Hindawi
Computational Intelligence and Neuroscience
Volume 2019, Article ID 9543490, 12 pages
https://doi.org/10.1155/2019/9543490

Research Article

A Stacked BiLSTM Neural Network Based on Coattention Mechanism for Question Answering

Linqin Cai, Sitong Zhou, Xun Yan, and Rongdi Yuan

Key Laboratory of Industrial Internet of Things and Networked Control, Ministry of Education, Chongqing University of Posts and Telecommunications, Chongqing 400065, China

Correspondence should be addressed to Rongdi Yuan; yuanrd@cqupt.edu.cn

Received 4 November 2018; Accepted 21 July 2019; Published 21 August 2019

Academic Editor: Friedhelm Schwenker

Deep learning is the crucial technology in intelligent question answering research tasks. Nowadays, extensive studies on question answering have been conducted by adopting the methods of deep learning. The challenge is that it not only requires an effective semantic understanding model to generate a textual representation but also needs the consideration of semantic interaction between questions and answers simultaneously. In this paper, we propose a stacked Bidirectional Long Short-Term Memory (BiLSTM) neural network based on the coattention mechanism to extract the interaction between questions and answers, combining cosine similarity and Euclidean distance to score the question and answer sentences. Experiments are tested and evaluated on publicly available Text REtrieval Conference (TREC) 8-13 dataset and Wiki-QA dataset. Experimental results confirm that the proposed model is efficient and particularly it achieves a higher mean average precision (MAR) of 0.7613 and mean reciprocal rank (MRR) of 0.8401 on the TREC dataset.

1. Introduction

Deep learning forms a more abstract high-level representation attribute feature by combining low-level features to discover the distributed feature representations of data. It provides an effective method for NLP research. In recent years, intelligent question answering in the NLP field has emerged as a prominent discipline research hotspot in both academia and industry, which has been widely used by many influential question answering systems. Answer selection plays a vital role in question answering task, and it mainly encodes QA pair and inputs them into the model to extract the key information and get the corresponding representation [1]. Thus, the main task is to map the question and answer sentences into a joint feature space to generate the codependent representation for them. In the end, an algorithm is utilized to calculate their similarity.

In the past few years, most question answering studies [2–4] were based on knowledge bases and FAQs, which use machine learning to analyze and retrieve keywords. Unfortunately, both of them lack relevant semantic analysis of the questions and answers, which results in a shortcoming of strong artificial dependency and poor scalability.

With the significant innovation of deep learning, deep neural networks are able to availably map the meaning of a single word in a sentence to a continuous representation of the entire sentence, and the meaning of the sentence representation obtained is more complete. Because deep learning reduces the need for manual feature engineering and adapting to new tasks, it has become an important research method for various tasks of NLP in the last several years, and a large number of researchers take advantage of its end-to-end model for sentence semantic analysis to implement question answering tasks. Feng et al. [5] and Wang and Nyberg [6] resorted convolutional neural networks (CNN) and Bidirectional Long Short-Term Memory Networks to capture single sentence semantics, respectively. Nevertheless, both of them ignored the interrelationship between encoded representations of question and answer.

Recently, the model based on the attention mechanism has been explored for question answering. Tan et al. [7] and Nie et al. [8] proposed a BiLSTM model that combines the attention mechanism to construct a better answer representation according to the input question sentences. The model takes the effect of the question on the answer list encoding into account, but they ignore the effect of the answer on the encoding representation of the question, which will cause some deviations in the final prediction result. For instance, the question 1 is "Michael, what are you eating?" and the question 2 is "Michael, why are you eating so much?" and the answer is "Yeah, I'm eating a hamburger." The words "what" & "eating" in question 1 and the words "I'm" & "eating hamburger" in answer have a certain semantic association, and we could easily infer that the answer is corresponding to the question 1. It means that each answer has some intrinsic connection with the question, and to some extent, the question representation is affected by different answers. In addition to analyzing the answers from the questions, we can also infer some results about the questions from the answers.

In this paper, we construct a deep learning architecture for question answering, where questions and answers are limited to a single sentence. The cores of our architecture are two distributed sentence models working in parallel, based on a stacked BiLSTM neural network. We map questions and answers to the corresponding distribution vectors and finally calculate the semantic similarity between them. BiLSTM neural networks have been widely used in recent years to deal with NLP issues [9–11]. Zhang and Ma [12] established a new deep learning model based on BiLSTM networks to accomplish the answer selection task and achieved favorable results. Motivated by this work, we utilize the stacked BiLSTM deep neural network that incorporates the coattention mechanism to semantically understand and model the QA pair, thus allowing model to capture long dependency sentence-level features and generate deeper codependent representations for the QA pair. Additionally, the cosine similarity and the Euclidean distance are reconciled as a new metric to measure the semantic similarity and distance between the questions and the answers. Experiments are settled on the Text REtrieval Conference 8-13 QA dataset and WikiQA dataset. Comparison shows that our experimental model achieved the best experimental results.

The main contributions of this paper are summarized as follows:

(i) A stacked BiLSTM neural network is resorted to attain the vector representation of the input sentence, which can effectively capture the semantics of the sentence.

(ii) Our model combines coattention mechanism and attention mechanism to encode sentences to obtain the interaction and influence between the QA pair.

(iii) The cosine similarity and the Euclidean distance are reconciled to calculate the degree of matching between two vectors. This method is able to take the distance and angle relationship between vectors into consideration.

The rest of this paper is organized as follows. Section 2 gives a brief review of related work. Section 3 presents the proposed framework and method for question answering. Section 4 is a detailed analysis and summary of the experimental results. We will draw a conclusion and discuss the next work in Section 5.

2. Related Work

Research in question answering has been greatly boosted by the Text REtrieval Conference series since 1999. Recently, a number of related works [12–15] have proposed many efficient models for question answering. We compare and correlate the proposed stacked BiLSTM neural networks, coattention mechanism, and scoring metric with our other methods in the literature as follows.

2.1. Long Short-Term Memory Neural Networks. Previously, traditional research approaches concentrated on syntactic matching between the questions and answers. Punyakanok et al. [3] was the earliest to propose the general question and answer matching model via dependency tree models. Later, both Heilman and Smith [2] and Khan et al. [16] presented a probabilistic tree edit algorithm to model sentence. Yao et al. [17] constructed a linear-chain conditional random field based on TREC-QA dataset, which extracted the answer as the answer sequence labeling problem of the tree editing sentence. Moreover, Zhou et al. [4] resorted lexical model based on word relations to select answer sentences. But these traditional models rely excessively on external conditions such as manual labeling of information, which requires a large amount of related work to achieve.

In the recent work of question answering, the mainstream is based on deep learning methods. Yih et al. [18] and Wang et al. [19] developed a semantic parsing framework by a semantic similarity model using convolutional neural networks. Wang and Nyberg [6] used a stacked BiLSTM network to sequentially read words from the question and answer sentences, which did not require any syntactic parsing or external knowledge resources such as WordNet. However, these models failed to consider the codependent representations of the questions and answers. Thus, we add attention mechanism to the deep neural networks to capture the associations between the QA pair.

