AMQAN: Adaptive Multi-Attention Question-Answer Networks for Answer Selection

Haitian Yang[1,2], Weiqing Huang[1,2(✉)], Xuan Zhao[3], Yan Wang[1(✉)], Yuyan Chen[4], Bin Lv[1], Rui Mao[1], and Ning Li[1]

[1] Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
{yanghaitian,huangweiqing,wangyan,lvbin,maorui,lining01}@iie.ac.cn
[2] School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China
[3] York University, Ontario, Canada
xuanzhao1003@gmail.com
[4] Sun Yat-sen University, Guangzhou, Guangdong, China
chenyy387@mail2.sysu.edu.cn

Abstract. Community Question Answering (CQA) provides platforms for users with various background to obtain information and share knowledge. In the recent years, with the rapid development of such online platforms, an enormous amount of archive data has accumulated which makes it more and more difficult for users to identify desirable answers. Therefore, answer selection becomes a very important subtask in Community Question Answering. A posted question often consists of two parts: a question subject with summarization of users' intention, and a question body clarifying the subject with more details. Most of the existing answer selection techniques often roughly concatenate these two parts, so that they cause excessive noises besides useful information to questions, inevitably reducing the performance of answer selection approaches. In this paper, we propose AMQAN, an adaptive multi-attention question-answer network with embeddings at different levels, which makes comprehensive use of semantic information in questions and answers, and alleviates the noise issue at the same time. To evaluate our proposed approach, we implement experiments on two datasets, SemEval 2015 and SemEval 2017. Experiment results show that AMQAN outperforms all existing models on two standard CQA datasets.

Keywords: Answer selection · Adaptive multi-attention · Community Question Answering

1 Introduction

In recent years, Community Question Answering (CQA), such as Quora and Stack Overflow, has become more and more popular. Users can ask questions of their own concerns, search for desired information, and expect answers from

F. Hutter et al. (Eds.): ECML PKDD 2020, LNAI 12459, pp. 584–599, 2021.
https://doi.org/10.1007/978-3-030-67664-3_35

expert users. However, with the rapid increase of CQA users, a massive amount of questions and answers accumulate in the community leading to a proliferation of low-quality answers, which not only make CQA difficult to operate normally, but also take users a lot of time to find the most relevant answer from many candidates seriously affecting their experience. Therefore, answer selection in CQA which aims to select the most possibly relevant answer in community becomes very important.

Generally speaking, there are two application scenarios for an answer selection task. The first scenario is to use CQA information retrieval technology to identify whether a given query is semantically equivalent to an existing question in the repository. If these two are semantically equivalent after retrieval, the corresponding answers of the existing question are referred to users as the relevant candidates. The second scenario is treating the existing questions and answers in the archive as "question-answer pairs" to determine whether these question-answer pairs are best match, namely, converting the answer selection task to a classification task. In this research, we focus on improving the performance of answer selection in the second application scenario.

A typical CQA example is shown in Table 1. In this example, Answer 1 is a good answer because it gives "check it to the traffic dept", which is more suitable for the question, while Answer 2, although related to the question, does not provide useful information, considered as a bad answer.

Table 1. An example question and its related answers in CQA

Question subject	Checking the history of the car
Question body	How can one check the history of the car like maintenance, accident or service history. In every advertisement of the car, people used to write "Accident Free", but in most cases, car have at least one or two accident, which is not easily detectable through Car Inspection Company. Share your opinion in this regard
Answer 1	Depends on the owner of the car.. if she/he reported the accidents i believe u can check it to the traffic dept.. but some owners are not doing that especially if its only a small accident.. try ur luck and go to the traffic dept.
Answer 2	How about those who claim a low mileage by tampering with the car fuse box? In my sense if you're not able to detect traces of an accident then it is probably not worth mentioning... For best results buy a new car :)

Compared with questions in other QAs, the given example in Table 1 shows unique characteristics of questions in CQA that is they usually consists of two parts, question subjects and question bodies. A question subject is a brief summary of the question with key words in it while a question body is a detailed

description of the question, including critical information and some extended information. In addition, for both questions and answers, there are a vast amount of redundant information, which affect the performance of answer selection solutions.

