Computers, Materials & Continua CMC, vol.62, no.3, pp.1273-1288, 2020

A Novel Bidirectional LSTM and Attention Mechanism Based Neural Network for Answer Selection in Community Question Answering

Bo Zhang[1], Haowen Wang[1, #], Longquan Jiang[1], Shuhan Yuan[2] and Meizi Li[1, *]

Abstract: Deep learning models have been shown to have great advantages in answer selection tasks. The existing models, which employ encoder-decoder recurrent neural network (RNN), have been demonstrated to be effective. However, the traditional RNN-based models still suffer from limitations such as 1) high-dimensional data representation in natural language processing and 2) biased attentive weights for subsequent words in traditional time series models. In this study, a new answer selection model is proposed based on the Bidirectional Long Short-Term Memory (Bi-LSTM) and attention mechanism. The proposed model is able to generate the more effective question-answer pair representation. Experiments on a question answering dataset that includes information from multiple fields show the great advantages of our proposed model. Specifically, we achieve a maximum improvement of 3.8% over the classical LSTM model in terms of mean average precision.

Keywords: Question answering, answer selection, deep learning, Bi-LSTM, attention mechanisms.

1 Introduction

Community question answering (CQA) systems are platforms in which users can ask or answer questions on any topic with few restrictions [Bouziane, Bouchiha, Doumi et al. (2015)]. Some CQA sites (e.g., Yahoo! Answers, Stack Overflow, and Baidu Zhidao) already include millions of users and large answer databases. However, these sites lack the quality control for the millions of answer, which makes it difficult for users to identify useful imformation. Answer selection tasks mainly involve recognizing relevant answers for generating useful question-answer (QA) pairs, which can enrich the knowledge base and improve applications such as chat-bots, search engines, and automatic question answering systems [Zhang, Zhu and Engineering (2016)]. Although some answer

[1] College of Information, Mechanical and Electrical Engineering, Shanghai Normal University, Shanghai, 200234, China.

[2] The Computer Science and Computer Engineering Department, University of Arkansas, Fayetteville, AR, 72703, USA.

[#] The author contributes equally to this work and should be considered co-first author.

[*] Corresponding Author: Meizi Li. Email: limeizi@shnu.edu.cn.

selection studies [Iyyer, Boyd-Graber, Claudino et al. (2014)] have shown performance improvements in recent years, CQA selection tasks remain challenging for two primary reasons. 1) Users can express similar meaning with different word choices in response to the same question, which creates lexical gaps during question matching. 2) Answers often consist of informal, ill-syntax, and variable-length sentences that complicate the modeling of semantic information.

Several studies have investigated sentence structure issues [Juárez-González, Téllez-Valero, Delicia-Carral et al. (2006); Narayanan and Harabagiu (2004); Echihabi and Marcu (2003)], mostly adopting feature engineering and relatively traditional natural language processing technology. These approaches rely on artificial feature extraction rules, such as problem types and answer patterns. They identify the problem by matching common sets in a limited domain and then extracts the answer using the corresponding pattern. Linguistic tools (e.g., grammar and dependency trees) have been introduced for more precise pattern matching. However, this requires manually setting the dialogue scene and developing a targeted dialogue template for each scene (the mode describes the user's possible problems and corresponding answers). As such, this technique is time-consuming and requires significant manual intervention. In addition, sparse data processing suffers from low efficiency and cannot effectively model semantic information because it overemphasizes the modeling of grammar.

As an alternative approach, deep learning methods such as recurrent neural networks (RNNs) and convolutional neural networks (CNNs) have established learning models through high-quality QA corpora. The deep neural networks attempt to model complex dialogue through numerical operations between vectors [Severyn and Moschitti (2013); Iyyer, Boyd-Graber, Claudino et al. (2014)]. RNN models have been widely used in answer selection because of their ability to extract contextual relationships from time series data [Iyyer, Boyd-Graber, Claudino et al. (2014); Wang and Nyberg (2015)]. RNN models often requires a large amount of corpus for active learning of potential syntactic and semantic features in the questions and answers. This compensates for deficiencies in artificial extraction features used to express the problem and helps to improve flexibility and robustness. RNN models can be considered a deep feedforward neural network in which all layers share the same weight. The intended purpose is to learn long-term dependence but theoretical and empirical evidence suggests it is difficult to learn and preserve information over time. As such, long short-term memory (LSTM) neural networks are often adopted to solve this problem by using three "gate" structures [Narayanan and Harabagiu (2004)].

Although above researches have produced breakthroughs in answer selection tasks, problems remain. Inputting words into the neural network requires converting them into vectors. The one-hot code that is often used does not reflect the semantic relationship between these words and long word vectors are likely to cause dimensional issues. Also, the LSTM neural network uses "one-way" time series modeling which prevents word meanings from being extracted effectively out of context. In addition, a word is added at each time step and the hidden state is updated on a recurring basis. The hidden states near the end of the sentence are thus expected to capture more information, which may result in a biased attentive weight towards later words. An attention mechanism can be used to alleviate this weakness by dynamically aligning the more informative parts of sentences

and focusing the answer selection module on keywords.

