to avoid man-made operations in the movie recommendation, the emotional analysis in the recommended system provides a new way of thinking. The traditional methods were based on knowledge, statistics and the hybrid approach. However these methods could obtain effect only base on the dataset with small amount and the one with not rich semantics. Therefore, in this paper we propose a new method based on convolution neural network and bi-directional LSTM RNN for dataset with large amount and rich semantics. Our approach does not need artificial labeled data and syntactic analysis, and only uses a small labeled train dataset. Also we prove the effectiveness and the accuracy of the proposed method by comparing with the current models including CNN, LSTM and Bi-LSTM. The experimental results demonstrate the effectiveness of the proposed method.

**Index Terms**—*Emotion Analysis, Statistics, Convolution Neural Network, Bi-directional LSTM RNN.*

### 1. Introduction

With the development of Internet, the entertainment of people become rich and colorful. Online movies are the products appearing in this environment. High quality and accurate file recommendation is an important guarantee for a good user experience. Traditional movie recommendation

methods may loss precision if the data was large and semantically rich, or there was some human-made affect by black-box operations. Therefore, extracting the emotional information from the film critics, will reflect the film in the great value.

At present, the use of traditional machine learning algorithms is a common method for emotional analysis, like SVM, Native Bayes, Information Entropy etc. Above the most of methods can be classified three ways: supervised learning, semi-supervised learning and unsupervised learning. Summed up the previous works, the supervised learning methods used in the field of emotional analysis have achieved good results[1]. However, the supervised learning depends on a large number of artificial labeling dataset, this means that the experiment needs to pay a high cost of artificial annotation. On the contrary, it does not need to label data in unsupervised learning methods, which makes that the results are not completely credible. And the semi-supervised learning method is a compromise approach, which comprehensively uses a small number of labeled samples and a large number of unlabeled samples to analyze the emotion of the comment, and also takes the result and cost into account.

The most of studies still use label data such as emotional dictionaries and syntactic analysis for emotional analysis. These methods can effectively improve the accuracy. However, the need of a large number of artificial labeled data leads to the versatility of the methods in other languages and other areas. Therefore, in this paper, we propose a method

based on deep learning(Convolution Neural Network and Bi-directional LSTM RNN). Our approach does not need artificial labeled data and syntactic analysis results, and only uses a small labeled train dataset. The experimental results demonstrate the effectiveness of the proposed method.

## 2. Related work

### 2.1. Deep Learning

The deep learning is derived from the study of neural networks. The structure of the deep learning includes multi-hidden multi-layer sensors. And it generates more abstract high-level represent attribute classes or characteristics by combing low-level features to discover distributed feature representations of data[2]. With the success of the deep learning in the field of image and speech recognition, the method has gradually appeared in the field of natural language processing, and also achieved gratifying results.

Bengio et al.[3] proposed a method to construct binary language model, which divided the dimension disaster by learning a distributed representation for each word, this representation can make the model know the number of adjacent semantic sentences at the relevant exponential level. Collobert et al.[4] designed a system named SENNA, they used a common architecture for natural language processing to learn the task-related features without using prior knowledge. Mikolov[5] succeeded in reducing the training complexity of the deep learning by using Log-Bilinear model. Socher et al.[6] introduced the approach of semantic detection based on recursive Autoencoders, which can reduce the semantic similarity of two sentences.

In this paper, we use the Convolution Neural Network(CNN) to extract the features of the comment. And it colligates the relevance of the word before and after the word through the Bi-directional LSTM RNN(Bi-LSTM). Our model is more suitable for emotional analysis of comment.

### 2.2. Emotion Analysis

Bo Pang proposed the emotional analysis in 2002, and the view has received important attention especially in the filed of online comment. Hatzivassiloglou et al.[7] carried out the vocabulary level of emotion tendencies. In their papers, they extract adjective-relate from large-scale corpus, and analyze the emotional polarity of these adjectives by logical regression. Then, the adjectives are grouped by clustering. Their results reach 82% accuracy. Pang et al.[8] took the film commentary as the experimental corpus, used three classification methods of machine learning: native bayes, maximum entropy model and SVM model, and also drew on the traditional natural language processing in the text classification technology.

