Emotional Analysis of Cigarette Consumers Based on CNN and BiLSTM Deep Learning Model

Jingju Lin1a, Zhongquan Jian2

1Cigarette Manufacturing Department, Longyan Tobacco Industry Co., Ltd., Longyan, Fujian, 364000, China
2School of Mechanics, Fuzhou University, Fuzhou, Fujian, 350116, China
a.ljj23231@fjtic.cn  *Corresponding author’s e-mail: 313974227@qq.com

Abstract. Cigarette online reviews can truly reflect the word-of-mouth of cigarettes, and help cigarette industrial and commercial enterprises to understand consumers' cigarette use experience and cigarette word-of-mouth dynamics. In order to extract effective consumer experience information from massive online reviews of cigarette consumption, this paper studies the text sentiment analysis of cigarette online reviews. This paper presents a feature fusion model of convolutional neural network and BiLSTM. Experimental results show that the proposed feature fusion model effectively improves the accuracy of text classification. The model can provide new insight for the evaluation of cigarette management, dynamically monitor the change of consumers' emotion, and grasp the trend of consumers' emotion in the tobacco market environment in time.

1. Introduction

In the information age, more and more cigarette consumers evaluate their consumption online. As the feedback of consumers' personal experience, the comments of these texts contain a large amount of useful information [1]. On the one hand, the previous consumer evaluation of the product can help potential consumers to understand the product in advance. On the other hand, it can also be used as feedback information to help cigarette industry and commerce enterprise to understand consumers’ purchase intention, track after-sales service of commodities, and improve their competitiveness. Therefore, this paper analyses the emotional tendency of cigarette consumers' comments through text sentiment analysis, so as to provide decision support for cigarette manufacturers and enterprises [2].

Deep learning uses multiple processing layers, consist complex structures or multiple nonlinear transformations, to perform high-level abstraction of data [3]. Kalchbrenner et al. apply CNN to natural language processing, and designed a dynamic convolution neural network model to deal with different length text [4-5]. In the model of English text classification proposed by Kim, the pre-processed word vector is used as the input, and the convolution neural network is used to realize the task of sentence level classification [6].

However convolutional neural networks pay more attention to local features and ignores the context meaning of words. Therefore, Bidirectional Long Short-Term Memory (BiLSTM) network is used to solve the problem where the CNN ignores the context meaning of words in this paper.

For serialization input, recurrent neural network (RNN) can effectively integrate the neighbouring location information and deal with various tasks of natural language processing [7]. There are many kinds of RNN models, the Bi-directional RNN is mainly used in text classification. Since the semantic
information of words in text is not only related to the information before the word, but also related to
the information after the word, the Bidirectional recurrent neural network composed of two RNNs can
further improve the accuracy of text classification.

The main work of this paper is as follows:

i. Using BiLSTM to extract text context features, fully considering that the semantics of a
word in natural language is not only related to the information before it, but also related to the
information after it.

ii. Integrating BiLSTM and CNN to maintain the ability of extracting context features from text
processing by BiLSTM and extracting local semantic features by CNN. It can also understand
the semantics of the text to be processed very well, so as to improve the accuracy of emotional analysis of
the text.

2. Work vector

2.1. Word embedding
The first step of the deep learning method for text classification is to vectorize the text, hence, a word
vector is used to represent the text and regarded as the input of the convolutional neural network and
the BiLSTM network model. The traditional text representation method is based on vector space
model or one-hot representation. For vector space model, the vector dimension is linearly related to
the number of words in the dictionary, which is easy to cause dimensional disaster with the increase of
words. One-hot representation is easy to apply, but it ignores the semantic correlation between words.
The word vector is proposed to improve the defects of the vector space model and one-hot
representation, which maps high-dimensional sparse feature vector into low-dimensional dense word
vector. It can effectively avoid the occurrence of dimensional disaster and directly calculate the
semantic correlation between words.

As an illustration, Bengio et al. use neural network probabilistic language model (NPLM) to
process text information. Mikolov et al. proposed the Word2Vec model based on NNLM (Neural
Network Language Model), which constructs word vectors by Continuous Bag-Of-Words (CBOW)
and Skip-gram. Different from NNLM, the Word2Vec does not only use the first \(n - 1\) words to
predict the \(n\)th word, but also use the window with the size of \(n\) to calculate the probability of the
occurrence of the centre word in the window. Both CBOW and Skip-gram are based on the Huffman
tree. The initial value of the intermediate vector stored in the non-leaf node in the Huffman tree is a
zero vector, and the word vector of the corresponding word of the leaf node is initialized randomly.