The primary objective of this study is the design of a new model, capable of identifying important words in QA pairs. The proposed algorithm represents vectors with more effective semantic information which can bridge the lexical gap between a question and its answer and places more attention on the most important words by giving them higher weights to improve answer selection accuracy. This paper includes two main contributions: 1) word embedding technical and Bi-LSTM models are used to encode vectorized representations of QA pairs with a fixed length. This facilitates the extraction of better semantic features in "forward" and "reverse" directions simultaneously. 2) We introduce an attention mechanism which can decide on the importance of other words in a QA pair when generating a word representation in order to focus on words containing key information for answer selection. We perform experiments on datasets acquired form multiple websites and included multiple fields of information. The results show that the performance of proposed model in this paper is significantly better than the conventional deep learning models based on CNN or RNN, both in precision and recall for answer selection tasks.

2 Related work

2.1 Lexical features approaches

Early answer selection models were based on lexical features, relying primarily on the artificial establishment of a set of rules. Common examples include pattern-based, knowledge-based, and noisy channel-based approaches [Bouziane, Bouchiha, Doumi et al. (2015)]. The earliest answer selection tasks in CQA systems were based on a fixed pattern, often making it difficult to distinguish the answer extraction step from the answer selection step. González et al. developed a QA system that used regular expressions to extract candidate answers from the collected answer paragraphs (which were based on the type of question) to retrieve an answer [Juárez-González, Téllez-Valero, Delicia-Carral et al. (2006)]. This system used a Naíve Bayes classifier to select candidate answers according to different characteristics. The candidate with the highest probability of being correct was then selected.

Shen et al. [Shen, Rong, Sun et al. (2015)] used a similar strategy, statistical methods for pattern matching, to extract answers. This approach used pattern confidence to calculate the similarity between a problem and the segment containing the candidate answer, establishing both strict and flexible matching patterns. Strict matching assumes the relationship between sentences to be the same, while flexible matching primarily uses WordNet to establish a relationship between words. Matching weights are then summed to accumulate a score for each provided answer. Priberam also developed a QA system using patterns to extract candidate answers, in which a validation module was used to ensure the correctness of answers [Amaral, Figueira, Martins et al. (2006)]. This was done by applying "sanity check" techniques, such as named entity matching. Narayanan et al. proposed a knowledge-based technique that modeled the relationship between events, entities, and their attributes [Narayanan and Harabagiu (2004)]. This system was capable of parsing documents, extracting related attributes, and associating them with potential answers.

Noisy-channel models are also used for error correction [Khan, Babanezhad, Lin et al. (2015)]. In this process, words are provided containing unusual, omitted, or redundant letters and the model calculates the probability that a given word is associated with another word. Echihabi et al. introduced a probabilistic noisy-channel model for question answering and demonstrated its use in the context of an end-to-end CQA system, in which sentences are input rather than words [Echihabi and Marcu (2003)]. This system calculated the probability of converting a sentence (taken from an information extraction system) into the original question. While these approaches have been shown to be effective for small-scale data, their performance suffers when processing larger data.

These traditional methods focused on syntactic matching between questions and answers. They had to use tedious task of numerous feature extraction that are utilized in traditional linguistic tools.

2.2 Deep learning approaches

Neural network models have recently been proposed to represent the meaning of sentences in a vector space and compare question and answer candidates in the hidden space [Feng, Xiang, Glass et al. (2015); Wang and Nyberg (2015); Xiang, Chen, Wang et al. (2017); Zhang, Li, Sha et al. (2017)]. The deep learning technique represented by convolutional neural networks establishes a joint learning model through high-quality problem-answer corpus and attempts to model complex QA processes using numerical operations between vectors. The advantage of this approach is the transforming of complex semantic analysis, text searching, and answer extraction into a learnable process. To this end, industry scholars have done considerable research on the application of deep learning networks to answer selection tasks.

Severyn et al. used multi-dimensional CNN models to generate vector representations of QA sentences with vector inputs [Severyn and Moschitti (2013)]. Yu et al. applied deep CNN sentence modeling to identify correct answers in CQA datasets [Shen, Rong, Sun et al. (2015)]. Feng et al. proposed a general deep learning framework based on CNNs for solving non-factual CQA tasks [Feng, Xiang, Glass et al. (2015)]. The experimental accuracy of this model (applied to two widely used answer selection benchmark datasets) has been greatly improved, which demonstrates the effectiveness of adding relational information.

Iyyer et al. used RNNs to model textual composition and applied it to CQA tasks in a quiz bowl [Iyyer, Boyd-Graber, Claudino et al. (2014)]. Wang et al. used a Bi-LSTM network to learn eigenvector representations of QA pairs from contextual information in the text [Wang and Nyberg (2015)]. Yang et al. proposed an attention-based neural network architecture that supports multiple input formats to learn key information in QA pairs [Xiang, Chen, Wang et al. (2017)]. Experimental results produced an F1 value of 58.35% for the SemEval-2015 CQA dataset, an increase of 2.21% compared to existing deep neural network-based approaches. Rush. proposed an attention-based summarization (ABS) system for information redundancy and noise problems, focusing on textual information that was useful for abstractive sentence summarization [Zhang, Li, Sha et al. (2017)]. The model shows significant performance gains on the DUC-2004 shared task compared with several strong baselines.

As described above, the effectiveness of vector representation is critical in CQA. Recent

studies using an RNN model have produced good performance. However, in the RNN architecture, input words are processed in a time sequence and hidden states are recurrently updated, assigning larger weights to later words. As such, we propose the use of attention mechanisms to represent QA pairs and resolve this attention bias problem.

