Scientific Research Publishing

Journal of Intelligent Learning Systems and Applications, 2018, 10, 121-134
http://www.scirp.org/journal/jilsa
ISSN Online: 2150-8410
ISSN Print: 2150-8402

A Sentence Similarity Estimation Method Based on Improved Siamese Network

Ziming Chi*, Bingyan Zhang*

College of Artificial Intelligence, Nankai University, Tianjin, China
Email: zimingchi@163.com, zhangbingyan@126.com

How to cite this paper: Chi, Z.M. and Zhang, B.Y. (2018) A Sentence Similarity Estimation Method Based on Improved Siamese Network. *Journal of Intelligent Learning Systems and Applications*, **10**, 121-134.
https://doi.org/10.4236/jilsa.2018.104008

Received: August 31, 2018
Accepted: October 21, 2018
Published: October 24, 2018

Copyright © 2018 by authors and Scientific Research Publishing Inc.
This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).
http://creativecommons.org/licenses/by/4.0/
Open Access

Abstract

In this paper we employ an improved Siamese neural network to assess the semantic similarity between sentences. Our model implements the function of inputting two sentences to obtain the similarity score. We design our model based on the Siamese network using deep Long Short-Term Memory (LSTM) Network. And we add the special attention mechanism to let the model give different words different attention while modeling sentences. The fully-connected layer is proposed to measure the complex sentence representations. Our results show that the accuracy is better than the baseline in 2016. Furthermore, it is showed that the model has the ability to model the sequence order, distribute reasonable attention and extract meanings of a sentence in different dimensions.

Keywords

Sentence Similarity, Sentence Modeling, Similarity Measurement, Attention Mechanism, Fully-Connected Layer, Disorder Sentence Dataset

1. Introduction

In recent years, semantic processing has attracted a huge amount of research interests [1], since the information scale requires great labor cost; and using such technology is far more economical. To be specific, textual understanding, especially sentence understanding, content search functions, and optimize Question Answering systems are important missions. When researchers are facing tons of articles, the information generated by machines, which regard the main topic of each passage, is useful. In addition, retrieving information sometimes requires identifying the meaning of different key sentences. For instance, an excellent QA system needs to comprehend the questions and choose the optimal answers from

*Co-first author

knowledge base. However, in actual, there is still a long way to go. Articles possessing similar ideas are always different sizes, containing large varieties of syntax. Furthermore, sentences have different lengths and structures. We present an approach to the subtask of deriving meaning from text, while aiming to analyze the similarity among sentences. That is to say, when given two sentences, the algorithms we present below will provide their level of similarity.

To address this problem, Jonas and Aditya [2] generated Siamese neural network, a special recurrent neural network using the LSTM, which generates a dense vector that represents the idea of each sentence. By computing the similarities of both vectors, the output would be labeled from 0 to 1, where 0 means irrelevant and 1 means relevant. Because of the structure of recurrent neural networks, especially the Long Short-Term Memory model of Hochreiter and Schmidhuber [3] can accept the variable length inputs, the length and structure's problems can be solved easily. The Siamese neural network performs very well according to three evaluation metrics: Pearson correlation (r), Spearman's ρ, and mean squared error for the SICK semantic textual similarity task [2]. Nevertheless, drawbacks remain in the Siamese neural network. Because of only employing the last hidden state's vector to represent each sentence, the crucial information in sentence may be attached less importance, and therefore the final vector alone cannot represent the idea of the sentence efficiently. In addition, the simple similarity function $g\left(h\left(a\right),h\left(b\right)\right)=\exp\left(-\left\|h_{T_1}\left(a\right)-h_{T_2}\left(b\right)\right\|_1\right)$ (vectors representing the idea of sentence) used in the model may not represent the computation of similarity accurately compared with the neural networks.

As a consequence, attention mechanism comes into being. Attention has been largely studied in Neuroscience and Computational Neuroscience. It is particularly originated from visual attention: many animals focus on specific parts of their visual inputs to compute the adequate responses and similar to the neural computation as we need to select the most pertinent piece of information, rather than use all available information. This efficient method has been applied to many Deep learning networks like speech recognition, translation, reasoning, and visual identification of objects.

In this paper, we employ the Siamese neural network and develop innovation points as follows. We amplify the contribution of important elements in the final representation, using an attention mechanism [4]. Each of the intermediate state would be set a weight which decides their contribution. Moreover, we rely on the dataset download from Stanford web, which includes around 360,000 couples of sentences. The dataset is larger and more abundant than the SICK dataset used by [2]. Finally, we replace the exponent similarity with a fully connected feed forward layer [5] so as to predict the similarity level. The fully connected layer (FNN) learns a special function of input variables (vector representing the sentence), making it possible to compare two sentences' similarity.