2.2. Coattention Mechanism. The attention mechanism is appropriate for inferring the mapping relationship between different modal data extremely. It can help a framework like a codec to properly acquire the interrelationships of multiple content models, thus expressing more effectively [1]. There are plenty of related works having explored the attention mechanism in question answering. Based on bidirectional recurrent neural networks, Bahdanau et al. [20] added the attention mechanism to the model to encode and decode the sentence in machine translation. Zhang et al. [21] examined inner attention mechanism and outer attention mechanism in discourse representation for implicit discourse relation

recognition. The result showed a marvelous improvement on marco-$F1$ point is 1.61%. Inspired by the related work in Bahdanau et al. [20] and Fu et al. [22], Tan et al. [7] and Xiang et al. [23] successively proposed an attention mechanism based on bidirectional single-layer LSTMs for question-answer matching, which is able to construct better answer representations according to the input question. Meanwhile, Lu et al. [24] took the lead in presenting a hierarchical coattention model for visual question answering. They used the coattention mechanism to compute a conditional representation of the image given the question and a conditional representation of the question given the image. Enlightened by this work, Xiong et al. [10] presented a dynamic coattention network (DCN) to obtain the co-dependent representations of question and document, and they used a dynamic point decoder to sort potential answers. The experiment achieved 0.8% EM and 2.1% $F1$ improvement on SQuAD dataset. A more refined coattention model was proposed by Zhang and Ma [12]. The author combined the coattention mechanism with the attention mechanism to encode the representation of questions and answers, and this model significantly utilized the inner relationship between questions and answers to enhance the experiment results. Our research also adopts a similar coattention mechanism to extract the statement features.

2.3. Scoring Mechanism. In many previous works such as Liu [25] and He et al. [26], cosine similarity has been proven to be an effective metric for evaluating the similarity between two chord vectors, and it has been widely used in complex queries and matching in recent years. However, Lee et al. [27] resorted the Euclidean distance as the classification decision-making function to measure the average distance between the new data point and the support vectors from different categories, and the data showed that it is efficient. Feng et al. [5] proposed two novel metrics GESD (Geometric mean of Euclidean and Sigmoid Dot product) and AESD (Arithmetic mean of Euclidean and Sigmoid Dot product) in their answer selection task. They proposed two metrics that are the best among all the comparison metrics. In the work of Yin et al. [15], the cosine similarity and the Euclidean distance were separately used to calculate the sentence similarity and measure the semantic distance between different sentences. The result revealed that the simultaneous use of two evaluation mechanisms is superior to using only cosine similarity metric. Unlike the previous research, our approach improves and optimizes previous methods by reconciling the two functions. Our results show that the method is efficient.

3. Proposed Question Answering Model

In this section, we describe the proposed question answering model based on deep learning, which is optimized based on the architecture of Tan et al. [1] and Xiong et al. [10]. The overview of the framework is constructed in Figure 1.

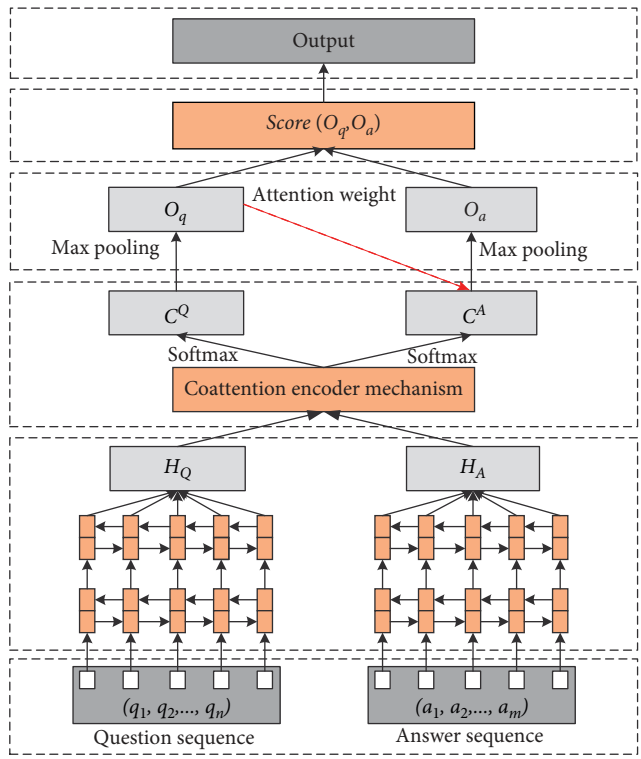

Figure 1: Framework of the proposed neural network model.

In Figure 1, we first utilize the pretrained GloVe to construct word embedding layer, and this word embedding provides the vector representation for each question and its candidate answers. Second, the stacked BiLSTM neural network serves as an encoder that extracts hidden features from each input sentence. Corresponding representations can be obtained by the questions based on the coattention mechanism. After entering the question vector into the maximum pooling, the attention mechanism is used to generate an answer embedding according to the question representation. At last, we combine cosine similarity and Euclidean distance to measure the degree of matching between the question vector and the answer vector.

3.1. A Stacked BiLSTM Neural Network. LSTM networks architecture was originally developed by Hochreiter and Schmidhuber [28]. More formally, an input sequence vector $x = (x_1, x_2, \ldots, x_n)$ is given, where n indicates the length of the input sentence. The core structure of the LSTM is the use of three control gates to control a memory cell activation vector c. The first forget gate determines how much of the cell state c_{t-1} at the previous time is retained until the current cell state c_t; the second input gate determines the extent to which the input x_t of the network is saved to the current cell state c_t; the third output gate determines how much of the cell state c_t is transmitted to the current output value h_t of the LSTM networks. The three gates are a fully connected layer, and its input is a vector and the output is a real number in [0, 1]. The basic LSTM cell architecture is shown in Figure 2, and its representation is as follows:

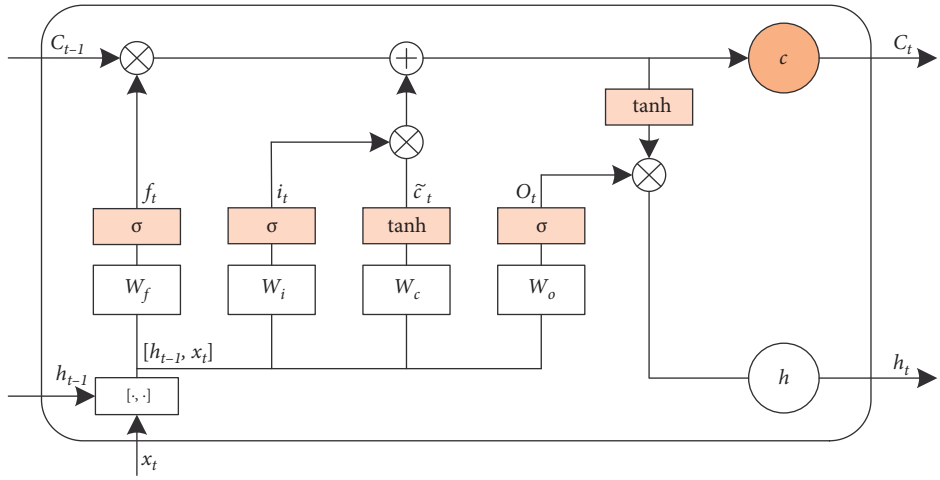

Figure 2: Architecture of Long Short-Term Memory cell.

$$\begin{aligned}
\text{Input gates: } i_t &= \sigma\left(W_{ix}x_t + W_{ih}h_{t-1} + b_i\right),\\
\text{Forget gates: } f_t &= \sigma\left(W_{fx}x_t + W_{fh}h_{t-1} + b_f\right),\\
\text{Output gates: } o_t &= \sigma\left(W_{ox}x_t + W_{oh}h_{t-1} + b_o\right),\\
\text{Cell states: } c_t &= f_t * c_{t-1} + i_t * \tanh\\
&\quad \cdot \left(W_{cx}x_t + W_{ch}h_{t-1} + b_c\right),\\
\text{Cell outputs: } h_t &= o_t * \tanh\left(c_t\right),
\end{aligned} \tag{1}$$

where σ is the logistic sigmoid function, x_t indicates t-th word vector of the sentence and h_t indicates the hidden state, W terms and b terms, respectively, represent weight matrices (e.g., W_{xf} represents the forget gate weight matrix) and bias vectors (e.g., b_i represents the input gate bias vector) for the three gates.

