Research on multi-feature fusion text classification model based on self-attention mechanism

Xiaoxia Luo, XuanHao Wang

College of Computer Science and Technology, Xi’an University of Science and Technology, Xi’an, 710600, China
592157543@qq.com, 18208208051@stu.xust.edu.cn

Abstract. For complex long texts, critical semantic information will be weakened and non-critical features will be forgotten. This paper proposes a multi-feature fusion text classification model based on autoattention mechanism. The preprocessed text data is represented by character-level vectors through the Bert model. Firstly, self attention mechanism (Self-Attention) is used to learn the dependence of text words to capture the internal structural information of the text. Secondly, according to Deep superposition convolutional neural network (DSCNN) and Bi-directional Gated Recurrent Unit (BiGRU) based on soft attention mechanism (Soft-Attention), the semantic features of text data are extracted separately, and two different feature extraction results are combined. Finally, the Softmax layer is used to classify the deep-extracted features, and the accuracy of the classification model is improved by adding a uniform distribution item to the cross-entropy loss function. The text is verified in the medical consultation data set, and the results show that the F1 value of the text classification of the model is as high as 88.91%, Text semantic understanding is better than mainstream models.

1. Introduction

Text classification is an important part of information retrieval processing. Through the characteristic representation of semantic information, the dimension of text vector sequence is reduced, the time and space order of memory sequence is reduced, so as to clarify text context information, capture key feature information, and feature fusion to determine the classification result.

In recent years, deep learning technology has been widely used to solve related work in the field of natural language processing. The convolutional neural network (CNN) model and the Recurrent Neural Networks (RNN) model are prominent in text classification[1]. Long and short-term memory network (LSTM) is an extension of the RNN model, which can reflect contextual semantic information in the mapping relationship between input and output sequences, and better understand the meaning of text sentences. Li Yang[2] et al. proposed parallel combination of CNN and BiLSTM networks to obtain text features respectively, and fusion of deep features to realize emotion classification. Wang Liya et al. [3] proposed to use BiGRU instead of BiLSTM model to simplify network structure and reduce time cost. Experimental verification showed that BiGRU network was superior to BiLSTM.

Aiming at the problem of insufficient understanding of the semantic features of complex texts, it is proposed to use self-attention mechanism to understand text structure information, and use deep superposition convolution and attention mechanism-based bidirectional gated recurrent network to
extract the semantic features of the text respectively to improve the classification accuracy of the text classification model.

2. Text feature processing and representation

2.1 Text feature processing
Feature processing operations mainly include removal of non-text parts of the data, common word segmentation, removal of stop words, feature cleaning and other feature processing processes. This paper deals with the medical consultation data set, and proposes to use regular expressions to filter out the text part, and use the PyHanLP library tokenizer to segment the text data, as the smallest unit of text information, make a stop vocabulary according to the design requirements, and remove it without affecting the text Semantic stop word information, and finally delete and clean up invalid and missing values.

2.2 Text representation
Vectorization of Chinese text data is to transform text information into a series of vectors that can express the semantic features of the text. The ELMO model proposed by Peters[5] takes into account the contextual semantic information of the text, and superimposes forward and backward semantics as word vector information through a two-layer bidirectional language model. Devlin [7] et al. proposed that the Bert model uses large-scale unlabeled corpus training to obtain rich semantic information of the text. This paper uses the Bert model to pre-train the text information and pre-train the deep two-way representation by jointly adjusting the context in all layers Text information.

3. Text classification model construction

3.1 Design ideas
According to the characteristics of the self-attention mechanism that can effectively mine the associated information of the text structure, the different feature extraction methods of the DSCNN network and the Soft-Attention-based BiGRU network are integrated. The model is mainly composed of three parts: The first part is the character-level representation layer, which vectorizes text information. The second part is the text semantic feature extraction layer, which captures the interdependent information of long text features, mainly calculates the weights between the input text word vectors, so as to obtain the long text syntactic structure features, and extract the fused context semantic features in parallel. The third part is the classifier layer, through the Softmax classification layer, the final feature is mapped to the (0,1) interval to determine the text classification category. The overall structure of the text classification model is shown in Fig.1.