2. Related Work

Comparison of sentence similarity is a basic and significant task across diverse

NLP applications, such as question answering [6], information retrieval [7] [8] and paraphrase identification [9] [10]. Most early researches on measurement of sentence similarity are based on feature engineering, which incorporates both lexical features and semantic features. [6] employed the WordNet based semantic features in the QA match task. [11] provided Microsoft Research Paraphrase Corpus (MRPC) for paraphrase identification task. [9] revealed that it is helpful for classifying false paraphrase cases with the dependency-based features in MRPC. [12] modeled sentence pairs utilizing the dependency parse trees. However, due to the excessive reliance on the manual designing features, these methods are suffering from high labor cost and non-standardization.

Recently, because of the huge success of neural networks in many NLP tasks, especially the recurrent neural networks (RNN), many researches focus on the using of deep neural networks for the task of sentence similarity. [2] proposed a Siamese neural network based on the long short-term memory (LSTM) [3] to model the sentences and measure the similarity between two sentences. [13] combined a stack of character-level bidirectional LSTM with Siamese architecture to compare the relevance of two words or phrases. [14] introduced a ConvNet variant which integrates various differences across many convolutions at varying scales to infer sentence similarity. [15] proposed the skip-thoughts model which extends the skip-gram method of word2vec from the word to sentence level. [16] generalized the order-sensitive chain-structure of standard LSTMs to tree-structured network topologies using Tree-LSTMs. [17] and [18] dealt with semantic similarity between community-based question-answer pairs. These models, however, model the sentences mainly using the final state of RNN which are limited to contain all information of the whole sentence.

Since [19] and [20] first applied attention mechanism in machine translation successfully, attention has been widely used in NLP area, such as text re-construction [21] [22] and text summarization [23] [24]. The attention mechanism also been introduced to the task of sentence similarity. The early work mainly focused on the weighted generation of each attention [25] [26] [27]. Recently the interaction between two sentences has been studied. [28] presented CAN network to pay attention on the generation of the hidden state of one sentence with the help the other sentence's hidden states and attention information. [29] uses GAN to extract the same information between two sentences which are used to measure the similarity of two sentences. In this paper, we focus on the generation of attention weight and ignore the interaction between sentences. And we propose to use fully-connected layer to replace the Manhattan distance measure to improve the performance of the attention mechanism.

3. Methodology

3.1. Framework

In this paper, our model is composed of two sub-models: sentence modeling and similarity measurement. In the sentence modeling part, we used a Siamese architecture [30] consisting of two sub-networks to get two sentences representation respectively. Each sub-network also has three layers: word embedding layer, LSTM layer and attention layer. As for the similarity measurement part, we use the fully-connected layer and logistic regression layer to compute the similarity of two sentence representing vectors from the sentence modeling part. The complete model architecture is shown as **Figure 1**.

The input of our model is two sentences, the words sequence of the first sentence $X_1 = \left(x_1^1, x_2^1, \ldots, x_{T_1}^1\right)$, the second words sequence of the second sentence $X_2 = \left(x_1^2, x_2^2, \ldots, x_{T_1}^2\right)$, where T_1 and T_2 are the number of the words of the two sentences.

3.2. Sentence Modelling

The sentence modeling part can process the sentence from word tokens into a fixed length vector. The aim of the sentence modeling part is to learn a function which can map a sentence to an appropriate vector which is favor for similarity measurement.

Embedding Layer. The word embedding layer try to map every word token in to a fix-sized vector E. The size of E is d_{modeling}. In our model we use the 300-dimensional GloVe word vectors, which are trained based on the global word co-occurrence [31].

LSTM/BiLSTM Layer. We use the bidirectional LSTM to model the sentence with the input-word embedding vectors E. Due to the gradients vanishing problem of RNN, we used the LSTM which can learn long range dependencies. Take sentence $X = \left(x_1, x_2, \ldots, x_T\right)$ as example, RNN update its hidden state h_t using the recursive mechanism.