To overcome the shortcoming of single LSTM cell that can only capture previous context but not utilize the future context, Schuster and Paliwal [29] invented bidirectional recurrent neural networks (BRNN) to combine two separate hidden LSTM layers of opposite directions to the same output. With this structure, the output layer is able to utilize related information from both the previous and future context. A BiLSTM calculates the input sequence $x = (x_1, x_2, \ldots, x_n)$ from the opposite direction to a forward hidden sequence $\overrightarrow{h_t} = (\overrightarrow{h_1}, \overrightarrow{h_2}, \ldots, \overrightarrow{h_n})$ and a backward hidden sequence $\overleftarrow{h_t} = (\overleftarrow{h_1}, \overleftarrow{h_2}, \ldots, \overleftarrow{h_n})$. The encoded vector y_t is formed by the concatenation of the final forward and backward outputs, $y_t = [\overrightarrow{h_t}, \overleftarrow{h_t}]$.

$$\begin{aligned}
\overrightarrow{h_t} &= \sigma\left(W_{\overrightarrow{hx}}x_t + W_{\overrightarrow{h}\overrightarrow{h}}\overrightarrow{h}_{t-1} + b_{\overrightarrow{h}}\right),\\
\overleftarrow{h_t} &= \sigma\left(W_{\overleftarrow{hx}}x_t + W_{\overleftarrow{h}\overleftarrow{h}}\overleftarrow{h}_{t+1} + b_{\overleftarrow{h}}\right),\\
y_t &= W_{\overrightarrow{yh}}\overrightarrow{h_t} + W_{\overleftarrow{yh}}\overleftarrow{h_t} + b_y,
\end{aligned} \tag{2}$$

where $y = (y_1, y_2, \ldots y_t \ldots, y_n)$ is the output sequence of the first hidden layer.

Some previous works represented that by stacking multiple BiLSTM in neural networks, the performance of classification or regression can be further improved [30–32]. Moreover, there is some related theoretical support to show that a deep hierarchical model is more efficient in representing some functions than a shallow one [6, 33]. We have defined a stacked BiLSTM network where the output y_t from the lower layer becomes the input of the upper layer. The stacked BiLSTM structure is illustrated in Figure 3:

$$h_t = W_{hi}\overrightarrow{h_t} + W_{hi}\overleftarrow{h_t} + b_h. \tag{3}$$

Defining $Q = (q_1, q_2, \ldots, q_n)$ and $A = (a_1, a_2, \ldots, a_m)$ to represent question sequences and answer sequences, respectively, where n and m indicate the length of the questions and answers, and q_t and a_t indicate the t-th words of the questions and answers. We run a stacked BiLSTM over the questions and answers to obtain their hidden state matrixes H_Q and H_A, and the mathematics is as follows:

$$\begin{aligned}
h_t^q &= \text{sBiLSTM}\left(h_{t-1}^q, h_{t+1}^q, q_t\right), \quad h_0^q = 0,\\
h_t^a &= \text{sBiLSTM}\left(h_{t-1}^a, h_{t+1}^a, a_t\right), \quad h_0^a = h_n^q,\\
H_Q &= \left[h_1^q, h_2^q, \ldots, h_n^q\right] \in R^{d*n},\\
H_A &= \left[h_1^a, h_2^a, \ldots, h_m^a\right] \in R^{d*m},
\end{aligned} \tag{4}$$

where d is the dimension of the hidden state.

3.2. Coattention Mechanism for Question Representation. Here, we implement a coattention mechanism to encode question according to the answer sequences, as shown in Figure 4. Motivated by the work of Xiong et al. [10], we try to enforce more question-answer interactions by designing more careful matrix multiplication, operations, and concatenations in the coattention mechanism.

We first perform matrix multiplication to calculate the affinity matrix L, which includes affinity scores corresponding to all pairs of question and answer words. It can be described as follows:

$$L = H_A^T H_Q \in R^{m*n}. \tag{5}$$

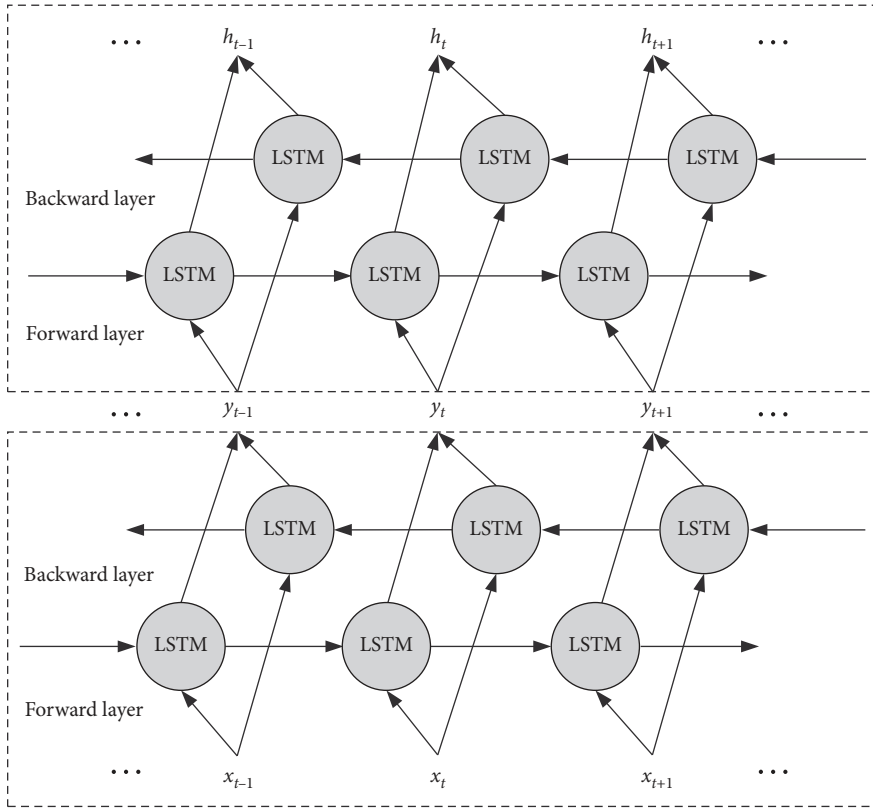

Figure 3: Architecture of the stacked BiLSTM networks.

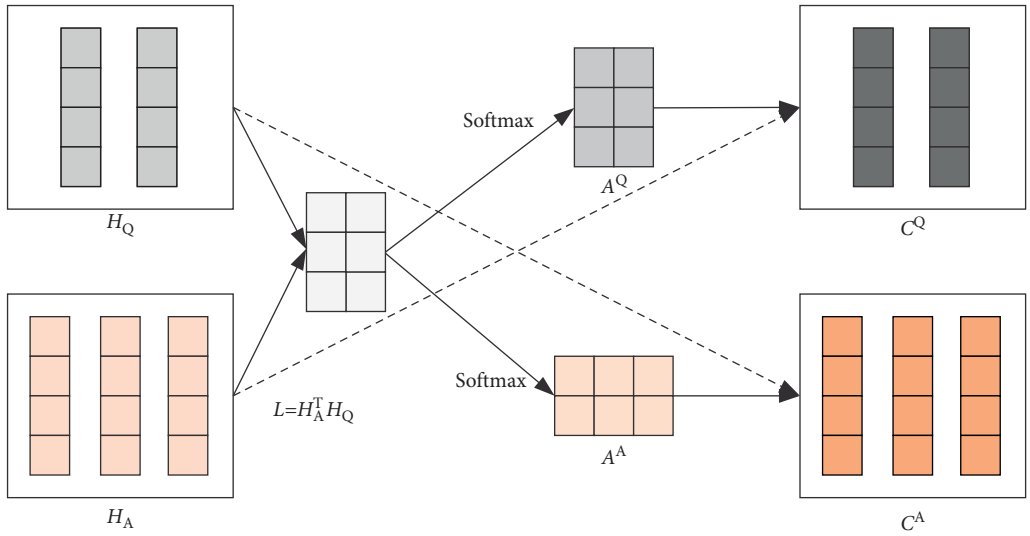

Figure 4: An illustration of the coattention mechanism.