![Fig.1 Text classification model diagram](image-url)
3.2 Character-level presentation layer
First of all, the text information after text feature processing is cut into characters one by one through
the tokenizer, and the mark is added, and a vectorized representation containing the semantic
information of the text is generated through the Bert model.
Suppose the text question Q={W₁, W₂, ..., Wₙ}, where n represents the length of the sentence text,
insert tags such as [CLS], [SEP] into some characters in the replacement sentence according to the
agreed rules, and use the encoder to convert the tags and text Perform encoding to generate a multi-
dimensional matrix S={X₁, X₂, ..., Xₘ}, Xᵢ is the encoding vector of a single character at the i-th position,
m is the number of characters and labels, and finally through the Bert model pre-training matrix S to
generate m-dimensional The text representation matrix Z={Y₁, Y₂, ..., Yₘ} is used as the input layer of
the classification model.

3.3 Self-Attention mechanism layer
In the understanding of text semantics, the model hopes to obtain the contextual semantic information
in the sentence vector, which can clarify the internal relevance of the sentence, so as to integrate it into
the text sentence structure information, and determine the primary and secondary relationship of the
text vocabulary.
Self-attention[10] can better understand the text structure. When the model processes the word
vector of any dimension in a sentence, it will let the vector see the vector of any other dimension in the
text, and can explore the clues between the text vocabulary, making the semantic understanding of the
text more complete.
The structure of the self-attention model refers to three matrices, Q(Query), K(Key) and V(Value),
which are used to transform text vectors. The specific expressions are as follows:

\[ Q' = W^Q \cdot W^* \cdot Y_i \]  
\[ K' = W^K \cdot W^* \cdot Y_i \]  
\[ V' = W^V \cdot W^* \cdot Y_i \]

In equations (1)-(3), Yᵢ is a character vector in matrix Z, and W, W^Q, W^K and W^V are four different
transformation weight matrices learned by the model.
Q', K', and V' are single text character vectors of \( Q, K, \) and \( V \) matrices, which are expressed as follows:

\[ Q = \begin{bmatrix} Q^1 & \cdots & Q^m \end{bmatrix} \]  
\[ K = \begin{bmatrix} K^1 & \cdots & K^m \end{bmatrix} \]  
\[ V = \begin{bmatrix} V^1 & \cdots & V^m \end{bmatrix} \]

In equations (4)-(6), m represents the dimension of the matrix Z.
The text relevance of self attention model is mainly to calculate the similarity between two vectors
by dot multiplication similarity attention formula. In order to reduce the influence of data dimension, it
is necessary to add standard normal distribution homogenization operation, which is represented as
follows.

In equation (7), d_k is the dimension value of input vector, softmax is normalization operation, and
B matrix is feature output of self attention mechanism layer, which is called text self attention feature
matrix in this paper.

\[ B = \text{softmax} \left( \frac{QK^T}{\sqrt{d_k}} \right) V \]  

3.4 DSCNN layer
The traditional CNN model has a relatively superior effect in text classification tasks, but it cannot pay
attention to semantic information clearly when facing longer text extraction features. Therefore, it is
necessary to stack multiple convolutional layers and adjust the size of the convolution window to
extract and merge local semantic features at different positions to improve the understanding of text semantics. Therefore, it is proposed to use the DSCNN model to extract text attention features. By setting up a multi-layer convolution feature extraction layer, the multi-layer convolution and the initial convolution are superimposed and summed to strengthen key local features and cooperate with the global maximum pooling layer (GlobalMaxPooling), the global maximum value feature of the text vector is pooled and extracted, and the deep superimposed convolution feature extraction is realized. The structure diagram is shown in Figure 2.

![Fig.2 DSCNN Model structure diagram](image)

By taking the feature matrix of the self-attention mechanism as the input of the deep overlay convolutional layer, using the multi-layer convolutional layer to extract the features multiple times, superimposing the first extracted text features with the final text features, and finally passing through the global maximum pooling layer, the output depth overlays the text information features extracted by the convolutional layer.

### 3.5 BiGRU layer
GRU[11] uses two gating units, an update gate $z_t$ and a reset gate $r_t$, to reset and update memory. The reset gate $r_t$ is used to control the previous hidden layer unit $h_{t-1}$ to semantically superimpose the current input data $x_t$ and write information into the current candidate set $\tilde{h}_t$; the update gate $z_t$ is used to judge the importance of the current input data $x_t$ to the overall semantics. By superimposing the candidate set $\tilde{h}_t$ on $h_{t-1}$, the comprehensive semantic $h_t$ at the current moment is obtained, which is expressed as follows:

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t])$$  \hspace{1cm} (8)

$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t])$$ \hspace{1cm} (9)

$$\tilde{h}_t = \tanh(W_x \cdot [r_t \ast h_{t-1}, x_t])$$ \hspace{1cm} (10)

$$h_t = (1 - z_t) \ast h_{t-1} + z_t \ast \tilde{h}_t$$ \hspace{1cm} (11)

In equations (8)-(11), $\sigma$ is the activation function of Sigmoid, $[h_{t-1}, x_t]$ reference the connection of two vectors, and $W_r, W_z$ and $W_x$ are the model learning weight matrix.