Tuney et al[9] used the point mutual information to determine the emotional polarity of the statement, and put forward a method that at first extracts the subjective sentence

and classes the emotion. The method uses the adjective seed set to score the words in the sentence, and then judges the emotional tendencies according to the aliquot. Lin et al.[10] used LSM model, JST model and Reverse-JST model to construct three unsupervised emotional analysis systems. However, due to the deep emotional analysis is bound to involve the analysis of semantics, as well as the phenomenon of emotional transfer in the text often appear, the method based on deep semantic emotional analysis is not ideal. Therefore, in this paper, we introduce a Bi-LSTM model in order to improve effectiveness of the deep semantic analysis.

## 3. Algorithms

In order to get better filming emotional analysis, we put forward a model based on CNN and Bi-LSTM. At first, we convert the film comment into a real number matrix representation, the matrix is input for the model. And output of the model is the emotional classification. The structure of the model is shown as Fig.1.

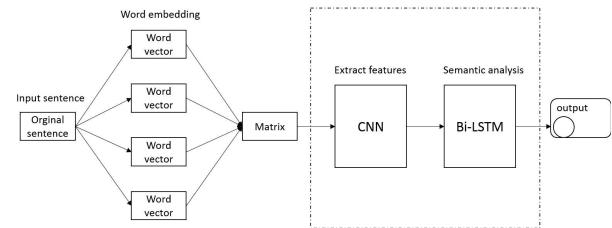


Figure 1. Structure of the model

### 3.1. Convolution Neural Network

The Convolution Neural Network(CNN) is a model that based on the deep neural network, in which the structure is a neural network with multi-layers. The standard CNN model consists of a convolution layer, a sample layer, a pool layer and a fully connected layer.

In the field of nature language processing, the structure of CNN is shown as Fig.2. The input is the matrix representation of the film review. The output of each layer is the input of the neurons of the next layer. Calculating through multi-layers convolution, the results from each convolution layer are nonlinearly converted[11].

The convolution layer consists of multiple filters of different window size. The number of filter is a super parameter which is confirmed by man-made. Each filter is a feature recognizer because the same filter parameter is shared. The mapping of different features obtained by different filter is called a feature map.

The input matrix is represented as follow:

$$X_{1:n} = X_1 \oplus X_2 \oplus \dots \oplus X_n \quad (1)$$

where n is the length of the review, the  $\oplus$  is concatenation operator. The concatenation of words  $X_i, X_{i+1}, \dots, X_{i+j}$  is expressed as  $X_{1:n}$ .

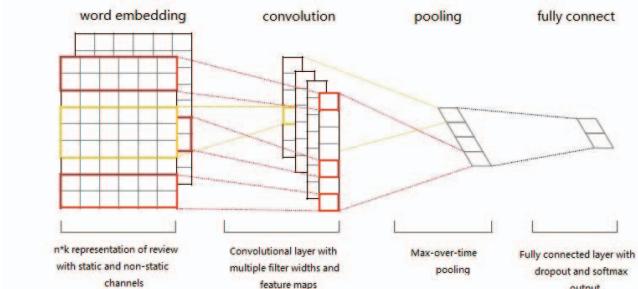


Figure 2. Standard CNN Model In The field of Nature Language Processing

The convolution calculate through a filter, the filter is expressed as  $w(W \in R)$ , which is applied a window of  $h$  words to produce a new feature. If the feature  $c_i$  is generated from a window size of  $h$ , the feature is calculated by:

$$c_i = f(W \cdot X_{i:i+h-1} + b) \quad (2)$$

where, the  $X_{(i : i + h - 1)}$  expresses the feature extracted from word  $X_i$  to  $X_{i+h-1}$ ,  $b$  is a bias term and  $f$  is a non-linear function. In this paper, we choose hyperbolic tangent as the non-linear function.