Softmax function is applied to standardize vector elements, and it is effective in dealing with multiclassification and probability distribution problems. Hence, the column- and row-based softmax functions are utilized to generate attention weights for the hidden states of question and answer separately in the following equation:

$$A^Q = \text{soft max}(L) \in R^{m*n},$$
$$A^A = \text{soft max}\left(L^T\right) \in R^{n*m}. \quad (6)$$

In order to obtain the attention vector of the question in light of each word of answers, we concatenate attention weights and affinity matrix to compute new context vectors C^Q and C^A. Here, C^Q and C^A are the results of the interaction between the question and the answer vector:

$$C^Q = H_A A^Q \in R^{d*n},$$
$$C^A = H_Q A^A \in R^{d*m}. \quad (7)$$

3.3. Attentive Attention Mechanism for Answer Representation. To reduce the information loss of stacked BiLSTM, a soft attention flow layer can be used for linking and integrating information from the question and answer words [1, 13]. In the proposed model, the attention mechanism is applied to the output of coattention. We assume that C_t^{Q} indicates t-th attention context vector of the question, and the max pooling is taken to convert the input into a fixed-length vector output O_{q}. Then, the softmax weights of all context vectors $(C_1^{\mathrm{A}}, C_2^{\mathrm{A}}, \ldots, C_m^{\mathrm{A}})$ can be learned autonomously according to O_{q} via the attention mechanism, and the weighted context vector O_{a} of the answer is used as the final representation:

$$\begin{aligned} O_{\mathrm{q}} &= \max_{0<t<=n} C_t^{\mathrm{Q}}, \\ M_{\mathrm{aq}}(t) &= \tanh\left(W_{am} C_t^{\mathrm{A}} + W_{qm} O_{\mathrm{q}}\right), \\ S_{\mathrm{aq}}(t) &\propto \exp\left(w_{ms}^{\mathrm{T}} M_{\mathrm{aq}}(t)\right), \\ O_{\mathrm{a}} &= \sum_{t=1}^{m} C_t^{\mathrm{A}} S_{\mathrm{aq}}(t). \end{aligned} \tag{8}$$

Here, W_{am} and W_{qm} represent the attention matrices of C_t^{A} and O_{q}, respectively. w_{ms} denotes the attention weight vector. The final representation O_{a} of answer is determined by the attention weight $S_{\mathrm{aq}}(t)$ for answer context vector of the t-th word. It is normalized by the softmax function, which is proportional to C_t^{A}. Higher values for $S_{\mathrm{aq}}(t)$ indicate higher correlation between C_t^{A} and the question, and the question vector will get more attention.

3.4. Answer Scoring Mechanism and Objective Function. In this work, we resort a method to reconcile cosine similarity and Euclidean distance to evaluate the degree of matching between the questions and answers. Cosine similarity represents the angle between two vectors, and the Euclidean distance represents the distance between two points in Euclidean space. We hope that the distance between the question and the answer semantic vector to be close enough and the angle is small enough, to maximize the similarity calculation between question and answer pair sentence vectors. The schematic diagram of cosine similarity and Euclidean distance is shown in Figure 5.

A vector representation of the question and answer is obtained from the hidden layer of the model. The cosine similarity and Euclidean distance calculation details are as below. $\mathrm{Score}(O_{\mathrm{q}}, O_{\mathrm{a}})$ is the final match result:

$$\mathrm{Score}_{\mathrm{cosine}}\left(O_{\mathrm{q}}, O_{\mathrm{a}}\right) = \frac{O_{\mathrm{q}} \cdot O_{\mathrm{a}}}{\left|O_{\mathrm{q}}\right|\left|O_{\mathrm{a}}\right|}, \tag{9}$$

$$\mathrm{Score}_{\mathrm{Euclidean}}\left(O_{\mathrm{q}}, O_{\mathrm{a}}\right) = \frac{1}{1 + \left\|O_{\mathrm{q}} - O_{\mathrm{a}}\right\|_2}. \tag{10}$$

Normalize the cosine similarity to the [0, 1] interval and it can be obtained as follows:

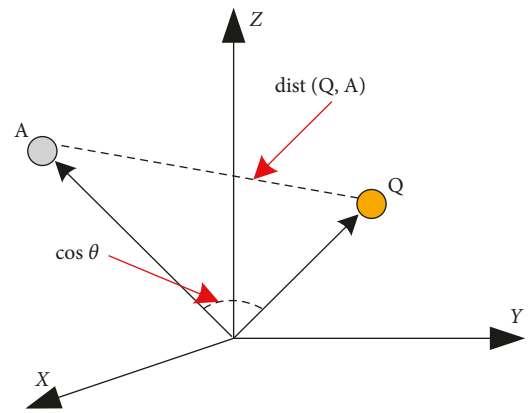

Figure 5: Schematic diagram of cosine similarity and Euclidean distance.

$$\begin{aligned} \mathrm{Score}_{\mathrm{cosine}}\left(O_{\mathrm{q}}, O_{\mathrm{a}}\right) &= 0.5\mathrm{Score}_{\mathrm{cosine}}\left(O_{\mathrm{q}}, O_{\mathrm{a}}\right) + 0.5, \\ \mathrm{Score}\left(O_{\mathrm{q}}, O_{\mathrm{a}}\right) &= \frac{2 \cdot \mathrm{Score}_{\mathrm{cosine}}\left(O_{\mathrm{q}}, O_{\mathrm{a}}\right) \cdot \mathrm{Score}_{\mathrm{Euclidean}}\left(O_{\mathrm{q}}, O_{\mathrm{a}}\right)}{\mathrm{Score}_{\mathrm{cosine}}\left(O_{\mathrm{q}}, O_{\mathrm{a}}\right) + \mathrm{Score}_{\mathrm{Euclidean}}\left(O_{\mathrm{q}}, O_{\mathrm{a}}\right)}, \end{aligned} \tag{11}$$

where $\cdot$ represents the point multiplication operation, $|O_{\mathrm{q}}|$ and $|O_{\mathrm{a}}|$ represent the modulus length of the corresponding vector, respectively. $\|O_{\mathrm{q}} - O_{\mathrm{a}}\|_2$ is the Euclidean distance between two points, and the values of equations (9) and (10) are in the range of [0, 1].

During training, the positive and the negative samples can be input simultaneously by using the hinge loss function. We define the hinge loss function as the training goal as below:

$$L = \max\left\{0, M - \mathrm{Score}\left(O_{\mathrm{q}}, O_{\mathrm{a}+}\right) + \mathrm{Score}\left(O_{\mathrm{q}}, O_{\mathrm{a}-}\right)\right\} + \lambda\|\theta\|, \tag{12}$$

where M is the constant margin, a+ and a− denote the positive answer and the negative answer, respectively. λ and θ represent regularization parameters and neural networks parameters separately.

In the process of training, we utilize the backpropagation algorithm to calculate the gradient $\partial L/\partial\theta$ and update the parameter θ to achieve the minimization of the objective function [34]. Finally, we update the parameters with the minimum objective function $L_{\min}$.

4. Experiments

In this section, we will introduce the detailed information of the experimental implementation, including TREC-QA (8-13) dataset and Wiki-QA dataset, model evaluation indicators, and selection of training parameters, and then, we will carefully analyze the experimental results on different datasets to prove that our proposed model has good accuracy and robustness.

4.1. Implementation Details

4.1.1. Datasets. In this part, we mainly introduce two public datasets, TREC-QA (8-13) dataset and Wiki-QA dataset, and we also introduce the source, data characteristics, and the number of Q&A pairs of these two datasets in detail.