In this paper, we propose a novel model, which comprehensively uses question subjects, question bodies, and corresponding answers for the answer selection task. Specifically, the information of answers with their noise filtered out are used as external resources. In order to obtain hierarchical text features, we concatenate word embeddings, character embeddings, and syntactic features as the input of the representation layer. Moreover, we use three heterogeneous attention mechanisms, including self-attention, cross attention, and adaptive co-attention to integrate answer information and effectively obtain text relevance. In addition, we also develop a gated fusion module adaptively integrating answer-based features as well as adopt a filtration gate module as a filter to reduce the noise introduced by answers. Meanwhile, the interaction layer enhances the local semantic information between questions and their corresponding answers. Finally, predictions are made based on the similarity features extracted from question-answer pairs.

The main contributions of our work can be summarized as follows:

(1) We consider the noise issues generated from adding answer information, and study how to integrate answer information into neural network to perform the answer selection task.
(2) We propose a novel method that treats question subjects and question bodies separately, integrating answer information into neural attention model so as to reduce the noise from answers and improves performance for answer selection task.
(3) Our proposed model outperforms all state-of-the-art answer selection models on various datasets.

The remaining of this paper is organized as follows: Sect. 2 gives an overview of the existing techniques related to our work; In Section 3, we introduce our proposed answer selection model, AMQAN, describe the details of the structure; in Sect. 4, we discuss the implementation of AMQAN, explain various parameters and the training process, and compare our model with other baselines; Sect. 5 summarizes the conclusion and suggests the further research potentials in the future.

2 Related Work

Answer selection in community is a challenging and complicated problem. Some work on this task [1–4] have shown the effectiveness mainly in high-quality questions. In the early years, answer selection task highly relied on feature engineering, linguistics tools and other external resources. Nakov et al. [5] studied a wide range of feature types, including similarity features, content features, and meta-features, which are automatically extracted from the SemEval CQA model.

Filice et al. [6] also designed various heuristic features and thread-based features, and yielded better solutions. Although these techniques have achieved good performance, the highly dependence on feature engineering results in indispensability of domain knowledge and an enormous amount of manpower.

Moreover, to successfully accomplish an answer selection task, semantic and syntactic features are also necessary. The most relevant candidate answer can be obtained by loose syntactic alteration through studying syntactic matching between questions and answers. Wang et al. [7] designed a generative model based on the soft alignment of quasi-synchronous grammar by matching dependency trees of question answer pairs. Heilman et al. [8] used a tree kernel as a heuristic algorithm to search for the smallest edit sequence between parse trees and features extracted from these sequences are fed into logistic regression classifier to determine whether an answer is relevant to the given question.

Some researchers focus on studying different translation model, such as word-based translation model and phrase-based translation model. Murdock et al. [9] proposed a simple translation model for sentence retrieval in factoid question answering. Zhou et al. [10] proposed a phrase-based translation model aiming to find semantically equivalent questions in Q&A archives with new queries.

Although research described above have solved a majority of answer selection tasks, there are still a great quantity of issues in suspense mainly because these questions are short in length and various in vocabulary. And in other cases, answer selection approaches solely based on questions themselves, but the questions cannot provide enough useful information, making it difficult to identify the similarity between different questions. However, since answers usually explain questions in detail, they can be used as a very useful supplementary resource. But simply combining answers with their corresponding questions may cause redundancies, which might also reduce the performance of answer selection solutions.

3 Proposed Model

3.1 Task Description

In this research, the answer selection task about community question answering can be described as a tuple of four elements (S, B, A, y). $S = [s^1, s^2, \cdots, s^m]$ represents a question subject whose length is m. $B = [b^1, b^2, \cdots, b^g]$ represents the corresponding question body whose length is g. $A = [a^1, a^2, \cdots, a^n]$ represents the corresponding answer whose length is n. And $y \in Y$ represents the relevance degree, namely, $Y = \{Good, PotentiallyUseful, Bad\}$ to determine whether a candidate can answer a question properly or not. More detailed, Good represents that the answer can provide a proper solution for the question, while PotentiallyUseful indicates that the answer might provide a useful solution to users and bad means the answer is not relevant to the question. Generally, our AMQAN model on the answer selection task in CQA can be summarized as assigning a label to each answer based on the conditional probability $Pr(y \mid S, B, A)$ with the given set $\{S, B, A\}$.