3 Model overview

In this study, an attention mechanism is introduced into a Bi-LSTM network and a neural network model is developed from the attentive Bi-LSTM. The model can generate a semantic coding vector containing a sequence attention probability distribution, which was determined by calculating the attention probability of the input sequence. At last, a final feature vector can represent the QA text was generated. Answer selection tasks can be formulated as follows. Given a question q and an answer candidate pool $\{a_1, a_2, \ldots, a_s\}$, identify the best answer candidate a_k, where $1 \leq k \leq s$. Answers in the candidate pool can be divided into positive answers a^+ and negative answers a^- for composite QA pairs. Then, the QA pairs are used as input to the answer selection model, which can obtain the representation vector for each QA pair. Each QA vector pair produces a similarity score to represent semantic distance, as shown in Fig. 1.

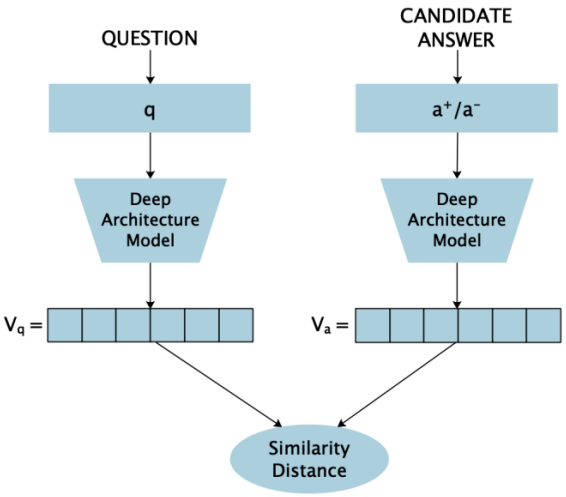

Figure 1: The framework of answer selection task

3.1 Attentive bi-LSTM network-based answer selection model framework

This study investigated answer selection in CQA systems. The framework was divided into three steps: vector representation, feature extraction, and similarity calculation. First, a word embedding technique was used to construct a vector representation of the QA text corpus. Then, a model based on the attentive Bi-LSTM network was developed to extract features. Finally, by calculating attention probability for semantic information in text sequences, the model pays more attention to the problem itself and ignores information in the answer text that is unrelated to the question. This can facilitates optimization of the final feature vector representation. The vector cosine distance similarity was used to measure the match between questions and answers. The structure of the attentive

Bi-LSTM model is shown in Fig. 2.

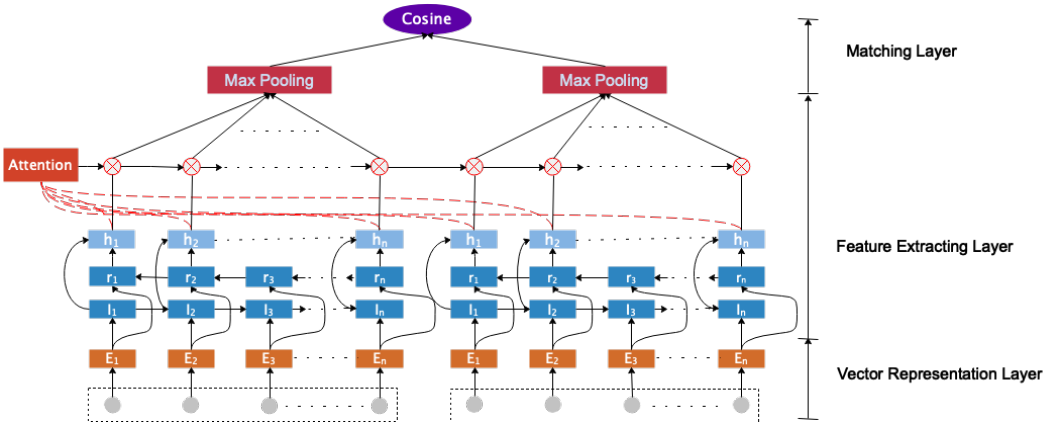

Figure 2: The attentive Bi-LSTM network-based answer selection model architecture

The model includes three primary components:

(1) Word vector representation layer-A word embedding technique was used to construct a vector for each word in the QA corpus. In contrast to other traditional representation models, such as Bag of Words (BOW) [Li, Li, Fu et al. (2016)], each word in the corpus was generated as a vector representation of dimension K. The input sequence can then be represented by $X=\{E_1, E_2, \ldots, E_t\}$, where E_t is a K-dimensional vector, as shown in Fig. 2.

(2) Feature extraction layer - The attentive Bi-LSTM network model proposed in this paper was used as a coding model to extract features from QA pairs. The question sequence X_q was used as input for the network model and the answer sequence X_a was generated by word embedding. The input sequence was encoded by the Bi-LSTM network layer to produce a tensor of dimension N [Liu, Cao and Yu (2018)]. The LSTM network model adds a memory gate mechanism to the RNN network to solve long-distance dependence and vanishing gradient issues. These QA pairs contain a significant amount of irrelevant information, which must be filtered to allow the model to focus on core words. Therefore, an attention mechanism was introduced to calculate an attention probability distribution for the hidden layer of the input sequence after Bi-LSTM encoding process [Ive, Gkotsis, Dutta et al. (2018)]. The vectors I_t and r_t, generated by the forward-LSTM and reverse-LSTM, were combined into the vector h_t for use as input in the next layer (Fig. 2). The attention probability was calculated for each word in the QA sequence and used to measure the influence of the word. The specific calculation method for attention probability of each word is in Section 3.2. Finally, after a maximum pooling, max features were acquired for the text vector to reduce the dimensionality and number of parameters in the training model. The Bi-LSTM network structure is shown in Fig. 3.

