Research on Chinese Text Semantic Matching Base on BERT and BiLSTM_Attention

Changqun Li¹, Zhongmin Pei¹*, Li Li², Zhangkai Luo¹ and Da Peng¹

¹ Science and Technology on Complex Electronic System Simulation Laboratory, Space Engineering University, Beijing, China
² Gaojing Network Management and Security Center, Beijing, China

*Corresponding author. Email: licqlcq@163.com

Abstract. In this paper, to further improve the matching accuracy of text semantic matching, which is based on BERT (Bidirectional Encoder Representations from Transformers) fine-tuning, we propose a novel model, which is combine BERT and Attention-Based Bidirectional Long Short-Term Memory Networks (BERT+BiLSTM_Attention). Indeed, outputs of the BERT are further given as inputs of the Bidirectional Long Short-Term Memory Networks (BiLSTM), after that the attention mechanism is further used to capture the needed interactive information and attend those informative words that have a significant impact on semantics. In order to optimize the model, the piecewise constant decay strategy is adopted to control the learning rate. In addition, the weights of BERT are updated to make our model more suitable to the downstream tasks. Finally, experimental results demonstrate that the accuracy of the proposed model is better than LSTM-DSSM model 7.1% in a large-scale Chinese question matching corpus. And 0.12 points higher than BERT based direct fine-tuning.

Keywords: semantic matching, BERT, BiLSTM_Attention.

1. Introduction
Text semantic matching, which matches a target text to a source text and estimates the semantic similarity between them, is one of the most important research problems in many domains, such as question answering [19], information retrieval [20], and recommendation [21].

The earlier attempts to text semantic matching is to compute a vector as the representation for each text piece, such as bag-of-words and latent Dirichlet allocation models [1], and then apply typical similarity metrics to compute the matching scores. Unfortunately, the performance of these traditional approaches is unsatisfactory, as they often fail to identify semantically similar text pairs without an exact expression match. In recent years, the advances in deep learning provide the opportunity to understand complex natural language, such as, recurrent neural networks [2], long short-term memory neural networks [3], and convolutional neural networks [4], which have been firmly established as state-of-the-art approaches to understanding the complex semantic correlations within and between texts. By now, these existing deep neural approaches for text semantic matching can be categorized into two classes: the representation-focused model and the interaction-focused model. Several studies have demonstrated the latter one is more reasonable and promising for text semantic matching [5][6][7]. However, a large number of parameters, the labeled training data and time is required to help the interaction-focused model better complete the training. In order to alleviate the problem of limited
labeled training data and to carry out model training faster, pre-training language model, such as, Embedding from Language Models (ELMO) \cite{8}, Generative Pre-trained Transformer (GPT) \cite{9}, and Googles BERT model \cite{10} is widely used.

Among the context-sensitive language models, BERT has taken the NLP world by storm. Base on the transfer learning\cite{11}, the model of semantic matching is initialized with the pre-trained parameters to provide embedding or encoding that contain semantics of the original sentence, and then is fine-tuned using the labelled datasets. Our model could further capture matching structural information of source and target objects by adopting BiLSTM and attention mechanism. The learning rate adopts the piecewise constant decay strategy, and sets different learning rate constant values in the training times interval, which helps to finely adjust model parameters. Finally, we choose updating pre-trained BERT parameters. Experiment shows model performance is further improved.

2. BERT+BiLSTM_Attention (BBA) model architecture

The model is initialized with a pre-trained language model BERT to get text semantics instead of Word2vec \cite{17} or Glove \cite{18}, the fixed length vector is used as the input of BiLSTM, and using the attention mechanism to capture interactive information, the attention layer is followed by two dense layers with different sizes of neurons. As shown in Figure 1, the model proposed in this paper contains five components:

(1) Input layer: input sentence to this model;
(2) BERT layer: initialize with BERT that encodes sentences of arbitrary length into fixed-length vectors;
(3) BiLSTM layer: utilize one-layer BiLSTM to get high level features from step (2);
(4) Attention layer: produce a weight vector, and merge word-level features from each time step into a sentence-level feature vector, by multiplying the weight vector;
(5) Output layer: the sentence-level feature vector is finally used for semantic matching.

BERT is a language representation model developed by Google in 2018 to pre-trained deep bidirectional representations by jointly conditioning on both left and right context in all layers. This is done by using masked language model and next sentence prediction two novel unsupervised prediction tasks. The advantage of BERT is that it can provide embedding or encoding that contain semantics of the original sentence for downstream tasks.

