Sentence Similarity Learning Method based on Attention Hybrid Model

Yue Wang¹,², Xiaoqiang Di¹,², Jinqing Li¹,², Huamin Yang¹,² and Lin Bi¹,²

¹ School of Computer Science and Technology, Changchun University of Science and Technology, Changchun, 130022, China
² Jilin Province Key Laboratory of Network and Information Security, Changchun, 130022, China
Email: lijinqing@126.com

Abstract. Sentence similarity learning is a vital task in Natural Language Processing (NLP) such as document summarization and question answering. In this paper, we propose a method to compute semantic similarity between sentences which is based on the attention hybrid model. Our method utilizes Bidirectional Long Short-Term Memory Networks (BLSTM) and Convolutional Neural Networks (CNN) to extract the semantic features of a sentence. And it learns the representation of each sentence with word-level attention. Then the attentive representations are concatenated and fed into the output layer to compute the score of sentences similarity. Finally, the public datasets of the Quora is used to test the proposed method and experiment results show that our method is effective and outperforms other methods.

1. Introduction
Sentence similarity learning is important for different tasks, such as automatic text summarization [1], information retrieval [2, 3], text categorization [4, 5, 6], and machine translation [7, 8]. Our research is oriented towards finding an efficient approach to learn and compute similarity between sentences.

Deep Neural Networks (DNN) has been used in measuring semantic similarity between sentences [6, 7]. Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN) are two architectures of DNN [9]. They are widely used in Nature Language Processing (NLP) tasks. CNN is a hierarchical architecture and RNN is a sequential architecture. CNN can extract local features, but it can’t handle the NLP tasks that are represented by the sequences whose length are usually unknown in advance [5]. Nevertheless, RNN can handle sequential problems by propagating historical information. One example is the Bidirectional Long Short-Term Memory Networks (BLSTM) model [10]. Afterwards, by combining the advantages of RNN and CNN, the hybrid two structures based on them have been proposed in the latest work.

In our work, we propose a method that an attention-based hybrid model is used to learn the semantic similarity of sentence pairs. Specifically, we utilize BLSTM and CNN to learn the representation of each sentence with word-level attention. The attentive representations compute sentence similarity score. Experimental results indicate that our proposed approach is effective and outperforms several strong methods, including CNN, BLSTM, Attention-BLSTM, and SVM.

The main contribution of our work can be summarized as follows: (1) we explore the attention-based hybrid model to measure sentence similarity; (2) we apply an attention mechanism, which can enhance the mutual relation between the aspect term and its corresponding sentences and prevent the irrelevant words from getting more attention; (3) we carry out the experiment on the Quora dataset by utilizing multiple methods.
The rest of this paper is organized as follows: Section 2 briefly introduces the work related to this study. Section 3 describes the model in detail. Experimental results are reported in Section 4. Section 5 concludes this paper.

2. Related Work
Sentence Similarity learning is an essential problem in Natural Language Processing (NLP) tasks. For example, given a pair of sentences, a measuring function is required to determine the matching degree between two sentences. The sentence $S_a$ and $S_b$ are represented by vectors $V_a$ and $V_b$ respectively, and then the similarity layer $\text{Sim}(V_a, V_b)$ calculates the score. When the score is less than the threshold $\theta$, it should be dissimilar and vice versa [11]. For measuring sentence similarity, there are various existing approaches such as lexical matching, linguistic analysis, and semantic features [12, 13].

More recently, researchers have begun to incorporate attention mechanisms into semantic features. The attention mechanism was firstly proposed by Bahdanau [8] for machine translation, in which the decoder changes the processed way of source sentences. Since then, attention mechanism has been widely used, Zhang [5] expressed sentences used by an attention pooling-based CNN, and the attention weight is obtained by intermediate sentence representation which is generated through BLSTM. Yujun [14, 15] used a Hybrid Attention Networks (HAN) model to obtain the semantic features between sentences by applying BLSTM and CNN.

The model used in our method is most closely related to the HAN model. We used the attention-based hybrid model that combines BLSTM and CNN to learn the representations of each sentence. Attention-based hybrid model was used to address the problem of text classification [15]. So, we adopt this model to calculate the sentence similarity.