CBOW predicts a word based on the context. The training process consists of three parts: input
layer, projection layer and output layer, shown as Figure 1. The input layer is the word vector of \(n - 1\)
words around the morphology \(w(t)\). For example, \(n\) is set to 5, the first two words of the word \(w(t)\)
are \(w(t - 2)\) and \(w(t - 1)\), the last two words are \(w(t + 1)\) and \(w(t + 2)\). Their corresponding vectors are
denoted as \(V(w(t - 1)), V(w(t - 2)), V(w(t + 1)), V(w(t + 2))\), respectively. From the input layer to the
projection layer, the vector forms of the four words are added directly. While from the projection layer
to the output layer, a Huffman tree is constructed. Starting from the root node, the values of the
projection layer are classified by logistic along the Huffman tree, and the middle vectors and word
vectors are constantly modified to get the word vector \(V(w(t))\) corresponding to the word \(w(t)\).

The Skip-gram model is just the opposite of CBOW, which is also comprised by input layer,
projection layer and output layer, shown as Figure 2. The input of Skip-gram is the vector form of the
current word \(w(t)\), and the output is the vector form of surrounding words. The surrounding words are
predicted by the current word. For example, the context window size is set to 4, the vector form
corresponding to the middle word \(w(t)\) is \(V(w(t))\), and \(V(w(t))\) is used to predict the vector form
corresponding to the surrounding 4 words, \(\text{context}(w) = \{V(w(t + 2)), V(w(t + 1)), V(w(t - 1)), V(w(t - 2))\}\).
The Skip-gram model calculates the surrounding word vectors by using the conditional probability value of the intermediate word vector $V(w(t))$. The formula is as follows:

$$P(V(w(i))|V(w(t)))$$

where $V(w(t)) \in \text{context}(w)$.

2.2. Word vector similarity
Comparing the word vector with the vector space model and one-hot, the dimension of the word vector has changed from a thousand-dimensional sparse vector to a low-dimensional dense vector form. The word vector contains the semantic and grammatical relationship in natural language. In addition, the word vector trained by Skip-gram model can calculate the semantic correlation between words more easily. The relationship between words is represented by the cosine distance between word vectors. The larger the cosine similarity value is, the greater the relationship between words is. The smaller the cosine similarity value is, the smaller the relationship between words is.

3. Work vector
3.1. BiLSTM model
The classic recurrent neural network (RNN) can mine the time series information and contextual semantic information of text. The perception for long ago information decreases as the input increases when RNN learns time series of any length. Generating the problems of long-term dependence and gradient disappearing. The Long Short Term Memory (LSTM) network improved from RNN can solve the problem of long-term dependence and gradient disappearance of RNN. Figure 3 shows the LSTM network model with three gated structures.

In Figure 3, $x$ represents the input at time $t$. $f_t$ is the forget gate, represents the output from time $t$ to time $t+1$. $o_t$ is the output gate, represents the output at time $t$. $h_{t-1}$ is the hidden layer, represents the output at time $t-1$. $\sigma$ denotes the Sigmoid function. $i_t$ denotes the input gate. $h_t$ is the hidden layer, represents the output at time $t$. $c_t$ represents the state at the current time. $\tilde{c}_t$ represents the intermediate quantity at time $t$.

The formula for each gate in LSTM is calculated as follows:

$$f_t = \sigma(W_f \times [h_{t-1}, x_t] + b_f)$$

$$i_t = \sigma(W_i \times [h_{t-1}, x_t] + b_i)$$

$$o_t = \sigma(W_o \times [h_{t-1}, x_t] + b_o)$$

$$h_t = o_t \times \tanh(c_t)$$

$$c_t = f_t \times c_{t-1} + i_t \times \tilde{c}_t$$

Figure 1. CBOW model.
Figure 2. Skip-gram model.
\[ \tilde{c}_t = \tanh(W_f \times [h_{t-1}, x_t] + b_f) \]  

where \( W_f \) represents the weight matrix of the connection of the forgetting gate. \( b_f \) represents the offset value of the forgetting gate. \( W_i \) represents the weight matrix of the input gate connection. \( b_i \) represents the offset value of the input gate. \( W_o \) represents the weight matrix of the output gate connection. \( b_o \) represents the offset value of the output gate. "\( \ast \)" means the multiplication of two matrix elements.