LSTM \cite{3} units are firstly proposed by Hochreiter and Schmidhuber to overcome gradient vanishing problem. The main idea is to introduce an adaptive gating mechanism, which decides the degree to
which LSTM units keep the previous state and memorize the extracted features of the current data input. As shown in the following equation:

\[ f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \]  
\[ i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \]  
\[ C_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \]  
\[ C_t = f_t \ast C_{t-1} + i_t \ast \tilde{C}_t \]  
\[ O_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \]  
\[ h_t = O_t \ast \tanh(C_t) \]

Experiment shows that LSTM be stable and powerful for modeling long-time dependencies in various scenarios such as speech recognition and machine translations, etc. The LSTM model can capture long-distance dependencies and process sequences in temporal order, they ignore future context. But for many sequence modelling tasks, it is beneficial to have access to future as well as past context. Bidirectional LSTM [12] or BiLSTM is an extension of traditional LSTM to train two LSTMs on the input sequence. The second LSTM is a reversed copy of the first one, BiLSTM model is therefore able to exploit information both from the past and the future. So BiLSTM can better capture the most important semantic information in a sentence.

Attentive neural networks have recently demonstrated success in a wide range of tasks ranging from question answering [13], machine translations [14], and speech recognition [15]. In general, semantics usually vary according to different parts of the sentence. For discriminating the semantic similarity of two sentences, not all words have the same effect. Based on that, the attention mechanism is introduced to attend those informative words that have a significant impact on semantics and aggregate their representations to form a sentence vector. In effect, the attention mechanism is to compute a context vector in a sentence.

The attention layer is followed by two dense layers with different sizes of neurons, the paper regards the task as a two-classification problem at the time of output and use cross-entropy loss to optimize.

3. Algorithm implementation

The paper uses the Google pre-trained Chinese Simplified version of the BERT model to get fixed length word embedding of the sentence as the input of the next layer. According to the test results of the BERT paper, the paper concatenates the last four layers of the BERT used as the input of BiLSTM.

\[ e = [e_{-1}; e_{-2}; e_{-3}; e_{-4}] \]

After the BERT layer, the sequence of word vectors is fed into BiLSTM to achieve another representation. Since we concatenate the last four layers of the BERT, the number of neurons in the BiLSTM layer is configured to be twice the BERT dimension. The network contains two sub-networks for the left and right sequence context, which are forward and backward pass respectively. The output of the \( j^{th} \) word is shown in the following equation:

\[ h_j = \left[ \overrightarrow{h_j}; \overleftarrow{h_j} \right] \]

Here, we concatenate the forward and backward pass outputs.

Then the paper uses the attention mechanism to optimize. The BiLSTM network will produce a hidden \( h_j \) state at each time step. To begin with, the vector of \( h_j \) and the hidden state \( s_{t-1} \) of BiLSTM at time \( t-1 \) is fed into a one-layer MLP to learn a hidden representation \( e_{ij} \), then the weight coefficient \( a_{ij} \) of each feature vector \( h_j \) is obtained through a softmax function. At last, the output vector \( c_i \) is weighted by the feature vectors\([h_1, h_2, ..., h_T]\). The formulas can be described as follows:

\[ e_{ij} = \tanh(s_{t-1}, h_j) \]

\[ a_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{T} \exp(e_{ik})} \]

\[ c_i = \sum_{j=1}^{T} a_{ij} h_j \]
where $T$ is the sentence length.

Finally, for the task of semantic matching, regard the task as a two-classification problem at the time of output. That is, the two sentences are similarly labeled as 1, otherwise 0. The above output vector $c$ is used as input:

$$\hat{y} = W_c c + b_c #(12)$$

The cost function is the negative log-likelihood of the true class labels $y$:

$$J(\theta) = -\frac{1}{m} \sum_{i=1}^{m} t_i \log(y_i) #(13)$$

where $t \in \mathbb{R}^m$ is the one-hot represented ground truth and $y \in \mathbb{R}^m$ is the estimated probability for each class by softmax and $m$ is the number of target classes.

To prevent model overfitting, the paper uses the dropout [16] strategy, by randomly omitting feature detectors from the network during forward propagation and achieve regularization of neural networks. Concretely, the paper employ dropout on the LSTM layer and the attention layer and dropout rates is 0.5.

4. Experiment

4.1. Datasets and evaluation metric

The paper conducts experiment in a large-scale Chinese question matching corpus, and uses multiple semantic matching methods to conduct comparative experiments to verify the effectiveness of the BBA model. The effects of different learning rates and loss functions on the performance of the model are analyzed.

Using Chinese question matching corpus published by Ant Financial on the competition platform, this public corpus contains 100,000 samples, the ratio of positive and negative samples is about 1:4.5, and the proportion of positive samples is lower.

Firstly, the paper train model on the training set, then tune the hyper parameters of model on the validation set, and finally model evaluation method is online evaluation.