3.2 Overview of Proposed Model

The structure of AMQAN can be represented as six layers distributed layer by layer from bottom to top, including Embedding layer, Encoder layer, Self-Attention layer, Cross Attention layer, Adaptive Co-Attention layer, and Interaction and Prediction Layer. The pipeline of the proposed framework is demonstrated in Fig. 1.

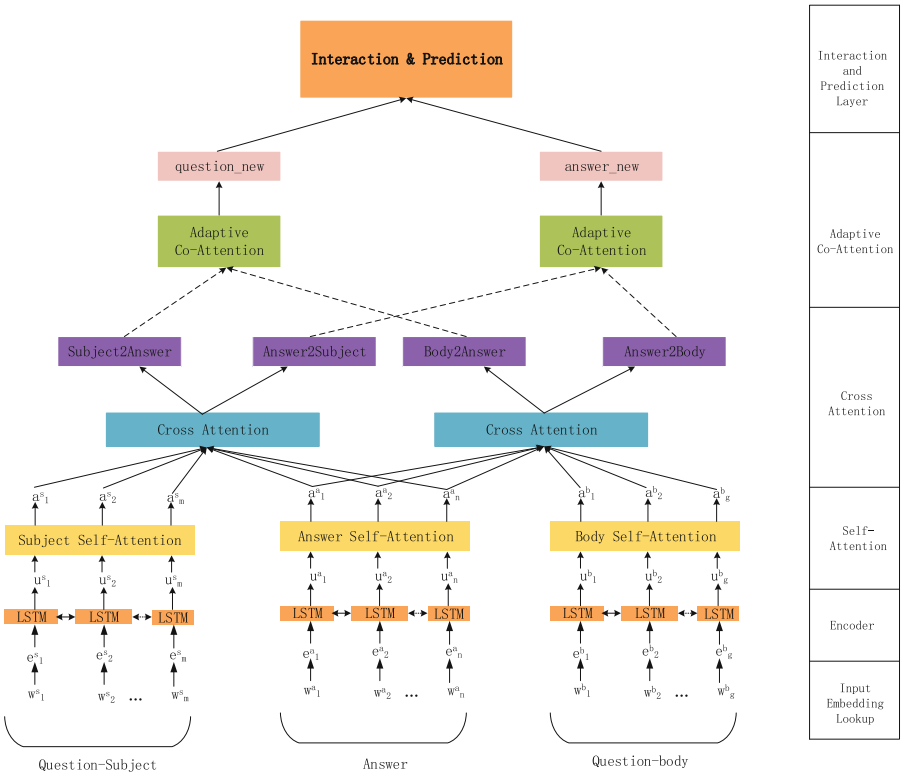

Fig. 1. Overview of our proposed AMQAN model

3.3 Word-Level Embedding

Word embeddings are comprised of three different modules: GloVe word representation trained on the Yahoo! Answers corpus proposed by Pennington et al. [11], character embeddings proposed by Kim et al. [12], and syntactic features based on one-hot encoding proposed by Chen et al. [12]. The relationship of words can be captured more accurately and precisely with the concatenation of three embeddings which are trained on the domain-specific corpus for the reason that texts in CQA are different mainly in grammar and spelling from others. It has also proved character embeddings are effective for OOV (out-of-vocabulary),

especially in CQA tasks and syntactic features based on one-hot encoding can provide more grammar information to make a better query representation.

We define $\{w_t^{subject}\}_{t=1}^{m}$, $\{w_t^{body}\}_{t=1}^{g}$ and $\{w_t^{answer}\}_{t=1}^{n}$ as word sets of all candidate question subjects, all candidate question bodies and all candidate answers, respectively. Here, m, g and n are the length of each question subject, each question body and each answer, respectively. Through this layer, each candidate question-subject, question-body and answer are converted into vectors $\{e_t^{subject}\}_{t=1}^{m}$, $\{e_t^{body}\}_{t=1}^{g}$ and $\{e_t^{answer}\}_{t=1}^{n}$, respectively.