3. An Attention-Based Hybrid Model
As shown in Figure 1, we propose an attention-based hybrid model combining BLSTM and CNN which contains four components:

1. Input layer: input two diverse sentences to this model;
2. Embedding layer: map each word of a sentence into a low-dimension vector;
3. Hybrid attention layer: produce a weight vector, and concatenate word-level features into a sentence feature vector by multiplying the weight vector;
4. Output layer: calculate the sentence similarity score by concatenating sentence feature vector.

Later in this section, these components will be presented in detail.

![Figure 1. The architecture of the Attention-based Hybrid model](image-url)
3.1. Embeddings
Given a sentence composed of N words \( S = \{w_1, w_2, \ldots, w_n\} \), every word \( w_i \) is converted into an embedding vector \( e_i \). A word \( w_i \) is transformed into its word embedding \( e_i \) by using the matrix-vector product:

\[
e_i = W^w v^i
\]  

(1)

The embedding matrix \( W^w \) is the parameter to be learned, \( W^w \in \mathbb{R}^{d^w \times |V|} \), where \( V \) is a fixed-sized vocabulary, and \( d^w \) is the size of word embeddings. It is a hyper-parameter to be set by user. The word vector \( v^i \) is a vector of size \( |V| \) which has value 1 at index \( e_i \) and 0 at all other positions. Then a sentence is fed into the next layer as a vector.

The goal of embedding layer is to represent each word in sentences with a \( d \)-dimensional vector. The whole embedding space is used, in which the embedding is updated after each batch.

3.2. Hybrid Attention
The motivation of attention is inspired by the observation that different words should have different contributions to the final semantic representation of a sentence. When reading a sentence, people often pay attention to one word or several words, and these words can reflect the meaning of the sentence. So, we use attention mechanism focused on word-level to implement this motivation.

We only pay attention to the words whose semantic relationship have a great impact on sentence meaning through word-level attention mechanisms. BLSTM and CNN can extract the feature representations in word-level attention architecture. In the attention layer, the output is the concatenated representations [15].

The BLSTM produces the output vectors \( [h_1, h_2, \ldots, h_n] \). As Eq.(2) shows, we can use an attention-weighted sum of output vectors to generate the representation \( S_\alpha \) of a sentence. The attention-weight \( \alpha_i \) is shown in Eq.(4), where \( W_\alpha \) is a word of weight. And Eq.(3) represents the output of hidden layer.

\[
S_\alpha = \sum_{i=1}^{n} \alpha_i h_i
\]  

(2)

\[
u_i = \tanh(W_c C_i + b_c)
\]  

\[
\beta_i = \text{softmax}(W_p v_i)
\]  

(3)

\[
\alpha_i = \text{softmax}(W_\alpha u_i)
\]  

(4)

The attention mechanism gets the representation \( S_\alpha \) for the output of forward and backward LSTM from the formulas described above. The BLSTM network proposed in [9, 16] can be utilized. Past features (via forward states) and future features (via backward states) for a specific time can be used. Except for the need to unfold the hidden states efficiently, the forward and backward passes over the unfolded network through time are performed in a similar way to the forward and backward passes of conventional networks.

The convolutional layer output vectors are \( [C_1, C_2, \ldots, C_n] \). Each element of the vector \( v_i \) is calculated by a tanh function using each convolution feature \( C_i \) in the hidden layer. And the attention weight \( \beta_i \) decides the information of convolution features by a softmax function. Afterwards, the pooling vector \( \sum_{i=1}^{n} \beta_i C_i \) is computed by a weighted sum of the convolutional layer output. We can compute attentive representation whose output vectors is \( C_\beta \) as follows:

\[
v_i = \tanh(W_c C_i + b_c)
\]  

\[
\beta_i = \text{softmax}(W_p v_i)
\]  

(5)

\[
C_\beta = \sum_{i=1}^{n} \beta_i C_i
\]  

(6)

Multiple local representations can be learned by using CNN. The attentive representations are produced by various CNNs through a max function and are then fed into the model to obtain the final pooling feature vector. And Eq.(8) shows that the representation \( S_\beta \) of a sentence, where \( k \) is the length of the convolution window.

\[
S_\beta = \arg\max(C_\beta k)
\]  