LSTM can only learn the above information of the text, but cannot use the below information of the text. Since the semantics of a word is not only related to the above information of the text, but also closely related to the below information of the text, BiLSTM is used to replace LSTM to introduce the below information. BiLSTM model is composed of two LSTM networks by superposition, shown in Figure 4.

As shown in Figure 4, there are two LSTM gates in opposite directions at each moment in the BiLSTM model. Among them, \( \tilde{h}_t \) represents the forward output of LSTM at time \( t \). \( \tilde{h}_t \) represents the reverse output of LSTM at time \( t \). \( h_t \) represents the output of BiLSTM at time \( t \). \( x_t \) represents the input at time \( t \).

The model structure of BiLSTM is shown in Figure 5, where \( V(w(i)) \) represents the word vector of the \( i \)th review text vocabulary, \( 1 \leq i \leq n \).

Supposing there is a comment text \( W = \{w(1), w(2), \ldots, w(n)\} \). Firstly, the word \( w(i) \) in the comment text \( W \) is transformed into the corresponding word vector \( V(w(i)) \) by the Word2Vec, and the sentence formed by the word \( w(i) \) is mapped to the sentence matrix \( S_y \), where \( S_y = \{V(w(1)), V(w(2)), \ldots, V(w(n))\} \), \( 1 \leq i \leq n \). Then use BiLSTM to extract the context feature of the sentence matrix \( S_y \), that is, use the forward LSTM to extract the above information feature of the review text, and the reverse LSTM to extract the below information feature of the review text.
3.2. Convolutional neural network model

Convolutional neural networks (CNN) [17] can extract local semantic features expressed by emotional words through convolutional layers. Therefore, CNN is used to extract local semantic features of text in this paper.

Supposing there is a comment text $W = \{w(1), w(2), \ldots, w(n)\}$. Firstly, the word $w(i)$ in the comment text $W$ is transformed into the corresponding word vector $V(w(i))$ by the Word2Vec, and the sentence formed by the word $w(i)$ is mapped to the sentence matrix $S_y$, where $S_y = \{V(w(1)), V(w(2)), \ldots, V(w(n))\}$, $1 \leq i \leq n$. CNN takes the matrix $S_y$ as the input of the convolutional layer. The convolutional layer uses a filter of size $r \times k$ to convolve the sentence matrix $S_y$ to extract the local semantic features of the matrix $S_y$. The calculation method is as shown as:

$$
c_y = f\left(F \times V\left(w(i:i+r-1)\right) + b\right)
$$

(8)

where $F$ represents the filter of $r \times k$, $f$ represents the non-linear conversion of ReLU, $V(w(i:i+r-1))$ represents the word vector of $r$ from $i$ to $i+r-1$ in $S_y$, $b$ represents the offset. $c_y$ represents the local semantic features of the $j$th sentence composed of $i$ words extracted by CNN.

3.3. Feature fusion model

The feature fusion model in this paper is composed of convolutional neural network and bidirectional long short memory network (BiLSTM), shown in Figure 6.

The first layer of the CNN part is the word embedding layer, which takes the sentence matrix of the word embedding layer as input, the columns of the matrix are the dimensions of the word vector, and the behaviour of the matrix is sequence-length. The second layer is the convolutional layer, which performs convolution operation to extract local features. When the word vector is 100-dimensional and the filter is $3 \times 100$, $4 \times 100$, $5 \times 100$, a better classification effect will be achieved. Therefore, convolution operation is performed with 128 filters of $3 \times 100$, $4 \times 100$, and $5 \times 100$ size, where stride size is set to 1, padding is VALID. And the local features of the sentence are obtained. In the third layer, the key features are extracted, the redundant features are discarded, and the fixed dimension feature vectors are generated. The output features of the three pooling operations are spliced together as a part of the input features of the first layer full connection layer.

In the part of BiLSTM, the first layer is the word embedding layer, which takes the sentence matrix of the embedding layer as the input, and each word vector dimension is set to 100 dimensions. The second layer and the third layer are hidden layers, and the size of hidden layer is 128. The current input is related to the previous and subsequent sequences. The input sequence is input into the model from two directions respectively. Through the hidden layer, the historical information and future information of the two directions are saved. Finally, the output parts of the two hidden layers are spliced to obtain the final output of BiLSTM.