With different size of window in the sentence, to produce a feature map:

$$c = [c_1, c_2, \dots, c_{n-h+1}] \quad (3)$$

The pooling layer includes maximum pooling, average pooling and weighted pooling. After pooling operation, we not only reduce the computational complexity, but also reduce the sensitivity of rotation, distortion, and greatly further enhance the robustness of the model. The number of filters determines the dimensions of the pooling layer.

The fully connected layer, in fact, is the hidden layer of the three-layer neural network to the output layer mapping.

In summary, the traditional CNN is usually through a linear filter to get the feature map, and then through the non-linear activation. In order to extract more abstract features and consider the computation complexity of the training, in this paper, we add a cross-channel convolution layer after the convolution layer to improve the expression ability of the model:

$$c_{i:j,k_1}^1 = f(w_{k_1}^1 \cdot X_{i:j} + b_{k_1}) \quad (4)$$

$$c_{i:j,k_2}^2 = f(w_{k_2}^2 \cdot c_{i:j,k_1}^1 + b_{k_2}) \quad (5)$$

where, the equation (3) is the traditional convolution layer, the equation (4) is the cross-channel convolution layer. In fact, the cross-channel convolution achieves a weighted linear reorganization of the input feature map, so that cross-channel integration can be performed with the same resolution of the feature map to learn the interrelated information between the more complex different channels. The cross-channel convolution neural network is shown as Fig.3.

The cross-channel CNN model uses variable-length convolutional filters to obtain more feature information, and

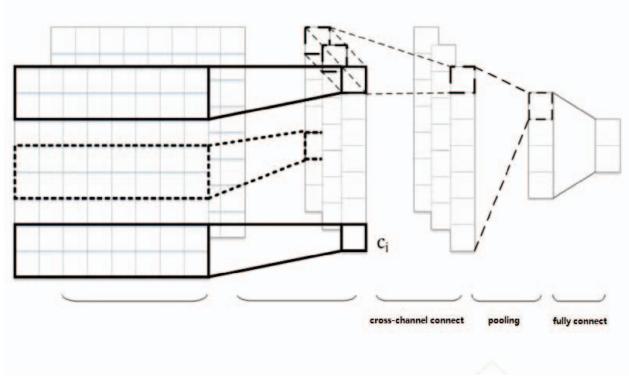


Figure 3. The Structure of Cross-channel CNN

adds a layer of cross-channel filter behind the linear convolution filter.

To avoid overfitting during the model training, Hinton[12] proposes to improve the performance of the neural network structure by Dropout. By randomly ignoring the neurons in the convolution layer, a large number of different network structures can be trained within a reasonable time to average the predicted probability.

### 3.2. Bi-directional LSTM RNN

The standard RNN processes the sequences in timing[13,14], they tend to ignore future contextual information. An obvious solution is to add a delay between the input and the target[15,16], which can add some future context information to the network. Based on this theory, the LSTM model is proposed. The LSTM model is an improvement of the RNN model by adding a component called LSTM unit in the RNN model. And the Bi-LSTM and LSTM in common is the LSTM unit. In all of these models above, the LSTM unit is capable of learning long-term dependencies without keeping redundant context information, and also works tremendously well on NLP tasks.

**3.2.1. LSTM unit.** The LSTM unit can store the history information by adding a memory unit. Through the input gate, the output gate and the forget gate can be updated by utilizing historical information. The structure of LSTM unit is shown as Fig.4.

As shown in Fig.4,  $x_t$  is the input of LSTM unit,  $c_t$  represents the value of the memory unit, and the output is  $h_t$ . The LSTM unit is updated according to the following six steps.

(1)The traditional RNN model calculates the candidate memory cell value  $\hat{c}_t$  of the current time as shown below:

$$\hat{c}_t = \tanh(w_{xc}x_t + w_{hc}h_{t-1} + b_c) \quad (6)$$

where,  $w_{xc}$  represents the weight of the input data in the LSTM unit, and  $w_{hc}$  represents the weight of the output in the LSTM unit for the last time.