The experiment is operated on the Text REtrieval Conference 8-13 QA datasets (http://nlp.stanford.edu/mengqiu/data/qg-emnlp07-data.tgz) to evaluate our model, which was created by Wang et al. [35] and further elaborated by Yao et al. [17]. As shown in Table 1, we use the 53417 Q&A pairs in TREC 8-12 to train the model, while using 1148 Q&A pairs and 1517 Q&A pairs in TREC 13 for development and testing, respectively. Among them, per question in the development set contains 2.7 positive answers and 11.3 negative answers; per question in the test set contains 3.2 positive answers and 14.0 negative answers. Following Yao et al. [17], candidate answer sentences with more than 40 words and questions with only positive or negative candidate answer sentences are removed from the assessment.

Wiki-QA (https://www.microsoft.com/en-us/download/details.aspx?id=52419) is an open domain Q&A dataset provided by the Microsoft team in 2015. The questions in Wiki-QA are mainly focused on the question of classification, number, and personal information. They are collected and organized by real data of users. The candidate answer statement comes from the topmost text paragraph returned by the Wikipedia input page. As shown in Table 2, after filtering out the question without the correct answer, a total of 1242 Wiki-QA questions were obtained, and 293 correct answer sentences matched the problem, and the data format of Wiki corpus is not much different from TREC-QA (8-13).

In this paper, all experiments were performed on Python, MATLAB, and their optimization toolboxes on a computer with an Intel Core 2 Duo 2.93 GHz processor and a Windows 7 operating system.

4.1.2. Evaluation Metrics. Following the previous works of Wang et al. [35] on this task, two evaluation metrics are utilized for our task: mean average precision (MAP) and mean reciprocal rank (MRR). MAP is the mean average precision score for each query. It reflects the performance of the retrieval system on all queries. The higher the order of related documents returned by the system, the larger the value of the corresponding MAP. MRR indicates the location of the first correct answer associated with the query. The more forward the answer stands, the larger the corresponding MRR value is. Higher values for MAP and MRR indicate better system performance. We resort the official trec_eval (http://trec.nist.gov/trec_eval/) scripts to calculate these metrics:

$$\begin{aligned} \mathrm{MAP} &= \frac{1}{N_{\mathrm{q}}} \sum_{i=1}^{N_{\mathrm{q}}} P_i(r), \\ P_i(r) = P_i(1) &= \frac{1}{n_{ai}} \sum_{k=1}^{n_{a_i}} \frac{k}{\mathrm{rank}_k}, \\ \mathrm{MRR} &= \frac{\sum_{i=1}^{N_{\mathrm{q}}} (1/\mathrm{rank}_i)}{N_{\mathrm{q}}}, \end{aligned} \tag{13}$$

where N_{q} represents the number of all queries and n_{ai} represents the number of all relevant correct answers for query i. $P_i(r)$ represents the average accuracy of the i-th query with recall ratio r. rank_k represents the position of the k-th correct candidate answer in the entire answer sequence after confidence ranking of the candidate answers for the query. rank_i represents the position in which the first correct candidate answer for query i is located in the set of candidate answers.

4.1.3. Experimental Setting. In this paper, different experimental factors are set to test and evaluate our proposed method, and then our method is compared with other most advanced methods under the same dataset. The neural network model is implemented with TensorFlow library. In the course of training, we continuously observe the performance on the test set and select the highest MAP and MRR score parameters for final evaluation. Our implementation is as follows:

(1) Word Embedding. Pretrained GloVe (https://github.com/stanfordnlp/GloVe) [36] is used as the word embedding layer offered by the shared task with 400 dimensions. In addition, each sentence is padded with OOV (out of vocabulary) handling method to the maximum length of fixed lengths, which is 40 words for question and answer. In the candidate answer pool, we set the number of negative answers $K = 5$.

(2) Parameter Initialization. During training, we set the minimum batch size to 40 and refer to the Adam [13] experiment on the TensorFlow to initialize the learning rate to 0.001. The margin M is fixed to 0.2 and the regularization parameter λ is set to $1e-5$. Furthermore, we experimented with single-layer BiLSTM, stacked BiLSTM, and stacked BiLSTM with coattention. Each layer of LSTM has a memory size of 200.

(3) Optimization Algorithm. Adam algorithm [37] is resorted with the decay rate of 0.95 to update the parameters and optimize our model. Subsequently, we add dropout layer after word embedding to avoid overfitting and set dropout rate to 0.5. In order to effectively control the weights within a certain range to avoid gradient explosions, the clip gradients method is used and the gradient threshold is set to 5.

Table 1: Details of TREC-QA (8-13) dataset.

Set	Source	Questions	Positive answers	Negative answers	Length
Train-All	TREC 8-12	1229	6403	47014	≤40
Dev	TREC 13	84	222	926	≤40
Test	TREC 13	100	284	1233	≤40
Total	TREC 8-13	1411	6909	49173	≤40

Table 2: Details of Wiki-QA dataset.

Set	Questions	Positive answers	Negative answers	Length
Train-All	873	1040	19320	16.27
Dev	126	140	2593	15.91
Test	243	100	5872	16.11
Total	1242	293	27785	16.17

4.2. Results and Analysis. In order to verify the validity and accuracy of the algorithm model of the fusion stacked BiLSTM network and the coattention mechanism in the intelligent question answering, we tested and verified the TREC-QA (8-13) dataset and Wiki-QA dataset, respectively, and the experimental results were analyzed and summarized.

4.2.1. Results and Analysis of TREC-QA (8-13) Dataset. We conducted a comparative experiment on single-layer BiLSTM, stacked BiLSTM, and stacked BiLSTM with coattention on the TREC-QA (8-13) dataset. Figure 6 compares the sentences of semantic analysis with or without coattention. Figure 7 reveals the variation in evaluation metrics with the epochs. Table 3 shows the details of experimental results for all mentioned baselines and our proposed model.

(1) Different from the traditional work of Yih et al. [18] and Yu et al. [38], who analyzed the problem from the perspective of sentence structure, it can be obviously discovered that both our experiments and many previous studies such as BiLSTM [1] and CNN [39] have achieved better performance. These researches show that the semantic analyses of sentences are very necessary for NLP tasks and the deep neural networks are able to make the sentence vectors more representatives.

(2) We found that our experimental results of the coattention mechanism were significantly better than most of the above results [1, 8, 38]. Specifically, comparing the results of line 15 with Nie et al. [8], our model achieved 3.52% gain for MAP and 3.83% gain for MRR. These experimental results strongly demonstrated that coattention mechanism and attention mechanism play an important role in improving NLP experimental results. The proper use of them allows the model to pay attention to the output vectors and extract the critical information well in the case of flexible input format. In this way, they can fix the lexical gap between questions and answers while capturing QA pair correlations.

(3) The experimental index of stacked BiLSTM is better than single-layer BiLSTM when compared line 11 and line 12 with line 13 and line 14, respectively. Furthermore, Wang and Nyberg [6] resorted three-layer BiLSTM networks and achieved an increase in MAP (1.52%) and MRR (1.49%) over single-layer BiLSTM of line 11. In general, the appropriate amount of multilayer BiLSTM networks helps to understand the relationship between words and words in a deep level and better extract the characteristics of the sentence itself.

(4) The best MAP (0.7613) and MRR (0.8401) are obtained by incorporating the coattention mechanism into a stacked BiLSTM neural networks and combining cosine similarity and Euclidean distance to calculate the matching degree between two vectors. Our experimental result outperforms the state-of-the-art baselines of Tan et al. [1, 7] by MAP (0.83%) and MRR (0.79%), respectively, which shows that combining the cosine similarity and the Euclidean distance balances the relationship between the angles and distances of two vectors to more effectively match questions and answers.

Firstly, we conducted comparative experiments in the model training process, selected the question and answer statement from the test set of TREC-QA (8-13) randomly, trained the model with/without coattention mechanism, and obtained the corresponding semantic vector representation through different models. The specific content verified that the presence or absence of a coattention mechanism had an impact on the analytical representation of the semantics of the statement. The comparison results are shown in Figure 6.