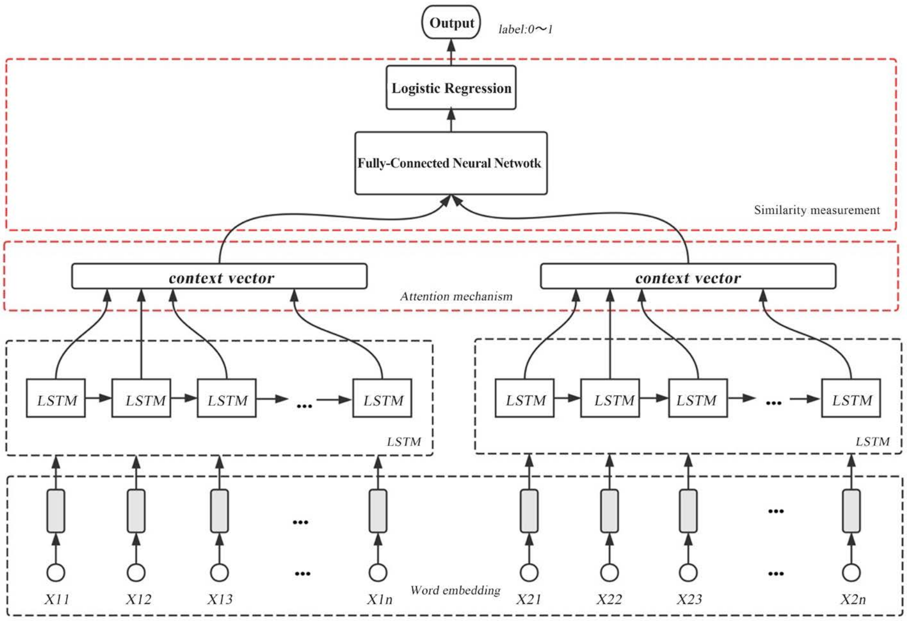

Figure 1. Siamese LSTM with context-attention mechanism and fully-connected neural layer.

$$h_t = \text{sigmoid}\left(Wx_t + Uh_{t-1}\right)$$

The LSTM also sequentially updates a hidden-state representation, but these steps also rely on a memory cell containing four components (which are real-valued vectors): a memory state c_t, an output gate that determines how the memory state affects other units, as well as an input (and forget) gate it (and h_t) that controls what gets stored in (and omitted from) memory based on each new input and the current state.

$$i_t = \text{sigmoid}\left(W_i x_t + U_i h_{t-1} + b_i\right)$$

$$f_t = \text{sigmoid}\left(W_f x_t + U_f h_{t-1} + b_f\right)$$

$$\tilde{c}_t = \tanh\left(W_c x_t + U_c h_{t-1} \vert + b_c\right)$$

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

$$o_t = \text{sigmoid}\left(W_o x_t + U_o h_{t-1} + b_o\right)$$

$$h_t = o_t \odot \tanh\left(c_t\right)$$

where W_i, W_f, W_c, W_o, U_i, U_f, U_c, U_o are weight matrices and b_i, b_f, b_c, b_o are bias vectors.

The BiLSTM contains two LSTM: forward LSTM and backward LSTM. The forward LSTM read the sentence from x_1 to x_T, while the backward LSTM read the sentence fromx$_T$ to x_1. The two LSTMs take in the word sequence in each order and generate two hidden state sequence respectively, H_f from forward LSTM and H_t from backward LSTM. The hidden state h_i^f integrates all the information of the words preceding the word x_i and h_i^b integrates the information of the words behind the word x_i. We obtain the final word annotation of x_i by concatenating the hidden states h_i^f and h_i^b.

$$h_i = h_i^f \Vert h_i^b,\ h_i \in R^{2L}$$

where || denotes the concatenation operation and L the size of each LSTM. Therefore, each word x_i can have an appropriate annotation h_i which contains the information from both directions. The BiLSTM structure is shown as Figure 2.

In this paper, we did experiment both on LSTM and BiLSTM. When we use LSTM, we model the sentence only use the forward direction.

$$h_i = h_i^f,\ h_i \in R^L$$

Attention Layer. The attention layer can use all the word annotations to form the sentence representationr. The attention mechanism can calculate a weight a_i for each word annotation h_i according the importance. The final sentence representation is the weighted sum of all the word annotations using the attention weight. In this paper, we adopted a similar attention mechanism as [32]. In this layer, a context vector u_h is introduced, which can be interpreted as a fixed query. This query helps to identify the informative words and it is randomly initialized and jointly learned with the rest of the attention layer weights.

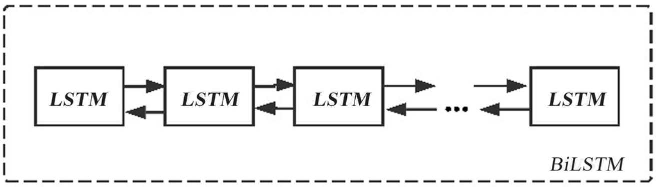

Figure 2. BiLSTM layer for sentence modeling.

$$e_i = \tanh\left(W_h h_i + b_h\right),\ e_i \in [-1,1]$$

$$a_i = \frac{\exp\left(e_i^T u_h\right)}{\sum_{i=1}^{T} \exp\left(e_t^T u_h\right)},\ \sum_{i=1}^{T} a_i = 1$$

$$r = \sum_{i=1}^{T} a_i h_i,\ r \in R^{2L}$$

where W_h, b_h, and u_h are the learnable parameters.

3.3. Similarity Measurement

The similarity measurement model functions as a binary classifier, which can learn the hidden function from the sentence representations to the class label. Our model is designed as an end-to-end model. The sentence modeling part and similarity measurement part can be trained together.