(3) Similarity calculation layer: This part mainly uses the cosine distance similarity of the vector as the evaluation criterion of answer selection model [Buck and Koehn (2016)].

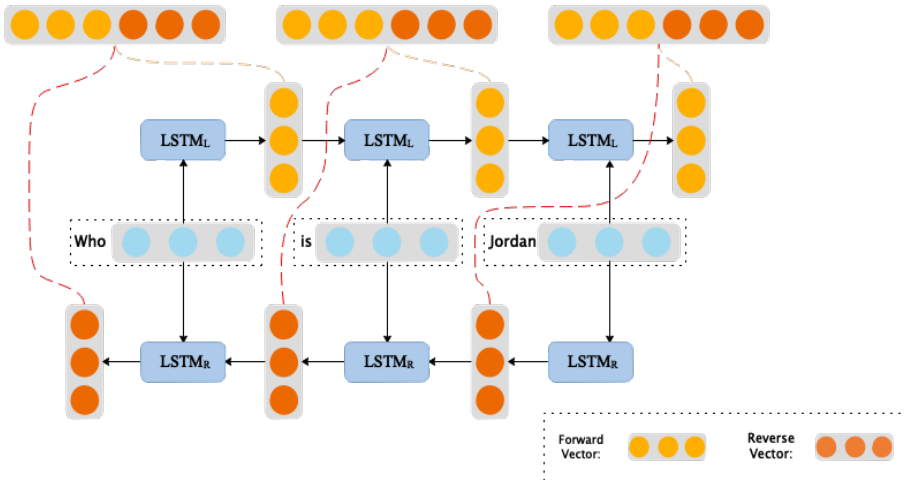

Figure 3: The Bi-LSTM network structure

3.2 Attention probability calculation

The included attention mechanism operates by retaining the intermediate output of the Bi-LSTM encoder input sequence, selectively learning inputs by training a new model, and correlating the output sequence with the results of selective learning. As a result, the probability of each item to be generated in the output sequence depends on which items were selected for the input sequence.

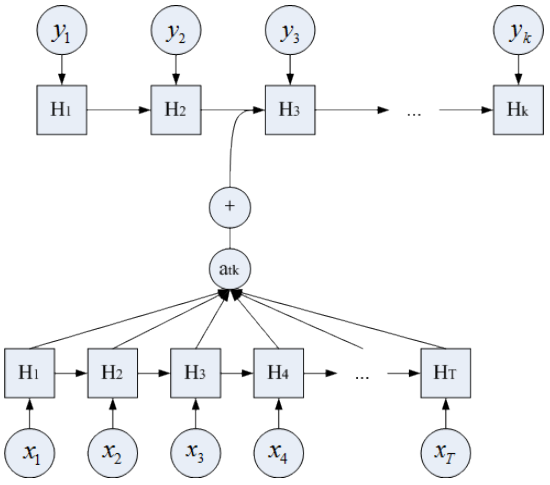

Figure 4: A calculation method for the attention probability distribution

As shown in Fig. 4, x and y are the input sequences of questions and answers, and H is the hidden vector generated by the input sequences. The a_{tk} node is the attention weight of the node k passed to the output t, which essentially determines the influence of the node on the output (i.e., the probability). This is equivalent to adding a single layer deep

neural network to the original model. Higher values of a_{tk} correspond to increased attention from output t being allocated to the input k, thereby increasing its influence. The attention distribution probability a_{ij} can be calculated using the following formula:

$$a_{ij} = \frac{\exp\left(e_{ij}\right)}{\sum_{i=1}^{T} \exp\left(e_{ik}\right)} \tag{1}$$

$$e_{ij} = a\left(s_{i-1}, h_j\right) \tag{2}$$

Here, S_{i-1} first performs a calculation using each h value and then uses a softmax operation to acquire an attention distribution vector for the output of time i in the T input hidden states.

3.3 Attention influence propagation

This model includes two inputs, a sequence of questions (represented as $X_q = \{X_1, X_2, ..., X_t\}$) and a sequence of answers (represented as $X_a = \{X_1, X_2, ..., X_t\}$). The terms $h_q(t)$ and $h_a(t)$ are hidden layer state values for the input sequences of QA text at each time step t. In the Bi-LSTM network structure, the output of the hidden layer is the splicing of the output $\vec{h}$ in the forward LSTM and the output $\overleftarrow{h}$ of the reverse LSTM. The term o_q is vector representation of the question sequence after a max-pooling operation, which is necessary to extract the critical features that can represent a sequence. This can reduce dimensionality and preserve the most important features in a sequence. This operation can be represented as follows:

$$m_{a,q}(t) = W_{am} h_a(t) + W_{qm} o_q \tag{3}$$

$$s_{a,q}(t) \propto \exp\left(W_{ms}^T \tanh\left(m_{a,q}(t)\right)\right) \tag{4}$$

$$\tilde{h}_a(t) = h_a(t) s_{a,q}(t) \tag{5}$$

Here, W_{am}, W_{qm}, and W_{ms} are attention parameters used in the output vector o_q to calculate softmax weights and multiply the answer vector of the current hidden layer to produce a new hidden layer output with attention weights $\tilde{h}_a(t)$.