BiGRU model can better capture the bidirectional semantic dependency by adding reverse operation on the basis of GRU. The overall structure can be divided into forward GRU and backward GRU. By stitching the bidirectional GRU features together at the same location in the hidden state, the final text representation can be understood in context.

### 3.6 Soft-Attention layer
Soft-Attention mainly calculates the relevance weight of data elements in the global through the understanding and analysis of global data. The larger the weight coefficient, the higher the importance of the information, and the more focused the current feature element.

---

4
The Soft-Attention algorithm first calculates the correlation score between data sequences according to the MLP alignment function, and then normalizes the correlation by softmax function. The specific calculation process is as follows.

\[
\text{score} = v^T \tanh(W[u; s] + b) \tag{12}
\]

\[
\alpha = \frac{\exp(\text{score})}{\sum \exp(\text{score})} \tag{13}
\]

\[
c_i = \sum \alpha_s u \tag{14}
\]

In equations (12)-(14), \(v^T\) and \(W\) are the weight matrix, \(B\) is the bias vector, \([u; s]\) represents the joint of the feature matrix \(u\) at the current moment and the feature matrix \(s\) recorded before the current moment, \(\alpha\) is the normalized weight matrix, and \(c_i\) represents the importance of the global feature.

3.7 Concatenation layer

The main task of the connection layer is to combine the DSCNN layer and BiGRU-Attention layer in the feature extraction layer to extract semantic features in parallel, so as to realize the semantic understanding method of text information from multiple perspectives.

The text vector passes through the DSCNN layer to obtain the feature vector \(X_C\); passes through the BiGRU-Attention layer to obtain the feature vector \(X_S\). Then, \(X_C\) and \(X_S\) are connected to construct the semantic feature \(X_{CS}\). The specific calculation process is as follows:

\[
X_{CS} = X_C \oplus X_S \tag{15}
\]

In equation (15), \(\oplus\) is the connector between vectors.

3.8 Softmax classification layer

This model uses the Softmax classifier to classify the obtained text features. The specific calculation formula is as follows:

\[
y = \text{softmax}(X_{CS}) \tag{16}
\]

\[
\tilde{y} = \text{argmax}(y) \tag{17}
\]

In equations (16)-(17), \(X_{CS}\) is the fusion feature output of the splicing layer, \(y\) is the output vector of feature normalization, and the data range of each dimension is between \([0,1]\). Dimension is the number of classification categories, \(\tilde{y}\) is the vector maximum index obtained by the argmax function, representing the final text category.

In the model training process, the classification task uses the cross-entropy loss function to calculate the error between the prediction and the true value to achieve better results. The cross-entropy loss function is:

\[
S(q|p) = -\sum_i q_i \log p_i \tag{18}
\]

In equation (18), \(p_i\) is the predicted value, \(q_i\) is the true value.

Softmax classifiers had problems with over-confidence in classification, which often led to the risk of overfitting. In this paper, uniform distribution is added to the cross-entropy loss function to improve classification accuracy, and the final loss function is as follows:

\[
J(\theta) = -(1-\eta)(-\sum_i q_i \log p_i) - \eta(-\sum_i \frac{1}{n} \log p_i) \tag{19}
\]

In equation (19), \(\eta\) is a custom weight parameter, \(n\) is the number of classification categories.
4. Experimental results and analysis

4.1 Experimental data
This experiment tested the performance of the text classification model. Using the real Q & A data of medical search and drug website, there were 32636 text question data, which were divided into 0~6 categories, corresponding to infectious, pediatrics, obstetrics and Gynecology, internal medicine, dermatology, surgery and facial features. Question text information is more complex in structure, the language description is too colloquial, and it is difficult to analyze the semantics of the text, and theoretical inference is needed to determine the true category of the question text. Verifying the effectiveness of the model and the feasibility of the algorithm on this data set can better reflect the excellent characteristics of the model.