Fully-Connected Layer. Each sentence modeling part outputs a fix-sized vector to represent the sentence respectively. We use one fully-connected layer to measure the similarity of the vectors. The input of this layer is the final representation: the concatenation of two sentence representations r_1 and r_2.

$$r = r_1 \| r_2,\ r \in R^{4L}$$

We choose the tanh (hyperbolic tangent) as this layer's activation function. Then the final representation passes through the fully-connected layer and output a vector for the logistic regression layer.

$$c = \tanh\left(W_c r + b_c\right)$$

Logistic Regression Layer. The regression layer took in the vector c and output a single value s between the 0 and 1 which stands for the degree of the similarity.

$$s = \text{sigmoid}\left(W_s r + b_s\right)$$

If s larger than 0.5, this sentence pair will be classified into relevant; Otherwise, it will be classified into irrelevant.

3.4. Assessment & Loss Function

To evaluate the performance of our model and check the effectiveness of every innovation, two metrics are used, namely accuracy (ACC), mean square error (MSE). The predicted label is 1 when the output $s \geq 0.5$. Otherwise, the predicted label is 0. For each sentence pair, the loss function is defined by the cross-entropy of the predicted and true label distributions for training:

$$\text{Loss} = y\log(s) + (1-y)\log(1-s)$$

where y is the true label, and s is the output which is probability of the label 1 and ($1-s$) is the probability of the label 0.

4. Experiment

4.1. Experiment Design

4.1.1. Dataset

In order to assess our proposed ideas, we utilize a large dataset downloaded from Stanford Web to train the model. The dataset includes 367,373 couples of sentences and the corresponding labels range from 0 to 1. It is separated in subsets, test set and training set, randomly. In general, training set has 330,636 couples and test set has 36,737 couples. The labels set by human represent the similarity between sentences. For instance, the relevant sentences "Children smiling and waving at camera" and "There are children present" are labeled by "1" and the irrelevant sentence "A person on a horse jumps over a broken down airplane." and "A person is at a diner, ordering an omelette." are labeled by "0".

Moreover, considering about whether our model is sensitive to the word order, we modify the dataset approximately, disorganizing the word order. The new dataset is named Disorder Set. It is showed as **Table 1**.

The experiment is done by training the Disorder Set and testing the normal word order dataset. If the result accuracy is far lower than training by normal dataset, we can conclude that our model is able to manage the word order in sentences.

4.1.2. Experiment Flowchart

We use the back propagation, which has random gradient descent and small batches whose size is 64, to shrink the cross-entropy loss. It is together with the Adam optimizer [33]. The gradients are clipped at unit criterion.

Compared with grid and random search, we employ the Bayesian optimization [34] method to find optimal hyper-parameter values in a comparative short time. Our LSTM layers' size is 50, BiLSTM's size is 100, and embedding layer's size is 300. Furthermore, dropout 0.2 are set at recurrent connections of the LSTMs. Lastly, a L2 regularization of 0.0001 is added at the loss function.

4.2. Results

To check the effect of our innovations for the model we compare our model with the baseline model displayed in the [2]. The baseline model uses the single directional LSTM without the attention mechanism to model the sentences and apply the Manhattan distance to measure the similarity of the sentence representations. What's more, to evaluate each innovation's contribution, the ablation method is used. We did the experiment on the baseline model, three sub-models and the final model respectively. The three sub-models are the BiLSTM model,

Table 1. Disorder sentences dataset.

Disorder Sentences Dataset	Label
Play on with a a the beach boy. The little on couple a a herself play girl by watch beach.	1
A performing trick uniformed railed is skier a yellow object across a. In is sports winter engaging somebody.	0
…	…

LSTM model with FNNM, LSTM model with attention mechanism. The final model is LSTM model with FNNM and attention mechanism. Table 2 shows the performance of various models on the dataset SNLI. The best result obtained is marked in bold.

We can see that the BiLSTM model performs worse. Compared with the baseline model, the accuracy of BiLSTM decreases 2.4% and the MSE rises 0.02. Therefore, the backward reading can destroy the model's sentence modeling ability. The influence of word order will be discussed in the Section 4.3.1.

To avoid the negative influence of BiLSTM, we test the effectiveness of attention and FNN by using LSTM. From Table 2, a significant improvement on the Acc and MSE can be observed in LSTM with FNN model compared with baseline model. And the performance of LSTM with attention mechanism model became worse. However, when we add attention mechanism to the model with FNN, the accuracy increases 0.6%. We analyzed the representation of the sentence from each model, we found that the representation from the model with attention contains many information. The Manhattan distance is not suitable to judge the similarity of two vectors. However, the fully-connected layer can learn a more complex function that is better to measure the vectors' similarity. Therefore, when we use the fully-connected layer, the attention can help to improve the performance. Therefore, the rationality of FNN and attention can be proven in the experiments.

What is more, we can see that the LSTM with both FNN and attention mechanism get the best performance, obtaining the improvement is up to 4.2% on accuracy and the decrease up to 0.031 on MSE compared with the baseline model.

4.3. Analysis

4.3.1. Sequence Order Analysis

The LSTM model is famous for its ability to model the sequential dependencies of the sentence. Therefore, we tried to use BiLSTM, which integrates the sequential information of both forward and backward direction, to improve the performance on the task of measuring the sentence similarity. However, the three evaluation-metrics both got worse on the BiLSTM model shown on Table 3. The only difference is the considering the backward reading order in the BiLSTM.