The BiLSTM model is used to extract the contextual semantic information of words and the global features of words in the text. In this paper, the features output by CNN and BiLSTM are fused before the first Fully Connected layers (FC). The fused features are taken as the input of the first fully connected layer, and dropout mechanism is introduced between the first fully connected layer and the second fully connected layer. Some trained parameters are discarded in each iteration, so that the weight update is no longer dependent on part of the inherent features to prevent over fitting. Finally, it is input into softmax classifier to output the classification results.
4. Experiment and analysis

4.1. Experimental environment
The experimental environment of this article is as follows: the operating system is Window10, the CPU is Intel Core i5-7500, the GPU is GeForce GTX 1050Ti, the graphics card driver is NVIDIA-SMI 384.111, the memory size is DDR3 8 GB, the development environment is Pytorch, and the development tool is PyCharm.

4.2. Experimental data
This paper selects the Septwolves series of products in Yanyue.com as the research object, uses the Python request library to crawl data, and crawls consumers' online reviews of the Septwolves series of products. The number of positive and negative samples is 10050 positive samples and 10050 negative samples, 90% of which are used as training set and 10% as test set.

4.3. Hyperparameter selection
The choice of hyperparameters directly affects the final experimental results. The convolution part of the fusion model and the parameters and corresponding parameter values in the single CNN is listed in Table 1. The parameters and corresponding parameter values in the two-way long and short-term memory network part of the fusion model in this paper and the single BiLSTM is shown in Table 2. Through the fixed parameter method, we compared the 100-dimensional and 200-dimensional word vector, the sliding window size was compared with 3, 4, 5, 7, the number of sliding windows was taken as 40, 80, 128, respectively. The proportion of dropout was compared with 0.3, 0.5 and 0.6, the L2 regular term λ compared the influence of 3, 5, 7. After comparing the above parameters on the accuracy of the model, CNN achieves good classification effect when taking the parameter values in Table 1.

BiLSTM are compared with 100-dimensional and 200-dimensional word vectors. The default number of layers is set to 2. The hidden layer size is compared with 128 and 256. Finally, it is found that the model classification accuracy rate is the highest when the word vector is 100-dimensional and the hidden layer size is 128. Adam is employed as the optimization function, which designs independent adaptive learning rates for different parameters by calculating the first-order moment estimation and the second-order moment estimation of gradient, and continuously updates the network parameters to accelerate the convergence of the model.
4.4. Experimental results and analysis
In order to verify the classification performance of the feature fusion model proposed in this paper, the model is compared with single CNN model and single BiLSTM model, and the parameters of CNN and BiLSTM in feature fusion model are the same as those in single CNN model and single BiLSTM model, which are shown in Table 1 and Table 2, and the learning rate is set to 0.001.

The final results are summarized in Table 3, the classification accuracy of the proposed fusion model is 4.35% higher than that of single CNN model and 3.32% higher than that of single BiLSTM model. Two complementary models are fused to achieve better classification accuracy than single model.

| Model                  | Accuracy/% |
|------------------------|------------|
| CNN model              | 90.51      |
| BiLSTM model           | 91.54      |
| The fusion model of this paper | 94.86      |

5. Conclusion
This paper analyses the emotional tendency of cigarette consumers based on the feature fusion model of convolutional neural network and BiLSTM network. The model can not only extract the local features of the text effectively by using convolutional neural network, but also take into account the global features of the text by using BiLSTM. The mode can establish a new evaluation method of cigarette management, dynamically monitor the emotional changes of consumers.

References
[1] Li, Y.H., Qian, L., Zhou, P.F. (2017) Sentiment Analysis and Mining of Product Reviews. Information Science, 35(1): 51-55.
[2] Li, Y., Li, Q.Q., Chen, Y.H. (2018) Examining Consumer Reviews of Overseas Shopping APP with Sentiment Analysis. Data Analysis and Knowledge Discovery, 2(11): 19-27.
[3] Sun, Z.Z., Xue, L., Chen, Y.H., et al. (2012) A Review of Deep Learning Research. Application Research of Computer, 29(8): 2806-2810.
[4] JOHNSON, R., Zhang, T. (2014) Effective use of word order for text categorization with convolutional neural networks. https://arxiv.org/abs/1412.1058.
[5] Kalchbrenner, N., GREFENSTETTE, E., BLUNSOM,P. (2014) A Convolutional Neural Network for Modelling Sentences. https://arxiv.org/abs/1404.2188.
[6] KIM, Y. (2014) Convolutional Neural Networks for Sentence Classification. http://emnlp2014.org/papers/pdf/EMNLP2014181.pdf.

[7] BRAHIMI, J., Dou, D. (2014) Chain based RNN for relation classification. http://www.aclweb.org/anthology/N15/N15-1133.pdf.