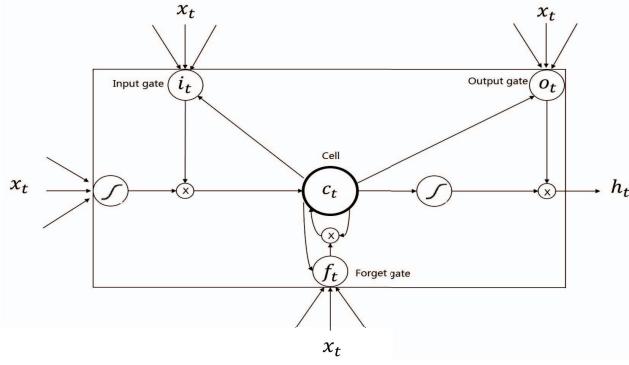


Figure 4. The Structure of LSTM Unit

(2) In the LSTM unit, the input  $i_t$  is calculated by the formula as:

$$i_t = \sigma(w_{xi}x_t + w_{hi}h_{t-1} + w_{ci}c_{t-1} + b_i) \quad (7)$$

$i_t$  indicates the value of the input gate, and the effect of the current input data on the memory cell status value is controlled by the input gate. The calculation of the input gate is called Peephole Connections, this scheme calculates the value  $i_t$  of the input gate from the current input data  $x_t$ , the output value  $h_{t-1}$  of the last time in LSTM unit, and the value of  $c_{t-1}$  of the previous time in memory unit.

(3) The value of the forget gate is denoted by  $f_t$ , the LSTM unit controls the historical information on the current memory cell state value by forget gate, and the value of the forget gate is calculated as:

$$f_t = \sigma(w_{xf}x_t + w_{hf}h_{t-1} + w_{cf}c_{t-1} + b_f) \quad (8)$$

(4) The following formula gives the calculation of the state value  $c_t$  of the current time in memory cell. The  $\odot$  means that elements and elements are calculated by multiplying points by point. The state  $\hat{c}_t$  of the candidate memory unit at the current time and the current state  $c_{t-1}$  of the current memory unit are updated,  $c_t$  and  $c_{t-1}$  can be controlled by the input gate and the forget gate.

$$c_t = f_t \odot c_{t-1} + i_t \odot \hat{c}_t \quad (9)$$

(5) The output gate is the gate used to control the memory unit status value, the value  $o_t$  of output gate as:

$$o_t = \sigma(w_{xo}x_t + w_{ho}h_{t-1} + w_{co}c_{t-1} + b_o) \quad (10)$$

(6) The output  $h_t$  of the LSTM unit is finally calculated based on the value  $o_t$  of the output gate.

$$h_t = o_t \odot \tanh(c_t) \quad (11)$$

The output of LSTM unit usually uses the sigmoid function, which ranges from (0,1).

**3.2.2. Bi-LSTM.** The Bi-LSTM model can be applied to extract contextual features from past and future. The Bi-LSTM is similar to LSTM but different. Bi-LSTM has parallel layers propagating in both directions, and the forward and backward propagation of each layer is performed in the form of a conventional neural network, through the two layers store information from both directions.

For the implicit layer of Bi-LSTM, the forward prediction is similar to the RNN. In addition to the input sequence being opposite to the two hidden layers, the output layer is updated until all of the two input layers have been processed:

```

For t=1 to T do
  Forward pass for the forward hidden layer,
  storing activations at each timestep
For t = T to 1 do
  Forward pass for the backward hidden layer,
  storing activations at each timestep
For all t, in any order do
  Forward pass for the output layer, using the
  stored activations from both hidden layers

```

The backward projection of the Bi-LSTM is similar to that of the RNN by time back propagation, except that all the output layer  $\delta$  are first calculated and then returned to the two different directions of the hidden layer:

```

For all t, in any order do
  Backward pass for the output layer, storing delta
  terms ai each timestep
For t = T to 1 do
  BPTT backward pass for the forward hidden layer,
  using delta terms from the output layer
For t=1 to T do
  BPTT backward pass for the backward hidden layer,
  using the stored delta terms from output layers

```

## 4. Experiments

According to the above-mentioned model based on CNN and Bi-LSTM. In order to verify the versatility of the model, we use the data of IMDB as dataset of the experiment.