Table 2. Experiments result.

Model	Acc (%)	MSE
(Mueller and Thyagarajan, 2016)	81.9	0.134
BiLSTM	79.5	0.154
LSTM + FNN	85.5	0.111
LSTM + attention	81.4	0.137
LSTM + FNN + attention	**86.1**	**0.103**

Table 3. Sequence order test on LSTM model.

Dataset	Acc (%)	MSE
Original Set	81.9	0.134
Disorder Set	81.0	0.140

This finding gave our motivation to check the LSTM's ability of modeling the word order. We created the Disorder Set whose sentences in training set have a random word order in training set and sentences in test set have a normal order. We trained our model on the disordered training set and check its performance on the normal order test set. At first, we checked the performance of the baseline model. Table 3 shows the results.

From Table 3, we can see that the performance of the model trained on the Disorder Set got 0.9 percent decrease on the ACC and a little increase on the MSE. In other words, the model can do well without order information.

Then we do the same experiment with the final model—LSTM with attention and FNN to check the model's ability to model the sequence order. The result is shown in Table 4 and Figure 3.

The result shows that the accuracy decrease is 1.3 percent and MSE also has a more increase without the order information. This phenomenon indicates that our model considers more sequence order, compared with the baseline. Without the word order information, model judged the similarity of two sentences even worse.

4.3.2. Sentence Representation

The sentence representation space is a multi-dimension vector, each dimension measures different meaning. Now we study the geometry of it. Because the l_1 metric is the combination of differences of each word, we assume that particular characteristics can be represented by encoding specific hidden units (the dimension of the sentence representation). The trained model computes the similarity of sentences by comparing the difference of entire characteristics.

We choose several dimensions of sentence representation space to support the idea. The figures showed in Figure 4 describe the values that different sentences possess among these dimensions of h_T.

The hidden unit showed in the top figure learns to distinguish the affirmation

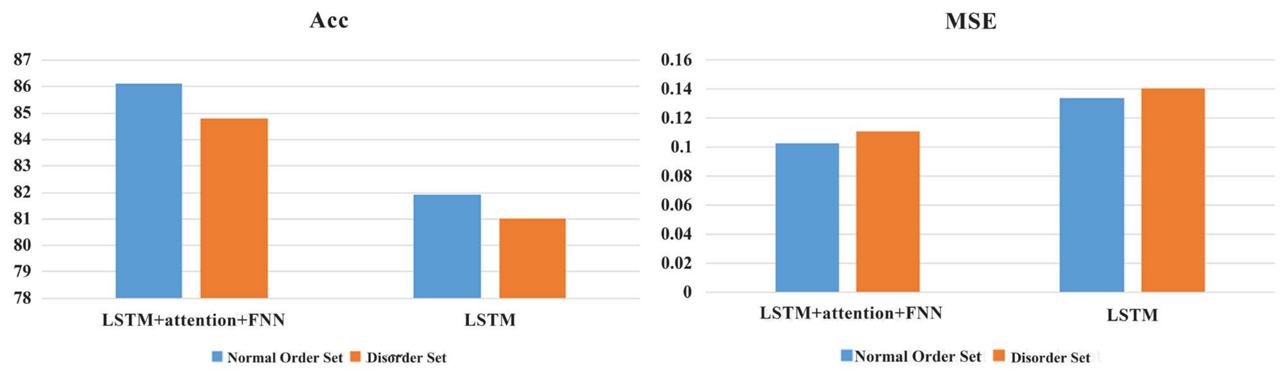

Figure 3. Comparison of sequence order modeling between LSTM and final model.

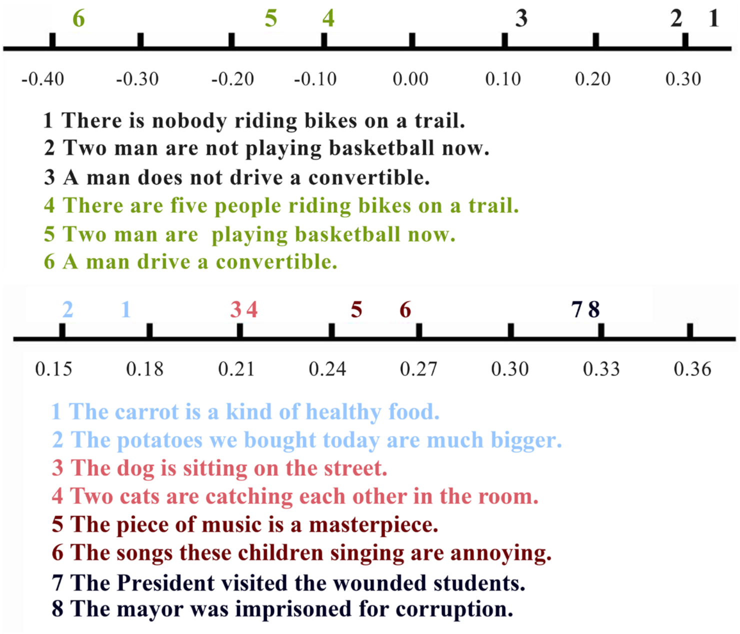
6
5
4
3
2
1
-0.40
-0.30
-0.20
-0.10
0.00
0.10
0.20
0.30
1 There is nobody riding bikes on a trail.
2 Two man are not playing basketball now.
3 A man does not drive a convertible.
4 There are five people riding bikes on a trail.
5 Two man are playing basketball now.
6 A man drive a convertible.
2
1
3 4
5
6
7 8
0.15
0.18
0.21
0.24
0.27
0.30
0.33
0.36
1 The carrot is a kind of healthy food.
2 The potatoes we bought today are much bigger.
3 The dog is sitting on the street.
4 Two cats are catching each other in the room.
5 The piece of music is a masterpiece.
6 The songs these children singing are annoying.
7 The President visited the wounded students.
8 The mayor was imprisoned for corruption.