4 Network training

The vanishing gradient problem is common in deep learning applications [Le and Zuidema (2016)]. Fig. 5 shows a hypothetical cyclic neural network that can predict values after multiple time steps. We assume a neural network model can be used to classify documents or make multiple predictions from a text sequence. After the prediction, the model receives an error and back propagates all time steps in the neural network [Engel and Bershad (1994)]. However, the gradient becomes smaller in each

time step of the backpropagation, eventually becoming so small at the beginning of the sentence that it does not effectively affect parameters needing to be updated. This is because the gradient dl/dh_t is either reduced or increased unless dh_{t-1}/dh_t is equal to one. When this gradient is repeatedly increased or decreased, the gradient of the loss function is increased or decreased exponentially. In neural network training, the optimization of three gradient descent algorithms (Adagrad, RMSprop, and Adam) has primarily been used to solve the problem of gradient disappearance.

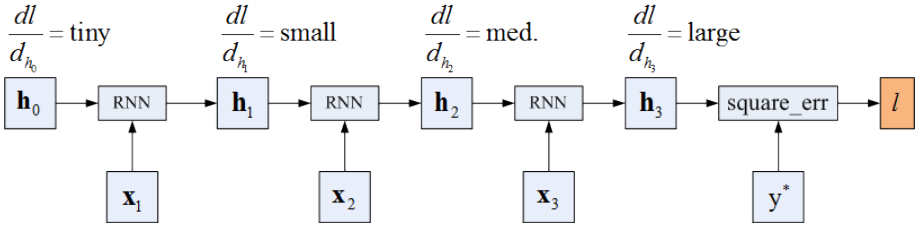

Figure 5: The vanishing gradient problem

Adam (Adaptive Moment Estimation) was used in this study to train the proposed model, which can calculate the adaptive learning rate for each parameter. Adam stores exponential decay averages for the previous squared gradient of AdaDelta, maintaining the average of the exponential decay for the previous gradient. This is essentially an RMSprop with a momentum term that dynamically adjusts the learning rate for each model parameter, using a first-order moment estimate and a second-order moment estimate for the gradient [Le and Zuidema (2016)]. This dynamically adjustment of learning rate makes the parameters trained by the model relatively stable. This calculation is performed as follows:

$$m_t = \mu * m_{t-1} + (1-\mu) * g_t \tag{6}$$

$$n_t = v * n_{t-1} + (1-v) * g_t^2 \tag{7}$$

$$\overline{m}_t = \frac{m_t}{1-\mu^t} \tag{8}$$

$$\overline{n}_t = \frac{n_t}{1-v^t} \tag{9}$$

$$\Delta\theta_t = -\frac{\overline{m}_t}{\sqrt{\overline{n}_t + \varepsilon}} * \eta \tag{10}$$

where m_t and n_t are the first and second moment estimates of the gradient, respectively. These can be considered an estimate of the expectation $E|g_t|$ and $E|g^2{}_t|$. The terms $\overline{m}_t$ and $\overline{n}_t$ are corrections of m_t and n_t, which can be approximated as unbiased estimates of the expectation.

5 Experimental verification

5.1 Datasets

The effectiveness of the answer selection model based on the attentive Bi-LSTM network was verified using the QA pair corpus acquired from multiple websites[3]. These data included information from the National Basketball Association (player and team stats), film and television summaries, and political news posts. The original data were stored in a database and exported in a custom JSON data format, which included more than a thousand tables[4]. Tab. 1 lists team sheet data from the 2006-07 Toronto Raptors season. The statistical information for each dataset is described in Tab. 2. Fig. 6 shows a histogram of question lengths, query lengths, and the number of columns. As seen in the figure, the length of the question, the length of the answer, and the number of columns were mostly concentrated between 10 and 15 characters, which essentially satisfied a normal distribution.

Table 1: The Toronto Raptors 2006-07 season data sheet (partial)

Game	3	4	11
Date	11/5	11/8	11/22
Team	San Antonio	Philadelphia	Cleveland
Team Score	L94-103 (OT)	W106-104 (OT)	W95-87 (OT)
High Points	Chris Bosh (19)	Chris Bosh (29)	Chris Bosh (25)
High Rebounds	Chris Bosh (7)	Chris Bosh (9)	Chris Bosh (14)
High Assists	T. J. Ford (5)	T. J. Ford (7)	Chris Bosh (6)
Location/Attendance	Air Canada Centre/18,098	Air Canada Centre/15,831	Air Canada Centre/19,800
Record	1-2	2-2	3-8