The paper adopts the accuracy evaluation metric to evaluate model performance. The definition is as follow:

$$\text{Accuracy} = \frac{T}{N} #(14)$$

where $T$ is the correct number of predicted samples, $N$ is the total number of tested samples. And the final score is calculated as follow:

$$\text{Score} = 100 \times \text{Accuracy} #(15)$$

4.2. Experiment results

For training, the batch size is set at 32 and use the pre-trained BERT model, the max length of sentences is configured as 128 to ensure the same maximum length as the BERT model. In this model, the categorical cross-entropy based loss function and the gradient descent algorithm with Adaptive Moment Estimation are used to learn the model parameters of neural networks. The learning rate adopts the piecewise constant decay strategy. The final stage parameters of Adaptive Moment Estimation are learning_rate=2e-06, beta_1=0.9, beta_2=0.999, epsilon=1e-08. The experiment results are shown in Table 1.

<table>
<thead>
<tr>
<th>Model</th>
<th>BERT train or not</th>
<th>BERT output</th>
<th>Accuracy(Score)</th>
</tr>
</thead>
<tbody>
<tr>
<td>LSTM-DSSM</td>
<td>-</td>
<td>-</td>
<td>79.80</td>
</tr>
<tr>
<td>BERT(Fine tuning)</td>
<td>yes</td>
<td>-</td>
<td>85.55</td>
</tr>
<tr>
<td>BERT+BiLSTM_Attention(BBA)</td>
<td>no</td>
<td>Concat Last Four Hidden</td>
<td>85.19</td>
</tr>
<tr>
<td>BERT+BiLSTM_Attention(BBA)</td>
<td>yes</td>
<td>Concat Last Four Hidden</td>
<td><strong>85.67</strong></td>
</tr>
</tbody>
</table>
The experimental results show that the performance of text semantic matching model based on BERT is much higher than that of LSTM-DSSM model. The BBA model proposed in the paper has carried out different experiments. The difference is whether the BERT is trained or not. It was found that performance is best when BERT and BiLSTM_Attention are trained simultaneously. And 0.12 points higher than direct fine tuning base on BERT.

The paper compares the effects of LSTM, BiLSTM and BiLSTM_Attention on the performance of the model. The result is shown in Figure 2.

![Experimental result](image)

**Figure 2.** Experiment results of different models.

Experiment results show that the performance based on the BiLSTM_Attention model is better. The model pay attention to word alignment information and semantic correlation information between two sentences through the attention mechanism, which can obtain richer semantic features. The paper also compares the standard LSTM neural networks and bidirectional LSTM networks without using the attention mechanism. More specifically, the BERT+BiLSTM model performs better than BERT+LSTM model when bidirectional semantic information is taken into consideration. From the results of the BERT+LSTM and BERT+BiLSTM models, it is possible that the bidirectional LSTM model can get more semantic features.

The experiment compares and analyzes the effects of different learning rate strategies on the performance of the model, namely the piecewise constant decay strategy and the exponential decay strategy, in which the exponential decay strategy uses different decay_rate and decay_steps. As shown in Figure 3 and Figure 4.

![Exponential decay](image)

**Figure 3.** Exponent decay strategy.
The experiment results are shown in Table 2.

Table 2. Experiment results of different learning rate strategy.

<table>
<thead>
<tr>
<th>Learning Rate Strategy</th>
<th>BERT</th>
<th>BERT+LSTM</th>
<th>BERT+BiLSTM</th>
<th>BBA</th>
</tr>
</thead>
<tbody>
<tr>
<td>piecewise constant</td>
<td>85.55</td>
<td>84.96</td>
<td>85.37</td>
<td>85.67</td>
</tr>
<tr>
<td>exponent decay, decay_rate=0.9</td>
<td>83.5</td>
<td>81.99</td>
<td>83.99</td>
<td>85.16</td>
</tr>
<tr>
<td>exponent decay, decay_rate=0.92</td>
<td>83.5</td>
<td>81.99</td>
<td>83.99</td>
<td>85.16</td>
</tr>
</tbody>
</table>

The experiment results are better with the piecewise constant decay strategy. By setting different learning rate constant values in the training time interval, the task can be finely adjusted, which helps to improve the performance of the model.

Finally, the experiment compares and analyzes the effects of different loss functions on the performance of the model, using cross-entropy loss and MSE loss, respectively. The experiment results are shown in Figure 5.

Figure 5. Experiment results of different loss functions.

From the experimental results, the choice of the loss function has a great impact on the performance of the model performance. The cross-entropy loss is more suitable for the task of classification, and the error is punished in an appropriate way when adjusting the parameters.
5. Conclusion
This paper uses the BBA model to solve the semantic matching problem and treat it as a classification task. And using some optimization tricks to improve model performance. Proving the validity of the model by testing it on Chinese question matching corpus.

As for future works, due to the Transformer, based solely on attention mechanisms, can be more parallelizable and requiring significantly less time to train, try to use transformer instead of LSTM for better performance.

Acknowledgement
This research was financially supported by the Key Laboratory Fund Project under project numbers DXZT-JC-ZZ-2016-001.

References