Figure 4. Analysis on the sentence representation vectors.

Table 4. Sequence order test on final model.

Dataset	Acc (%)	MSE
Original Set	86.1	0.103
Disorder Set	84.8	0.111

and negation, differentiating sentences with words like "not" or "nobody" from the positive sentences, no matter what the rest mean. In the bottom figure we can see the sentences with same theme will cluster together. The sentence modeling part learns to detect the theme of the sentence, such as "vegetable", "animal", "music", "politics" and so on. Therefore, from this analysis, we found that the sentence representation actually extracts many features for classification.

4.3.3. Attention Distribution

The aim of incorporating the attention mechanism for the task is to let the model give the more important word more attention. The method we use is to leverage all the word annotations to model sentence instead of the traditional way, only using the final word annotation. The final sentence representation is the weighted sum of all word annotations according to their importance. In this way, the key point is the weight calculation. In our model, how to distribute the weights to the words is determined by parameters W_h, b_h, and u_h which are automatically learned from the training data. To check the rationality of these weights, we randomly choose several sentence pairs to test it. Figure 5 displays the weights of the words in on sentence. The height of the bar above the word stands for the relevant value of the weight for this word's annotation.

In Figure 5, we can see that the words "parade" and "woman" are assigned larger weights while "Hispanic" and "Latin" get the relevant smaller weights. This phenomenon compiles our intuition that the words "parade" and "woman", compared with the words "Hispanic" and "Latin", play a more important role in determining sentence similarity. The same phenomenon can be found in other sentence pairs, so we can conclude that weights are actually distributed to those words which are the key points to determine the relevance of the two sentences. In this way, an appropriate sentence representation is learned according to the final goal that to find the relevance of sentences. Therefore, the effectiveness of attention can get a good explanation.

5. Conclusions and Future Work

In this paper, we focused on the task of the measurement and the similarity between two sentences. We employed a Siamese network and generated two innovations, including attention mechanism layer and fully connected layer. The dataset we

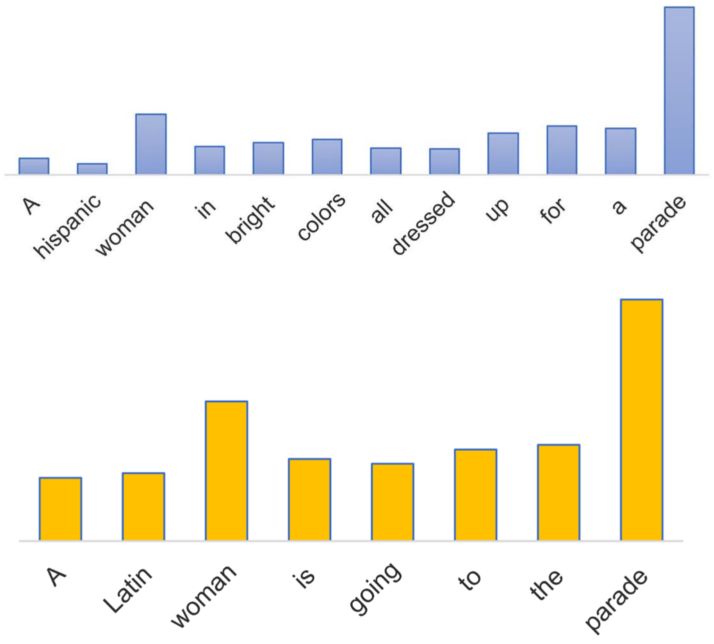

Figure 5. Attention weights distribution among the words in sentence pair.

used is huge and comprehensive, which benefits the training process. In the experiment, the model with both innovations achieved the best performance. Finally, we analyzed our model comprehensively, including the ability of modeling the sequence order, sentence representation and attention distribution. The results showed that our innovation is reasonable and effective.