### 4.1. Dataset

The IMDB dataset contains 25000 movie reviews that are marked by "positive" or "negative". The reviews are pre-processed to encode each comment as a word index(number) sequence. For convenience, the word is indexed based on the frequency of occurrence in the entire dataset, the "3" represents the third frequency in the data. This can quickly filter out the desired results. For example, some time it needs the words of top 10000, but excludes the words of top 20. At the same time, "0" does not represent a special word, but represents some unknown words.

In this experiment, 80% of the dataset is selected as the training set, 10% as the test set and 10% as the verification set. The distribution of the dataset is shown in the following table. The experimental platform uses Python 2.7.

TABLE 1. THE DISTRIBUTION OF THE DATASET

| dataset          | percentage | amount |
|------------------|------------|--------|
| training set     | 80%        | 20000  |
| test set         | 10%        | 2500   |
| verification set | 10%        | 2500   |

## 4.2. Model Design

In this experiment, we used CNN, LSTM, Bi-LSTM and our proposed model for comparative experiments, and we perform 1, 5, 10 and 30 iterations for each model.

For the CNN model, we choose that the max value of features is 5000, the dimension of embedding layer is 50. The deep learning algorithm is a gradient reduction. There are two ways to update each parameter. One method is traverse the entire dataset to calculate the loss function, and calculate the gradient of each parameter, and then update the gradient. This method will traversal the dataset, when the parameter is updated. This result is computed with a large amount of computation, but it could not support online learning. The method is called batch gradient descent. The other method is stochastic gradient descent, this method is faster, but the convergence performance is no good. In order to overcome the shortcomings of the two methods, we use a method called mini-batch gradient decent. Our method divides the data into several batches, and updates the parameters by batch. Using this method it can reduce both randomness and computational complexity. In this CNN model, we choose the number of batches is 32.

In the LSTM model, Bi-LSTM model and our model, the number of features and batch are same as CNN. In LSTM and Bi-LSTM model, we define that the number of LSTM unit is 128. The results are shown as TABLE 2. The model we proposed has better performance in both accuracy and recall than other models from the results.

## 5. Conclusion

In this paper, we propose an emotional analysis method based on Convolution Neural Network and Bi-directional LSTM RNN. The model improves the accuracy compared to CNN, LSTM and Bi-LSTM model. The performance of the algorithm is very close to the current performance of traditional algorithms.

There are still many problems that need to be further explored in the deep learning methods. For instance, the neural network model is more probably to fit the situation because this structure has lots of parameters in training. In the next step, we will focus on how to reduce the structural risk of the model, and to find a deep learning algorithm that is more suitable for emotional analysis.

## Acknowledgments

This research is supported by NSFC (No.61672020, 61662069, 61472433), Project funded by China Postdoc-

toral Science Foundation (No.2013M542560, 2015T81129), Project of Shandong Province Higher Educational Science and Technology Program (No.J16LN61) and Project of Yan-tai science and technology program (No.2016ZH054).