(7)
3.3. Similarity Function

In our work, we use cosine similarity function to compute similarity value of two sentences. As shown in Eq.(9), given two vectors $s_1$ and $s_2$ as input, the function will output the similarity score $S(s_1, s_2)$. We employ a linear function to sum up all the feature vectors and apply a sigmoid function to constrain the similarity within the range $[0, 1]$.

$$S(s_1, s_2) = \frac{s_1^T s_2}{||s_1|| ||s_2||}$$  \hspace{1cm} (9)

4. Experiments

4.1. Datasets

The Quora duplicate questions public dataset contains 404k pairs of questions from the Quora website. Each sentence pair of the three datasets is marked as 1 if the two sentences are for the same meaning, and otherwise 0. In experiments, we used the tokenizer in the NLTK Python package to pre-process sentences and exclude sentence pairs with non-ASCII characters.

4.2. Baselines

We select four machine learning baselines including Support Vector Machine (SVM), Gradient Boosting Decision Tree (GBDT), Logistic Regression (LR), and Random Forest (RF), and select three neural network baselines including CNN, BLSTM, and Attention-BLSTM [17] to compare with our method.

4.3. Experimental Setup

We initialize word representation in the word embedding layer with the 300-dimensional GloVe word vectors pre-trained from the Common Crawl Corpus [18]. Embeddings for words that are not presented in the GloVe model are randomly initialized with the uniform distribution $[0.1, 0.1]$. The ADAM optimizer is chosen to update parameters to perform optimization [19]. The back propagation algorithm is used to compute gradients for all parameters during training. We fix the word embeddings to 100 dimensions, learning rate to 0.01, dropout rate to 0.3, kernel size to 3, batch size to 64 and epoch to 100. During the process of training, we do not update the pre-trained word embeddings. For all the experiments, we choose the model which works best on the train set, and then evaluate it on the validation set.

4.4. Results and Analysis

Sentences are fed into the models, and their hyper-parameters are the same. We used four metrics: Accuracy, Precision, Recall and F1 to evaluate the performance of the proposed models. By comparing these four indicators of other methods with our method, it is obvious that our method achieves the best performance. This is because we adopt word-level hybrid attention mechanism to improve the weight of meaningful words in a sentence to comprehensively consider the local information and summary information.

From Table 1, we use models to learn different classifiers in the Quora dataset. We can see that the first kind of models s based on traditional machine learning. Random Forest performs better than the other three models. The second kind of models is based on deep learning methods which learn the vector representations for sentences. And our method is more accurate than other methods in different datasets.
Table 1. Comparison with previous results in Quora.

| Method     | Accuracy | Precision | Recall  | F1    |
|------------|----------|-----------|---------|-------|
| SVM        | 0.7532   | 0.5623    | 0.5172  | 0.6089|
| GDBT       | 0.7371   | 0.5374    | 0.5063  | 0.5877|
| LR         | 0.7444   | 0.5467    | 0.5291  | 0.6044|
| RF         | 0.7725   | 0.5853    | 0.5449  | 0.6374|
| CNN        | 0.8182   | 0.8183    | 0.8183  | 0.8182|
| BLSTM      | 0.7553   | 0.7553    | 0.7553  | 0.7553|
| Att-BLSTM  | 0.8222   | 0.8222    | 0.8222  | 0.8222|
| This work  | 0.8977   | 0.8977    | 0.8976  | 0.8973|

Figure 2 also shows the results about accuracy and loss of test set and verification set for 100 epoch times in the attention-based hybrid model by combining BLSTM and CNN on the Quora dataset. As is illustrated in graph (a), both validation set accuracy and accuracy show an ascent tendency along with the increase of epochs and the results become stable after 80 epochs, while our accuracy is a little higher. For line graph (b), both validation set accuracy and loss function show the decrease tendency along with epochs. However, the former fluctuates sharply in the first 50 epochs while the latter declined during the whole training time and finally becomes a bit lower. Therefore, according to the analysis above, we can find that our method achieves the best results in terms of prediction accuracy and obtains the lowest loss in the dataset.

4.5. Visualization of Attention

In order to validate that our method is able to select the important words of a sentence, we visualize the hybrid attention layer for several pairs of sentences from the Quora dataset.