There still remain plenty of work and limitations to deal with. First, we find the performance of extracting the sentence word order is not good enough by training the dataset with disorder sentences. Compared with the model trained with normal order sentences, the performance of model trained with disorder sentences only has 1.3% decrease in accuracy which is not large enough. Although our model performs better than origin Siamese network, there still needs a lot of works. In ideal, we want the network to perform distinctively when training by normal sentences or disorder sentences, which indicates that the model can truly extract the word order. Furthermore, the evaluation of sentence similarity should be improved. All labels in dataset are labeled by human, so there are plenty of subjective factors inside. For example, two sentences with opposite emotions and similar scenes can be labeled to be irrelevant or relevant, depending on different judgments. What's more, we didn't consider the interaction between two sentences when we model two sentences. When comparing the similarity of two sentences, it is in line with our intuition that the sentence modeling process should take the other sentence into account. The next work we will do is to consider the other sentence's hidden states while calculating the weights in attention mechanism, which may decrease the operation and training time possibly. We will explore all these works in future.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] Marelli, M., Bentivogli, L., Baroni, M., Bernardi, R., Menini, S. and Zamparelli, R. (2014) Semeval-2014 Task 1: Evaluation of Compositional Distributional Semantic Models on Full Sentences through Semantic Relatedness and Textual Entailment. *Proceedings of the 8th International Workshop on Semantic Evaluation*, Dublin, 23-24 August 2014, 1-8.

[2] Mueller, J. and Thyagarajan, A. (2016) Siamese Recurrent Architectures for Learning Sentence Similarity. *AAAI*, **16**, 2786-2792.

[3] Hochreiter, S. and Schmidhuber, J. (1997) Long Short-Term Memory. *Neural Computation*, **9**, 1735-1780. https://doi.org/10.1162/neco.1997.9.8.1735

[4] Rocktäschel, T., Grefenstette, E., Hermann, K.M., Kočiský, T. and Blunsom, P. (2015) Reasoning about Entailment with Neural Attention. arXivpreprint arXiv: 1509.06664.

[5] Baziotis, C., Pelekis, N. and Doulkeridis, C. (2017) Datastories at Semeval-2017 Task 6: Siamese LSTM with Attention for Humorous Text Comparison. *Proceedings of the 11th International Workshop on Semantic Evaluation*, Vancouver, 3-4

August 2017, 390-395.

[6] Yih, W.T., Chang, M.W., Meek, C. and Pastusiak, A. (2013) Question Answering Using Enhanced Lexical Semantic Models. *Proceedings of the* 51*st Annual Meeting of the Association for Computational Linguistics*, **1**, 1744-1753.

[7] Huang, X. and Hu, Q. (2009) A Bayesian Learning Approach to Promoting Diversity in Ranking for Biomedical Information Retrieval. *Proceedings of the* 32*nd international ACM SIGIR conference on Research and development in information retrieval*, Boston, 19-23 July 2009, 307-314.

[8] Wang, Y., Hu, Q., Song, Y. and He, L. (2017) Potentiality of Healthcare Big Data: Improving Search by Automatic Query Reformulation. 2017 *IEEE International Conference on Big Data*, Boston, 11-14 December 2017, 807-816. https://doi.org/10.1109/BigData.2017.8257996

[9] Wan, S., Dras, M., Dale, R. and Paris, C. (2006) Using Dependency-Based Features to Take the "Para-Farce" out of Paraphrase. *Proceedings of the Australasian Language Technology Workshop* 2006, Sydney, 30 November-1 December 2006, 131-138.

[10] Ji, Y. and Eisenstein, J. (2013) Discriminative Improvements to Distributional Sentence Similarity. *Proceedings of the* 2013 *Conference on Empirical Methods in Natural Language Processing*, Seattle, 18-21 October 2013, 891-896.

[11] Dolan, B., Quirk, C. and Brockett, C. (2004) Unsupervised Construction of Large Paraphrase Corpora: Exploiting Massively Parallel News Sources. *Proceedings of the* 20*th international conference on Computational Linguistics*, Geneva, 23-27 August 2004, 350.

[12] Heilman, M. and Smith, N.A. (2010) Tree Edit Models for Recognizing Textual Entailments, Paraphrases, and Answers to Questions. *Human Language Technologies: The* 2010 *Annual Conference of the North American Chapter of the Association for Computational Linguistics*, Los Angeles, 2 June 2010, 1011-1019.

[13] Neculoiu, P., Versteegh, M. and Rotaru, M. (2016) Learning Text Similarity with Siamese Recurrent Networks. *Proceedings of the* 1*st Workshop on Representation Learning for NLP*, Berlin, 11 August 2016, 148-157. https://doi.org/10.18653/v1/W16-1617

[14] He, H., Gimpel, K. and Lin, J. (2015) Multi-Perspective Sentence Similarity Modeling with Convolutional Neural Networks. *Proceedings of the* 2015 *Conference on Empirical Methods in Natural Language Processing*, Lisbon, 17-21 September 2015, 1576-1586. https://doi.org/10.18653/v1/D15-1181

[15] Kiros, R., Zhu, Y., Salakhutdinov, R.R., Zemel, R., Urtasun, R., Torralba, A. and Fidler, S. (2015) Skip-Thought Vectors. *Advances in Neural Information Processing Systems*, Montreal, 7-12 December 2015, 3294-3302.