In Figure 6, the top row of the four matrices represents the semantic parsing results after the action of the coattention mechanism. The following line does not have this mechanism. It can be seen from the figure that after adding the coattention mechanism, the more critical words of the four sentences get more weights; they are more prominent in the process of parsing the expression of the statement, and the verbs such as "is" and "the." The semantic weight ratio of the articles is correspondingly reduced. The analysis shows that the coattention mechanism has the ability to capture the relationship between the statement itself and the statement and can make the semantic expression of the statement more fully without adding additional artificial conditions.

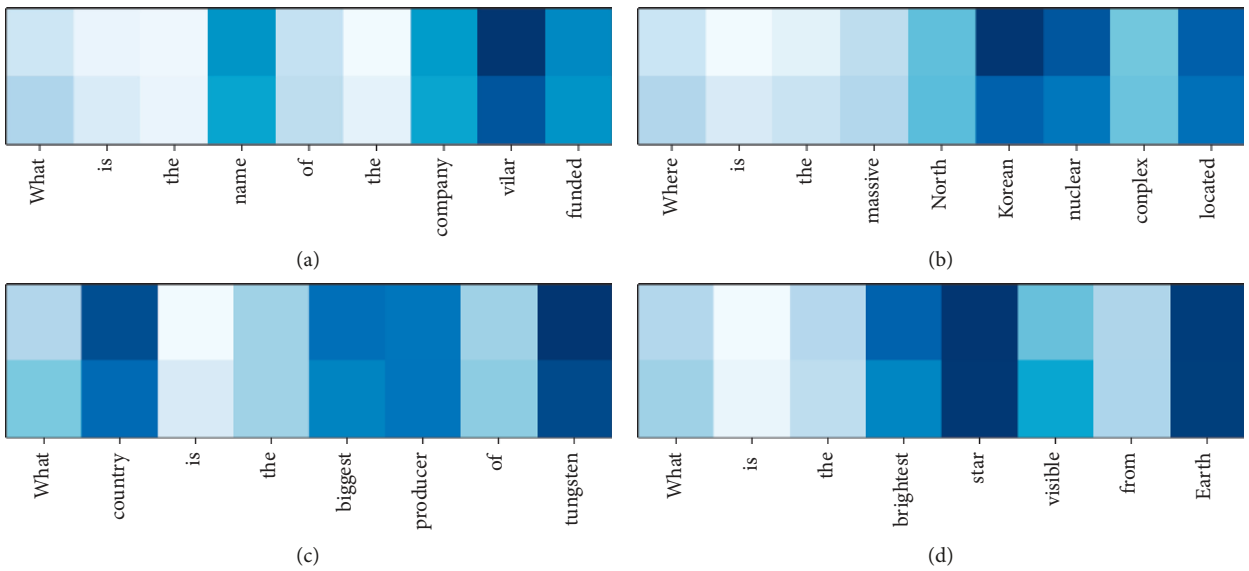

Figure 6: Comparison of sentence semantic analysis with or without coattention.

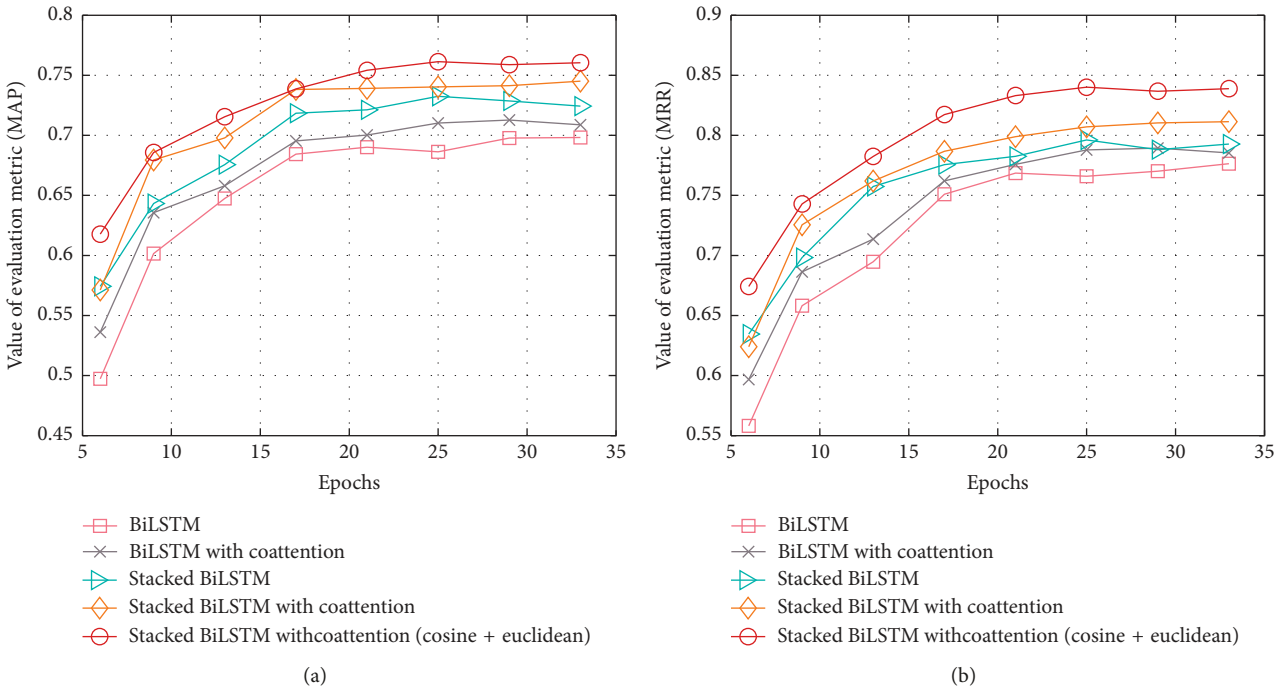

Figure 7: Variation in evaluation metrics with the epochs: (a) MAP and (b) MRR.

Secondly, we verified the epoch sensitivity of the above several models under different iteration periods. Figure 7 shows the variation in MAP and MRR for each model. We performed a comparative experiment of five models, including BiLSTM, stacked BiLSTM, stacked BiLSTM with coattention, BiLSTM with coattention, and stacked BiLSTM with coattention; furthermore, we also presented changes in MAP and MRR for the same model at different epochs.

We performed an epoch-number sensitivity analysis on our proposed model, which varied from 5 to 35. Figure 7 displays the changes in the validation data for MAP and MRR when we change the number of epochs. We observed that both MAP and MRR changed with increasing the number of epochs but tended to be stable after epoch 25. However, the MAP and MRR values of some models have a decreasing trend as the epoch number increases more than 30. It reflects that a certain range of iterations is able to enhance the learning ability of the model and improve the experimental results.

We presented an optimized deep model by using stacked BiLSTM, coattention mechanism, attention mechanism, and a combined similarity metric, and our experimental results

Table 3: Experimental results of different baselines and our proposed model on Train-All data.