4.2 Analysis of experimental results
In order to verify the feasibility and effectiveness of the model, the results are compared with the current mainstream models. The comparative experimental models include: CNN, LSTM, GRU, BiGRU, CNN and BiGRU parallel network, CNN and BiGRU parallel network, CNN and BiGRU series network based on Attention, DSCNN, DSCNN and BiGRU parallel network based on Attention, DSCNN based on Self-Attention and BiGRU series network based on attention. The experimental evaluation results are shown in Table 1.

| label | model                      | precision | recall | F1    |
|-------|----------------------------|-----------|--------|-------|
| 1     | CNN                        | 0.779687  | 0.769111 | 0.773638 |
| 2     | LSTM                       | 0.837203  | 0.841241 | 0.839032 |
| 3     | GRU                        | 0.841103  | 0.838236 | 0.841835 |
| 4     | BiGRU                      | 0.847298  | 0.843289 | 0.842092 |
| 5     | CNN+BiGRU                  | 0.855359  | 0.850461 | 0.852037 |
| 6     | CNN+Attention_BiGRU        | 0.850559  | 0.851051 | 0.847653 |
| 7     | CNN Attention_BiGRU        | 0.853216  | 0.843936 | 0.848576 |
| 8     | DSCNN                      | 0.867807  | 0.848104 | 0.857955 |
| 9     | DSCNN+Attention_BiGRU      | 0.850562  | 0.851438 | 0.850553 |
| 10    | Paper model                | 0.88316   | 0.890151 | 0.889132 |

The comparison results of the first three models show that BiGRU text classification model achieves better classification effect than CNN, LSTM and Gru models. The CNN+BiGRU model adds a CNN network to the BiGRU model, and the F1 value is slightly improved. The comparative experiments in label 6 and 7 show that the CNN network and the Attention-based BiGRU network parallel network have more accurate classification results. The DSCNN model is better than the previous network. The comparison experiment of 9 and 10 in the table shows that the text classification model with self-attention mechanism has made a big improvement. The model in this paper uses a uniformly distributed cross-entropy loss function on the Self-Attention DSCNN+Attention_BiGRU model, and the F1 value has reached The high value of 88.91%, it can be seen that the classification results of this model are more accurate and the classification accuracy is higher.

Taking into account the over-fitting of the model, the text adds a uniform distribution term to the cross-entropy loss function to prevent over-fitting. It can be seen from Figure 3 and Figure 4 that although the loss value of the uniformly distributed cross-entropy loss function model is increased, it is still within the normal range. The uniformly distributed loss function can obviously converge faster and weakly improve the model classification Accuracy.
5. Conclusion
This article focuses on the study of the classification model of complex text sentences. In order to make the model face the text sentences that need to be understood, it adds a self-attention mechanism to analyze the structure of text between words, uses convolutional networks to extract semantic features of the text, and passes bidirectional gated loops. The neural network understands context semantics, fuses multiple features into final text semantic features, and classifies sentences according to the classifier. Through comparative experiments, it is found that the self-attention mechanism effectively improves the classification accuracy, the deep superposition convolution can obtain more semantic features, and the uniform distribution of the cross-entropy loss function effectively prevents over-fitting. The next step will be to consider the use of a multi-view text classification model facing a single text multilingual data set to improve the generalization ability of the model.

References
[1] ZHAO Yunshan, DUAN Youxiang.Convolutional Neural Networks Text Classification Model Based on Attention Mechanism[J].Journal of Applied Sciences,2019, 37(004):541-550.
[2] Li Yang, DONG Hongbin.Text sentiment analysis based on feature fusion of convolution neural network and bidirectional long short-term memory network[J].Journal of Computer Applications,2018,38(11):3075-3080.
[3] WANG Li-ya, LIU Chang-hui, CAI Dun-bo.Chinese text sentiment analysis based on joint network and attention model.Microelectronics & Computer,2020,37(1):80-86
[4] Mikolov T, Chen K, Corrado G, et al. Efficient Estimation of Word Representations in Vector Space[J]. Computer ence, 2013.
[5] Pennington J, Socher R, Manning C. Glove: Global Vectors for Word Representation[C]// Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2014.
[6] Peters M E, Neumann M, Iyyer M, et al. Deep contextualized word representations[J]. 2018.
[7] Devlin J, Chang M W, Lee K, et al. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding[J]. 2018.
[8] Vaswani A, Shazeer N, Parmar N, et al. Attention Is All You Need[J]. arXiv: Computation and Language, 2017.
[9] Cho K, Van Merrienboer B, Gulcehre C, et al. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation[J]. arXiv: Computation and Language, 2014.