[16] Tai, K.S., Socher, R. and Manning, C.D. (2015) Improved Semantic Representations from Tree-Structured Long Short-Term Memory Networks. https://arxiv.org/abs/1503.00075

[17] Zhao, Z., Lu, H., Zheng, V.W., Cai, D., He, X. and Zhuang, Y. (2017) Community-Based Question Answering via Asymmetric Multi-Faceted Ranking Network Learning. *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, San Francisco, California, USA, 4-9 February 2017, 3532-3539.

[18] Fang, H., Wu, F., Zhao, Z., Duan, X., Zhuang, Y. and Ester, M. (2016) Community-Based Question Answering via Heterogeneous Social Network Learning. *Proceedings of the Thirty AAAI Conference on Artificial Intelligence*, Phoenix, Arizona, USA, 12-17 February 2016.

[19] Bahdanau, D., Cho, K. and Bengio, Y. (2014) Neural Machine Translation by Jointly

Learning to Align and Translate. https://arxiv.org/abs/1409.0473

[20] Luong, M.T., Pham, H. and Manning, C.D. (2015) Effective Approaches to Attention-Based Neural Machine Translation. https://arxiv.org/abs/1508.04025

[21] Li, J., Luong, M.T. and Jurafsky, D. (2015) A Hierarchical Neural Autoencoder for Paragraphs and Documents. https://arxiv.org/abs/1506.01057

[22] Rush, A.M., Chopra, S. and Weston, J. (2015) A Neural Attention Model for Abstractive Sentence Summarization. https://arxiv.org/abs/1509.00685

[23] See, A., Liu, P.J. and Manning, C.D. (2017) Get to the Point: Summarization with Pointer-Generator Networks. https://arxiv.org/abs/1704.04368

[24] Paulus, R., Xiong, C. and Socher, R. (2017) A Deep Reinforced Model for Abstractive Summarization. https://arxiv.org/abs/1705.04304

[25] Zhang, X., Li, S., Sha, L. and Wang, H. (2017) Attentive Interactive Neural Networks for Answer Selection in Community Question Answering. *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, San Francisco, California, USA, 4-9 February 2017, 3525-3531.

[26] Tan, M., Santos, C.D., Xiang, B. and Zhou, B. (2015) LSTM-Based Deep Learning Models for Non-Factoid Answer Selection. https://arxiv.org/abs/1511.04108

[27] Santos, C.D., Tan, M., Xiang, B. and Zhou, B. (2016) Attentive Pooling Networks. https://arxiv.org/abs/1602.03609

[28] Chen, Q., Hu, Q., Huang, J.X. and He, L. (2018) CA-RNN: Using Context-Aligned Recurrent Neural Networks for Modeling Sentence Similarity. *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, New Orleans, LA, USA, 2-7 February 2018.

[29] Chen, Q., Hu, Q., Huang, J.X. and He, L. (2018) CAN: Enhancing Sentence Similarity Modeling with Collaborative and Adversarial Network. *Proceedings of* 41*st International ACM SIGIR Conference on Research & Development in Information Retrieval*, Ann Arbor, MI, USA, 8-12 July 2018, 815-824.

[30] Bromley, J., Guyon, I., Le Cun, Y., Säckinger, E. and Shah, R. (1994) Signature Verification Using a "Siamese" Time Delay Neural Network. *Advances in Neural Information Processing Systems*, Denver, Colorado, 29 November-2 December 1993, 737-744.

[31] Pennington, J., Socher, R. and Manning, C. (2014) Glove: Global Vectors for Word Representation. *Proceedings of the* 2014 *Conference on Empirical Methods in Natural Language Processing* (*EMNLP*), Doha, Qatar, 25-29 October 2014, 1532-1543. https://doi.org/10.3115/v1/D14-1162

[32] Yang, Z., Yang, D., Dyer, C., He, X., Smola, A. and Hovy, E. (2016) Hierarchical Attention Networks for Document Classification. *Proceedings of the* 2016 *Conference of the North American Chapter of the Association for Computational Linguistics*: *Human Language Technologies*, San Diego, California, 12-17 June 2016, 1480-1489.

[33] Kingma, D.P. and Ba, J. (2014) Adam: A Method for Stochastic Optimization. https://arxiv.org/abs/1412.6980

[34] Bergstra, J., Yamins, D. and Cox, D.D. (2013) Making a Science of Model Search: Hyperparameter Optimization in Hundreds of Dimensions for Vision Architectures. *Proceedings of the* 30*th International Conference on Machine Learning*, Atlanta, Georgia, USA, 16-21 June 2013.