## References

- [1] TANG Hui-feng, TAN Song-bo and CHENG Xue-qi, Research on Sentiment Classification of Chinese Reviews Based on Supervised Machine Learning Techniques[J]. JOURNAL OF CHINESE INFORMATION PROCESSING 2007, 21(6): 88-94.
- [2] SUN Zhi-jun, XUE Lei, Xu Yang-ming and WANG Zheng, Overview of deep learning[J]. Application Research of Computers 2012, 29(8):2806-2810.
- [3] Socher R, Huang E H, Pennington J, et al. Dynamic Pooling and Unfolding Recursive Autoencoders for Paraphrase Detection[C] NIPS. 2011, 24: 801-809.
- [4] Collobert R, Weston J, Bottou L, et al. Natural language processing (almost) from scratch[J]. Journal of Machine Learning Research, 2011, 12(Aug): 2493-2537.
- [5] Mikolov T, Kombrink S, Burget L, et al. Extensions of recurrent neural network language model[C]//Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on. IEEE, 2011: 5528-5531.
- [6] Socher R, Manning C D, Ng A Y. Learning continuous phrase representations and syntactic parsing with recursive neural networks[C]//Proceedings of the NIPS-2010 Deep Learning and Unsupervised Feature Learning Workshop. 2010: 1-9.
- [7] Hatzivassiloglou V, McKeown K R. Predicting the semantic orientation of adjectives[C]//Proceedings of the eighth conference on European chapter of the Association for Computational Linguistics. Association for Computational Linguistics, 1997: 174-181.
- [8] Pang B, Lee L, Vaithyanathan S. Thumbs up?: sentiment classification using machine learning techniques[C]// Acl-02 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, 2002:79-86.
- [9] Turney P D. Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews[J]. Proceedings of Annual Meeting of the Association for Computational Linguistics, 2002:417-424.
- [10] Lin C, He Y, Everson R. A comparative study of Bayesian models for unsupervised sentiment detection[J]. Grb Coordinates Network, 2010, 12255:144-152.
- [11] Kalchbrenner N, Grefenstette E, Blunsom P. A Convolutional Neural Network for Modelling Sentences[J]. Eprint Arxiv, 2014, 1.
- [12] Hinton G E, Srivastava N, Krizhevsky A, et al. Improving neural networks by preventing co-adaptation of feature detectors[J]. Computer Science, 2012, 3(4):pgs. 212-223.
- [13] Member, IEEE, Schuster M, et al. Bidirectional Recurrent Neural Networks[J]. IEEE Transactions on Signal Processing, 1997, 45(11):2673-2681.
- [14] Shudong L, Zhongtian J, Aiping L, Lixiang L, Xinran L. Cascading dynamics of heterogeneous scale-free networks with recovery mechanism[J]. Abstract and Applied Analysis, Volume 2013, Article ID 453689,2013.13 pages.
- [15] Weiping W, Lixiang L, Haipeng P, Jrgen K, Jinghua X, Yixian Y. Anti-synchronization of coupled memristive neutral-type neural networks with mixed time-varying delays and stochastic perturbations via randomly occurring control[J]. Nonlinear Dynamics, 2016832143-2155.
- [16] Shudong L, Lixiang L, Yan J, Xinran L, and Yixian Y. Identifying Vulnerable Nodes of Complex Networks in Cascading Failures induced by Node-based Attacks[J]. Mathematical Problems in Engineering, 2013, Volume 2013, Article ID 938398.

TABLE 2. THE RESULTS OF DIFFERENT MODELS

|             | iterations  | 1       | 5       | 10      | 30      | average |
|-------------|-------------|---------|---------|---------|---------|---------|
| CNN         | accuracy(%) | 0.87804 | 0.89048 | 0.88768 | 0.88331 | 0.88487 |
|             | recall(%)   | 0.87661 | 0.87464 | 0.88297 | 0.87286 | 0.87677 |
| LSTM        | accuracy(%) | 0.81036 | 0.83376 | 0.84648 | 0.83696 | 0.83189 |
|             | recall(%)   | 0.78497 | 0.89115 | 0.86034 | 0.83977 | 0.84405 |
| Bi-LSTM     | accuracy(%) | 0.85368 | 0.82228 | 0.83208 | 0.81568 | 0.83093 |
|             | recall(%)   | 0.84838 | 0.75530 | 0.82638 | 0.80281 | 0.80821 |
| CNN+Bi-LSTM | accuracy(%) | 0.85991 | 0.89787 | 0.92843 | 0.91623 | 0.90061 |
|             | recall(%)   | 0.88525 | 0.90122 | 0.87243 | 0.94887 | 0.90194 |