Idx	Model	MAP	MRR
1	Probabilistic quasi-synchronous grammar [35]	0.6029	0.6852
2	Tree edit models [2]	0.6091	0.6917
3	Linear-chain CRF [17]	0.6307	0.7477
4	LCLR [18]	0.7092	0.7700
5	Bigram + count [38]	0.7113	0.7846
6	Three-layer BiLSTM + BM25 [6]	0.7134	0.7913
7	Convolutional deep neural networks [39]	0.7459	0.8078
8	BiLSTM/CNN with attention [7]	0.7111	0.8322
9	Attentive LSTM [1]	0.7530	0.8300
10	BiLSTM encoder-decoder with step attention [8]	0.7261	0.8018
11	BiLSTM	0.6982	0.7764
12	Stacked BiLSTM	0.7127	0.7893
13	BiLSTM with coattention	0.7325	0.7962
14	Stacked BiLSTM with coattention	0.7451	0.8114
15	Stacked BiLSTM with coattention (cosine + Euclidean)	**0.7613**	**0.8401**

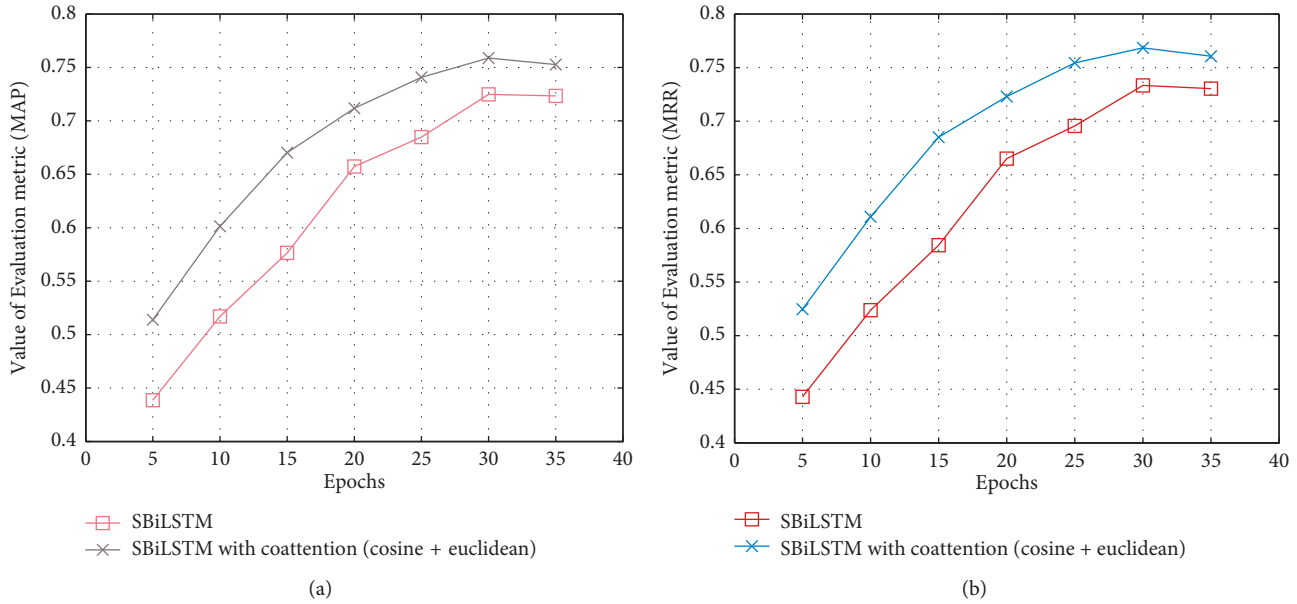

Figure 8: Variation in evaluation metrics with the epochs: (a) MAP and (b) MRR.

are shown in line 11 to line 15 of Table 3. We compared and summarized our observations as follows.

4.2.2. Results and Analysis of Wiki-QA Dataset. We did further comparison experiments on the Wiki-QA dataset. Validation of the model on the Wiki-QA dataset makes the proposed approach more convincing. The parameter initialization and preset aspects of the model on the Wiki-QA dataset are basically consistent with the settings of the TREC dataset, where the batch size of the dataset is 30. Because it is also the order of information retrieval and candidate answer rankings, according to the official evaluation data, the evaluation metrics are selected as MAP and MRR.

We also validated the various models of the design under different epochs on the Wiki-QA dataset, as shown in Figure 8. It can be seen from the figure that the model tends to be stable as the epoch reaches 30 times. When the number of epoch continues to increase, both MAP and MRR have a slight downward trend. The experimental results not only prove that the problem-solving of the model architecture analysis in this paper is effective for the sentence semantics, but also prove that the model has good accuracy and robustness.

The experimental results of each model under the Wiki-QA dataset are shown in Table 4. Compared with the current related research, the model results are superior to most baseline models [40, 41]. Comparing the results of line 1 and line 5 of Table 4, it can be seen that the stacked BiLSTM model is much more accurate than the single-layer LSTM model. In addition, the best experimental results of the model compared with the model in [42], the average accuracy is 0.05% higher than the model in [42].

Table 4: Experimental results of different baselines and our model on the Wiki-QA dataset.

Idx	Model	MAP	MRR
1	LSTM with attention [40]	0.6639	0.6828
2	CNN-Cnt [41]	0.6520	0.6086
3	wGRU-sGRU-Gl2 [42]	0.7537	0.7658
4	wGRU-sGRU-Gl2-Cnt [42]	**0.7638**	**0.7825**
5	Stacked BiLSTM	0.7248	0.7333
6	SBiLSTM-coA (cosine + Euclidean)	**0.7643**	**0.7751**

In the field of intelligent question answering, these data results confirm that the model has some excellent performance in the statement semantic capture representation of questions and answers and can better represent semantic features.

5. Conclusion

In this paper, we proposed a stacked BiLSTM neural network based on the coattention mechanism for question answering. Stacked BiLSTM is used to sentence semantic understanding and modeling; coattention mechanism and attention mechanism are utilized to obtain the co-dependent representation of questions and answers; the combination of cosine similarity and Euclidean distance is used to calculate the similarity between the question and the answer. As reported in Section 4.2, we conduct experiments on the datasets of TREC-QA (8-13) and Wiki-QA, and then experiments on the TREC-QA (8-13) dataset demonstrated that the best MAP (0.7613) and MRR (0.8401) are achieved by using our model. We obtained a certain degree of improvement in MAP (0.83%) and MRR (0.79%) compared with other optimal baselines. Experimental results show that the proposed model is efficient for question answering. Note that, the experiment was only tested on two small datasets. The future work would focus on the implementation of replacing the original coattention mechanism with dynamic coattention network plus (DCN+) and incorporating CNN into the model to improve the experimental results. In addition, the implementation of the proposed model in other large-scale datasets such as SQuAD and SemEval-cQA will be an important issue for the next work.

Data Availability

This work involved data from the Text REtrieval Conference (TREC) 8-13 datasets and Wiki-QA datasets. We used the 53417 Q&A pairs in TREC 8-12 to train the model, while using 1148 Q&A pairs and 1517 Q&A pairs in TREC 13 for development and testing, respectively. All researchers can access the data in the following site: http://nlp.stanford.edu/mengqiu/data/qa-emnlp07-data.tgz, https://www.microsoft.com/en-us/download/details.aspx?id=52419. The data are divided into train data and development/test data.

Conflicts of Interest

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

Acknowledgments

This work was supported by the National Key R&D Program of China (2017YFE0123000), the Innovation Project of Graduate Research in Chongqing (no. CYS19273), and the Key R&D Program of Common Key Technology Innovation for Key Industries in Chongqing (no. CSTC2015zdcy-ztzx60001).

References

[1] M. Tan, C. D. Santos, B. Xiang, and B. Zhou, "LSTM-based deep learning models for non-factoid answer selection," *Computer Science*, vol. 1, 2015.

[2] M. Heilman and N. A. Smith, "Tree edit models for recognizing textual entailments, paraphrases, and answers to questions," in *Proceedings of the 2010 Human Language Technologies Conference of the North American Chapter of the Association for Computational Linguistics, NAACL HLT*, pp. 1011–1019, Los Angeles, CA, USA, June 2010.

[3] V. Punyakanok, D. Roth, and W. Yih, "Natural language inference via dependency tree mapping: an application to question answering," *Computational Linguistics*, vol. 6, no. 9, 2004.

[4] G. Zhou, Y. Zhou, T. He, and W. Wu, "Learning semantic representation with neural networks for community question answering retrieval," *Knowledge-Based Systems*, vol. 93, pp. 75–83, 2016.