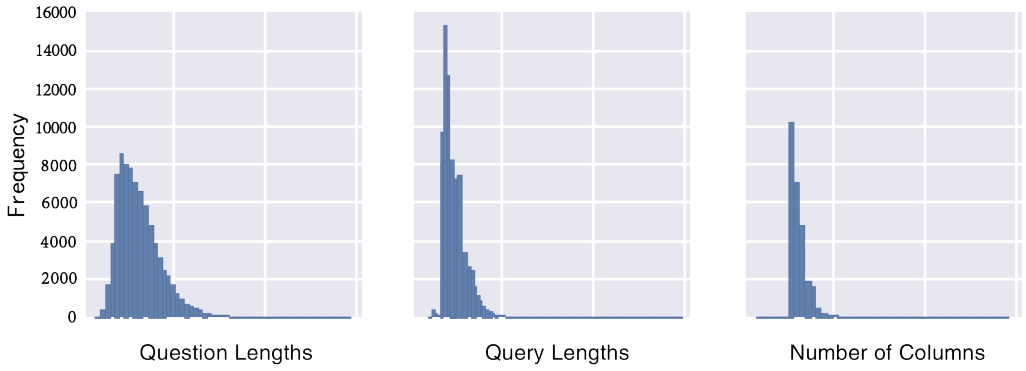

Figure 6: A statistical histogram of question length, query length, and the number of columns

[3] http://www.spox.com, https://www.yahoo.com/news, www.washingtonpost.com
[4] https://github.com/Bynow76/AS_data/

Table 2: Data set statistics

Training Data	Verification Data	Test Data	Total
61297	9145	17284	87726

5.2 Evaluation metrics

The goal of answer selection is to identify the most correct option from a candidate pool, which is essentially a sorting task. This study utilized common text evaluation indices to assess the proposed model, including accuracy, precision, recall, and F1 score. These can be expressed as follows:

$$\text{Accuracy} = \frac{TP + TN}{TP + FN + FP + TN} \tag{11}$$

$$\text{Precision} = \frac{TP}{TP + FP} \tag{12}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{13}$$

$$\text{F1} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{14}$$

5.3 Comparison of training optimization algorithms

The optimizer has a significant impact on the convergence of the learning model during the training process. Therefore, to determine the impact of different optimization algorithms on the answer selection model, a comparative experiment was conducted using five different algorithms (Adam, AdaGrad, SGD, RMSprop, and AdaDelta) [Duchi, Hazan and Singer (2011)].

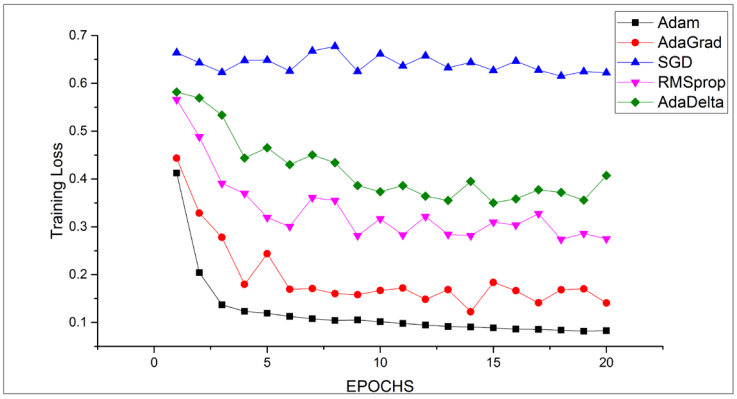

Figure 7: The performance of different optimization algorithms for answer selection tasks

Fig. 7 shows the loss values for different optimization algorithms during answer selection. As seen in the figure, SGD performance was poor and the loss value was not effectively reduced within 20 epochs. Since AdaGrad can adaptively adjust the learning rate, as the

gradient grows from small to large, the learning rate decreases from large to small [Le and Zuidema (2016)]. This produced better convergence during answer selection. Both Adadelta and RMSprop are extensions of AdaGrad, which attempts to reduce the learning rate in a monotonically decreasing trend. The difference is that RMSprop distributes the learning rate by exponentially attenuating the mean of the squared gradient. Adam achieved better training results in a shorter training time, which also verified the analysis in Section 4.

5.4 Comparison of answer selection models

The effectiveness of the proposed model was verified by a comparison with CNN, LSTM. Fig. 8 shows the accuracy rate of the attentive Bi-LSTM network model for training and test data. The black line indicates training data accuracy and the red line indicates verification data accuracy. We mapped each word to a 300-dimensional vector representation with 60 hidden layer nodes in the Bi-LSTM network. To prevent overfitting, the dropout value was set to 0.3 [Srivastava, Hinton, Krizhevsky et al. (2014)] and the batch size was set to 64. The learning step for the corpus model was completed mostly in epoch 5. The Bi-LSTM in the attentive multi-Bi-LSTM included two layers. As seen in Fig. 9, this model is mostly consistent with the attentive Bi-LSTM model. The corpus model has been learned in epoch 30. Parameters in the LSTM model were mostly consistent with these two model settings, the performance of which is shown in Fig. 10. The feature extraction layer used a CNN model with a filter size of 3, a ReLU activation function, and a step size of 1. Model performance is shown in Fig. 11. It is evident from the figure that the model convergence rate with attention mechanisms was significantly faster than the other two models.