The matrix of Figure 3 represents each word weight for a pair of sentences. The darker colour reveals higher attention weight value while the lighter part has little importance. In Figure 3, "How", "Why", "friends" and "China" have high attention weights for several words. From this, it can be seen that important words are considered in the attention process. We can observe that the model successfully identifies the important degrees of the sentence.
5. Conclusion
In this work, we introduce a method that uses attention-based hybrid model for measuring sentence similarity between two sentences. Firstly, we use BLSTM and CNN to extract semantic features from the sentences. Then we explore the attention-based neural network model to pay attention on the similarity of one sentence. At last, we test our approach on the Quora dataset, which achieves a higher accuracy and precision. The experiment result reveals that our method is efficient and performs significantly better than previous methods such as CNN, BLSTM, Attention-BLSTM, and SVM.

6. References
[1] Ren P, Chen Z, Ren Z, et al. 2017 Leveraging contextual sentence relations for extractive summarization using a neural attention model. Proc. of the 40th Int. ACM SIGIR Conf. on Research and Development in Information Retrieval ACM pp 95-104.
[2] Neumayer R, Mayer R, Nørvåg K. 2011 Combination of feature selection methods for text categorization. European Conf. on Information Retrieval. Springer, Berlin, Heidelberg, pp 763 - 766.
[3] Li H, Lu Z. 2016 Deep learning for information retrieval. Proc. of the 39th Int. ACM SIGIR conf. on Research and Development in Information Retrieval. ACM pp 1203-1206.
[4] Bayer T, Stein M, Kressel U, et al. 1996 Domain and Language Independent Feature Extraction for Statistical Text Categorization R.
[5] Johnson R, Zhang T. 2015 Effective Use of Word Order for Text Categorization with Convolutional Neural Network. Proc. of the 2015 Conf. of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies pp103 -112.
[6] Jiang J Y, Lee S J. 2007 A weight-based feature extraction approach for text classification. Innovative Computing, Information and Control, 2007. ICICIC’07. 2nd. International Conf. on. IEEE, pp 164-164.
[7] Mikolov T, Le Q V. 2013 Sutskever I. Exploiting Similarities among Languages for Machine Translation J.
[8] Bahdanau D, Cho K H, Bengio Y. 2017 Neural Machine Translation by Jointly Learning to Align and Translate J.
[9] Graves A, Mohamed A, Hinton G. 2013 Speech recognition with deep recurrent neural networks. Acoustics, speech and signal processing (icassp), 2013 IEEE Int Conf. on. IEEE pp 6645-6649.
[10] Sutskever I, Vinyals O, Le Q V. 2014 Sequence to sequence learning with neural networks. Advances in neural information processing systems. pp 3104-3112.
[11] Huang J, Yao S, Lyu C, et al. 2017 Multi-Granularity Neural Sentence Model for Measuring Short Text Similarity. Int. Conf. on Database Systems for Advanced Applications. Springer, Cham, pp 439-455.
[12] Katakis I M, Varlamis I, Tsatsaronis G. 2014 Pythia: Employing lexical and semantic features for sentiment analysis. Joint European Conf. on Machine Learning and Knowledge Discovery in
Databases. Springer, Berlin, Heidelberg, pp 448-451.

[13] Huang W J, Liu C L. 2013 *NCCU-MIG at NTCIR-10: Using Lexical, Syntactic, and Semantic Features for the RITE Tasks* NTCIR.

[14] Zhou Y, Li C, Xu B, et al. 2017 *Hierarchical Hybrid Attention Networks for Chinese Conversation Topic Classification* Int. Conf. on Neural Information Proc. Springer, Cham, pp 540-550.

[15] Zhou Y, Xu J, Cao J, et al. 2017 *Hybrid attention networks for Chinese short text classification* J.Computación y Sistemas, 2017, 21(4) pp 759-769.

[16] Graves A. 2015 *Generating Sequences with Recurrent Neural Networks* J.

[17] Zhou P, Shi W, Tian J, et al. 2016 *Attention-based bidirectional long short-term memory networks for relation classification* Proc. of the 54th Annual Meeting of the Association for Computational Linguistics vol 2, pp 207-212.

[18] Pennington J, Socher R, Manning C. 2014 *Glove: Global vectors for word representation* Proc. of the 2014 conf. on empirical methods in natural language processing (EMNLP) pp1532-1543.

[19] Kinga D, Adam J B. 2015 *A method for stochastic optimization* Int. Conf. on Learning Representations (ICLR).