[5] M. Feng, B. Xiang, M. R. Glass et al., "Applying deep learning to answer selection: a study and an open task," in *Proceedings of the IEEE Workshop on Automatic Speech Recognition and Understanding, ASRU*, pp. 813–820, Scottsdale, Arizona, December 2015.

[6] D. Wang and E. Nyberg, "A long short-term memory model for answer sentence selection in question answering," in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL-IJCNLP*, pp. 707–712, Beijing, China, July 2015.

[7] M. Tan, C. D. Santos, B. Xiang, and B. Zhou, "Improved representation learning for question answer matching," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL*, pp. 464–473, Berlin, Germany, August 2016.

[8] Y.-P. Nie, Y. Han, J.-M. Huang, B. Jiao, and A.-P. Li, "Attention-based encoder-decoder model for answer selection in question answering," *Frontiers of Information Technology & Electronic Engineering*, vol. 18, no. 4, pp. 535–544, 2017.

[9] C. Yue, H. Cao, K. Xiong, A. Cui, H. Qin, and M. Li, "Enhanced question understanding with dynamic memory networks for textual question answering," *Expert Systems with Applications*, vol. 80, pp. 39–45, 2017.

[10] C. Xiong, V. Zhong, and R. Socher, "Dynamic coattention networks for question answering," in *Proceedings of the International Conference on Learning Representations*, Toulon, France, April 2017.

[11] Y. Ma, H. Peng, T. Khan, E. Cambria, and A. Hussain, "Sentic LSTM: a hybrid network for targeted aspect-based sentiment analysis," *Cognitive Computation*, vol. 10, no. 4, pp. 639–650, 2018.

[12] L. Zhang and L. Ma, "Coattention based bilstm for answer selection," in *Proceedings of the IEEE International Conference*

on Information and Automation, ICIA 2017, pp. 1005–1011, Macau SAR, China, July 2017.

[13] T. Chen, R. Xu, Y. He, and X. Wang, "Improving sentiment analysis via sentence type classification using BiLSTM-CRF and CNN," *Expert Systems with Applications*, vol. 72, pp. 221–230, 2017.

[14] X.-Y. Duan, S.-L. Tang, S.-Y. Zhang et al., "Temporality-enhanced knowledgememory network for factoid question answering," *Frontiers of Information Technology & Electronic Engineering*, vol. 19, no. 1, pp. 104–115, 2018.

[15] W. Yin, H. Schütze, B. Xiang, and B. Zhou, "ABCNN: attention-based convolutional neural network for modeling sentence pairs," *Transactions of the Association for Computational Linguistics*, vol. 4, pp. 566-567, 2016.

[16] M. Khan, F. Kuhn, D. Malkhi, G. Pandurangan, and K. Talwar, "Efficient distributed approximation algorithms via probabilistic tree embeddings," *Distributed Computing*, vol. 25, no. 3, pp. 189–205, 2012.

[17] X. Yao, B. V. Durme, C. Callison-Burch et al., "Answer extraction as sequence tagging with tree edit distance," in *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL HLT*, pp. 858–867, Atlanta, Georgia, June 2013.

[18] W.-T. Yih, X. He, and C. Meek, "Semantic parsing for single-relation question answering," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014*, pp. 643–648, Baltimore, MD, USA, June 2014.

[19] P. Wang, L. Ji, J. Yan et al., "Concept and attention-based CNN for question retrieval in multi-view learning," *ACM Transactions on Intelligent Systems and Technology*, vol. 9, no. 4, pp. 1–24, 2018.

[20] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *Computer Science*, https://arxiv.org/abs/1409.0473, 2014.

[21] B. Zhang, D. Xiong, J. Su, and M. Zhang, "Learning better discourse representation for implicit discourse relation recognition via attention networks," *Neurocomputing*, vol. 275, pp. 1241–1249, 2018.

[22] H. Fu, Z. Niu, C. Zhang, J. Ma, and J. Chen, "Visual cortex inspired CNN model for feature construction in text analysis," *Frontiers in Computational Neuroscience*, vol. 10, 2016.

[23] Y. Xiang, Q. Chen, X. Wang, and Y. Qin, "Answer selection in community question answering via attentive neural networks," *IEEE Signal Processing Letters*, vol. 24, no. 4, pp. 505–509, 2017.

[24] J. Lu, J. Yang, D. Batra et al., "Hierarchical question-image co-attention for visual question answering," in *Proceedings of the 30th Annual Conference on Neural Information Processing Systems, NIPS 2016*, pp. 289–297, Barcelona, Spain, December 2016.

[25] C. Liu, "Discriminant analysis and similarity measure," *Pattern Recognition*, vol. 47, no. 1, pp. 359–367, 2014.

[26] Y. He, S. Xiang, C. Kang, J. Wang, and C. Pan, "Cross-modal retrieval via deep and bidirectional representation learning," *IEEE Transactions on Multimedia*, vol. 18, no. 7, pp. 1363–1377, 2016.

[27] L. H. Lee, C. H. Wan, R. Rajkumar, and D. Isa, "An enhanced support vector machine classification framework by using Euclidean distance function for text document categorization," *Applied Intelligence*, vol. 37, no. 1, pp. 80–99, 2012.

[28] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[29] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Transactions on Signal Processing*, vol. 45, no. 11, pp. 2673–2681, 1997.

[30] Z. Liu, M. Yang, X. Wang et al., "Entity recognition from clinical texts via recurrent neural network," *BMC Medical Informatics and Decision Making*, vol. 17, no. 2, p. 67, 2017.

[31] C. Wang, H. Yang, and C. Meinel, "Image captioning with deep bidirectional LSTMs and multi-task learning," *ACM Transactions on Multimedia Computing, Communications and Applications*, vol. 14, no. 2, 2018.

[32] T. Liu, S. Yu, B. Xu, and H. Yin, "Recurrent networks with attention and convolutional networks for sentence representation and classification," *Applied Intelligence*, vol. 48, no. 10, pp. 3797–3806, 2018.

[33] Y. Bengio, "Learning deep architectures for AI," *Foundations and Trends in Machine Learning*, vol. 2, no. 1, pp. 1–127, 2009.

[34] J. H. Friedman, "Greedy function approximation: a gradient boosting machine," *The Annals of Statistics*, vol. 29, no. 5, pp. 1189–1232, 2001.

[35] M. Wang, N. A. Smith, and T. Mitamura, "What is the jeopardy model? A quasi-synchronous grammar for qa," in *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, EMNLP-CoNLL 2007*, pp. 22–32, Prague, Czech Republic, June 2007.

[36] J. Pennington, R. Socher, and C. D. Manning, "Glove: global vectors for word representation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014*, pp. 1532–1543, Doha, Qatar, October 2014.

[37] D. Kingma and J. Ba, "Adam: a method for stochastic optimization," *Computer Science*, 2014, https://arxiv.org/abs/1412.6980.

[38] L. Yu, K. M. Hermann, P. Blunsom, and S. Pulman, "Deep learning for answer sentence selection," *Computer Science*, 2014, https://arxiv.org/abs/1412.1632.

[39] A. Severyn and S. Moschittiy, "Learning to rank short text pairs with convolutional deep neural networks," in *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2015*, pp. 373–382, Santiago, Chile, August 2015.

[40] Y. Yang, W.-T. Yih, and M. C. Wikiqa, "A challenge dataset for open-domain question answering," in *Proceedings of the Conference Empirical Methods Natural Language Processing*, Lisbon, Portugal, September 2015.

[41] Y. Miao, L. Yu, and P. Blunsom, "Neural variational inference for text processing," in *Proceedings of the International Machine Learning Conference*, pp. 1727–1736, New York, NY, USA, June 2016.

[42] C. Tan, F. Wei, Q. Zhou et al., "Context-aware answer sentence selection with hierarchical gated recurrent neural networks," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 26, no. 3, pp. 540–549, 2018.