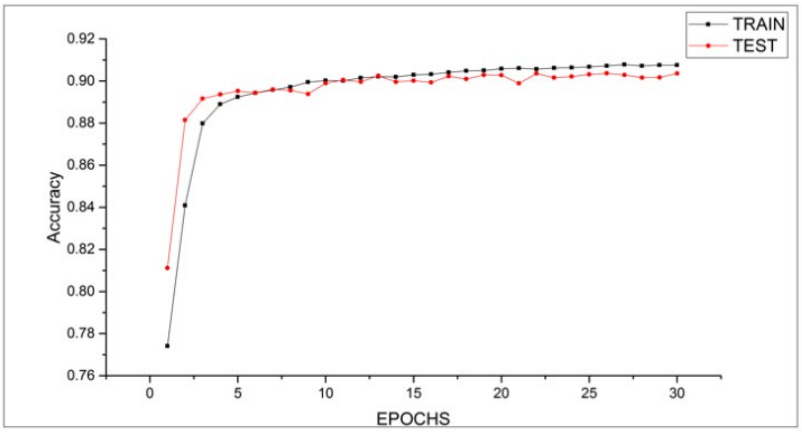

Figure 8: Accuracy curves for the attentive Bi-LSTM model

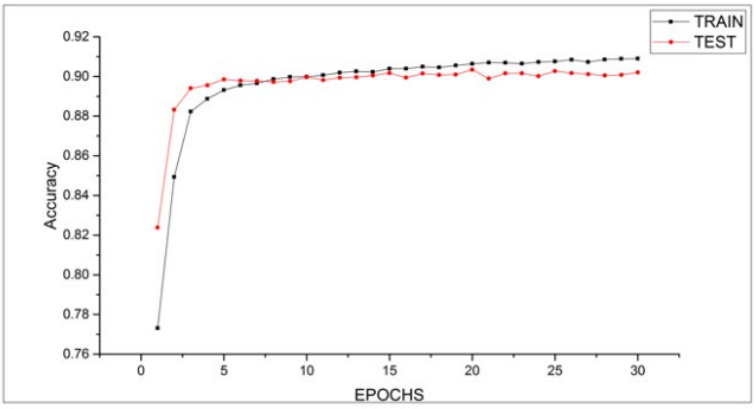

Figure 9: Accuracy curves for the attentive multi-Bi-LSTM model

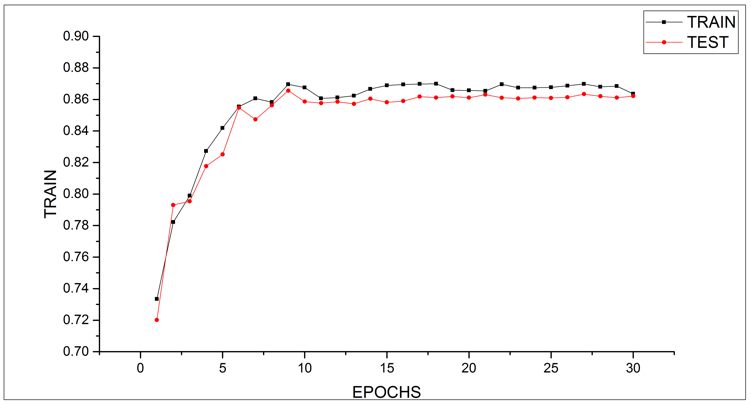

Figure 10: Accuracy curves for the LSTM model

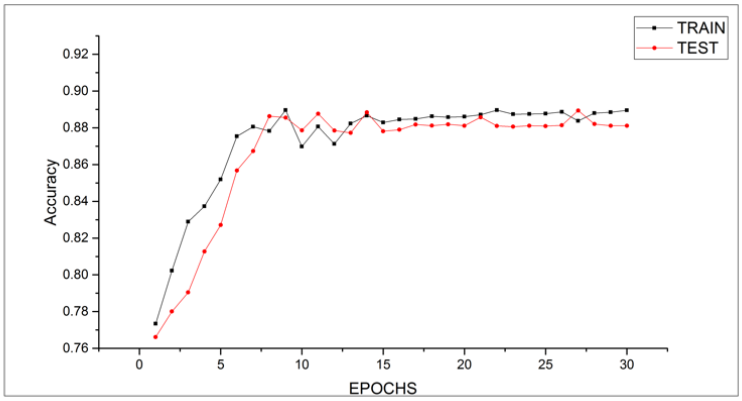

Figure 11: Accuracy curves for the CNN model

5.5 Result and analysis

After parameter tuning, experimental validation was performed using the data described above. The results of this test are shown in Tab. 3, where it is evident that the attentive Bi-LSTM model and the attentive multi-Bi-LSTM model developed in this study have improved answer selection accuracy. All performance indices were higher than for traditional LSTM and CNN models. The plots in Figs. 8-11 demonstrate that the models containing attention mechanisms have converged by epoch 3, which is significantly faster than in other models. The sequence model based on LSTM effectively improved performance, in comparison with CNN. This suggests the LSTM model to be more conducive for capturing context information in answer selection tasks, while CNNs are useful for acquiring local features.

The attentive Bi-LSTM model and the attentive multi-Bi-LSTM models exhibited similar accuracy, outperforming conventional techniques. This indicates that the attention mechanism can significantly improve answer selection efficiency by focusing on useful information. These results also suggest that a single-layer Bi-LSTM network based on this attention mechanism could extract useful features in text samples.

Table 3: Experimental comparison results (%)

	Accuracy	Precision	Recall	F1 Score
Attentive Bi-LSTM	90.7	84.2	72.7	76.9
Attentive Multi Bi-LSTM	90.9	85.4	72.3	77.1
LSTM	86.9	83.1	72.2	75.3
CNN	88.8	82.8	71.3	75.2

6 Conclusions

This study introduced an attention mechanism which can calculate the attentive weights of other words in a QA pair on the current output word and combined it with a Bi-LSTM network to design a novel network for answer selection tasks in CQA. The model considers contextual data and filters out redundant information by assigning different weights. The potential of the proposed method was also demonstrated by comparison with conventional techniques included CNN or LSTM based models, it improved the accuracy of answer selection. In future research, we plan to incorporate other textual features such as location, keyword, and grammar information to explore their impact on model performance.

Acknowledgement: This work was supported in part by the National Natural Science Foundation of China under Grant 61572326, and Grant 61802258; the Natural Science Foundation of Shanghai under Grant 18ZR1428300; the Shanghai Committee of Science and Technology under Grant 17070502800 and Grant 16JC1403000.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

Amaral, C.; Figueira, H.; Martins, A.; Mendes, A.; Mendes, P. et al. (2006): Priberam's question answering system for portuguese. *Accessing Multilingual Information Repositories*.

Bouziane, A.; Bouchiha, D.; Doumi, N.; Malki, M. (2015): Question answering systems: survey and trends. *Procedia Computer Science*, vol. 73, no. 73, pp. 366-375.

Buck, C.; Koehn, P. (2016): Quick and reliable document alignment via TF/IDF-weighted cosine distance. *Proceedings of the First Conference on Machine Translation.*

Duchi, J.; Hazan, E.; Singer, Y. (2011): Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, vol. 12, no. 7, pp. 257-269.

Echihabi, A.; Marcu, D. (2003): A noisy-channel approach to question answering. *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics*.

Engel, I.; Bershad, N. J. (1994): A transient learning comparison of Rosenblatt, backpropagation, and LMS algorithms for a single-layer perceptron for system identification. *IEEE Transactions on Signal Processing*, vol. 42, no. 5, pp. 1247-1251.

Feng, M.; Xiang, B.; Glass, M. R.; Wang, L.; Zhou, B. (2015): Applying deep learning to answer selection: a study and an open task. *IEEE Workshop on Automatic Speech Recognition and Understanding*.

Ive, J.; Gkotsis, G.; Dutta, R.; Stewart, R.; Velupillai, S. (2018): Hierarchical neural model with attention mechanisms for the classification of social media text related to mental health. *Proceedings of the Fifth Workshop on Computational Linguistics and Clinical Psychology: From Keyboard to Clinic*.

Iyyer, M.; Boyd-Graber, J.; Claudino, L.; Socher, R.; Daumé III, H. (2014): A neural network for factoid question answering over paragraphs. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*.

Juárez-González, A.; Téllez-Valero, A.; Delicia-Carral, C.; Montes-y-Gómez, M.; Villaseñor-Pineda, L. (2006): Using machine learning and text mining in question answering. *Evaluation of Multilingual and Multi-Modal Information Retrieval*.

Khan, M. E.; Babanezhad, R.; Lin, W.; Schmidt, M.; Sugiyama, M. (2015): Convergence of proximal-gradient stochastic variational inference under non-decreasing step-size sequence. *Journal of Comparative Neurology*, vol. 319, no. 3, pp. 359-86.

Le, P.; Zuidema, W. (2016): Quantifying the vanishing gradient and long distance dependency problem in recursive neural networks and recursive LSTMs. *Proceedings of the 1st Workshop on Representation Learning for NLP*.

Li, J. Q.; Li, J.; Fu, X. H.; Masud, M. A.; Huang, J. Z. (2016): Learning distributed word representation with multi-contextual mixed embedding. *Knowledge-Based Systems*, vol. 106, no. C, pp. 220-230.

Liu, X.; Cao, D.; Yu, K. (2018): Binarized LSTM language model. *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, vol. 1 (Long Papers).

Narayanan, S.; Harabagiu, S. (2004): Question answering based on semantic structures. *Proceedings of the 20th International Conference on Computational Linguistics*.

Zhang, N.; Zhu, L. J.; Center, E. (2016): A survey of Chinese QA system's question analysis. *Technology Intelligence Engineering*, vol. 2, no. 1, pp. 32-42.

Severyn, A.; Moschitti, A. (2013): Automatic feature engineering for answer selection and extraction. *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*.

Shen, Y.; Rong, W.; Sun, Z.; Ouyang, Y.; Xiong Z. (2015): Question/Answer matching for CQA system via combining lexical and sequential information. *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*.

Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. (2014): Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929-1958.

Wang, D.; Nyberg, E. (2015): A long short-term memory model for answer sentence selection in question answering. *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, vol. 2 (Short Papers).

Xiang, Y.; Chen, Q.; Wang, X.; Qin, Y. (2017): Answer selection in community question answering via attentive neural networks. *IEEE Signal Processing Letters*, vol. 24, no. 4, pp. 505-509.

Zhang, **X. D.; Li, S. J.; Sha, L; Wang, H. F.** (2017): Attentive interactive neural networks for answer selection in community question answering. *Proceedings of Thirty-First AAAI Conference on Artificial Intelligence*.