3.4 Encoder

We use bidirectional LSTM (Bi-LSTM) encoders to convert question-subjects, question-bodies, and answers into coded form based on temporal dependency. A H-dimensional contextual representation for each word is obtained by concatenating outputs of two layers whose directions are opposite. For a question-subject, the input of Bi-LSTM is the embedding of a question-subject, denoted as $\{e_t^{subject}\}_{t=1}^{m}$, and the outputs are returns from the Bi-LSTM, denoted as $U^{subject} = \{u_t^{subject}\}_{t=1}^{m} \in R^{m \times H}$. Therefore, the encoded-question-subject is calculated as follows:

$$u_t^{subject} = BiLSTM_{subject}(u_{t-1}^{subject}, e_t^{subject}) \tag{1}$$

Similarly, we get the encoded-question-body and the encoded-answer, respectively.

$$u_t^{body} = BiLSTM_{body}(u_{t-1}^{body}, e_t^{body}) \tag{2}$$

$$u_t^{answer} = BiLSTM_{answer}(u_{t-1}^{answer}, e_t^{answer}) \tag{3}$$

3.5 Self-attention

In this layer, we use a self-attention mechanism proposed in [14] to convert question-subjects, question-bodies and answers of different lengths into fix-length vectors. Since the significance of a certain word varies in a question (or an answer) and between questions (or answers) the self-attention technique will assign different weights to the certain word according to where it occurs. Let $A^{subject}$, A^{body} and A^{answer} be self-attention question-subject representation set, self-attention question-body representation set and self-attention answer representation set, respectively. Given a specific feature of a question-subject representation set $U^{subject} = \{u_t^{subject}\}_{t=1}^{m}$ as the input, a self-attention question-subject representation set $A^{subject} = \{a_t^{subject}\}_{t=1}^{m}$ is generated by this layer. The details are illustrated in the following Equations:

$$c_t^{subject} = w_{subject}^{T} \cdot (tanh(W_{subject} \cdot u_t^{subject})) \tag{4}$$

$$\alpha_t^{subject} = \frac{\exp(c_t^{subject})}{\sum_{j=1}^{m} \exp(c_j^{subject})} \tag{5}$$

Similarly, we get self-attention question-body representation set and self-attention answer representation set as follows.

$$\alpha_t^{body} = \frac{\exp(c_t^{body})}{\sum_{j=1}^{g} \exp(c_j^{body})} \tag{6}$$

$$\alpha_t^{answer} = \frac{\exp(c_t^{answer})}{\sum_{j=1}^{n} \exp(c_j^{answer})} \tag{7}$$

3.6 Cross Attention

The Cross Attention layer, which is proposed in [15,16], aims at fusing words information in question-subjects with words information in answers. The cross attention mechanism used in this research has been proven to be a vital composition of the best model for reading comprehension task. Specifically, this Cross Attention layer is to compute the relevance between each word in question-subjects and each word in answers by bidirectional attention mechanism, including *Subject*2*Answer*, the attention mechanism from question-subjects to answers, and *Answer*2*Subject*,the attention mechanism from answers to question-subjects. By computing the similarity between question-subjects and answers, we get a matrix denoted as $SAS \in R^{m \times n}$. Then two similarity matrices, $\bar{S}_{Subject2Answer} \in R^{m \times n}$ and $\bar{S}_{Answer2Subject} \in R^{m \times n}$, are generated after normalization over each row and column by softmax function.

Let $s_{x,y} \in R$ be an element in similarity matrix $SAS \in R^{m \times n}$, where rows represent question-subjects and columns represent answers. Given $A^{subject}$ and A^{answer} as inputs, we get two cross-attention matrices $A_{Subject2Answer} \in R^{m \times H}$and $A_{Answer2Subject} \in R^{m \times H}$ as final outputs. The computation process can be described as the following equations:

$$s_{x,y} = w'^{T}_{subject} \cdot [a_x^{subject}; a_y^{answer}; a_x^{subject} \odot a_y^{answer}] \tag{8}$$

$$\bar{S}_{Subject2Answer} = softmax_{row}(SAS) \tag{9}$$

$$\bar{S}_{Answer2Subject} = softmax_{col}(SAS) \tag{10}$$

$$A_{Subject2Answer} = \bar{S}_{Subject2Answer} \cdot A^{answer} \tag{11}$$

$$A_{Answer2Subject} = \bar{S}_{Subject2Answer} \cdot \bar{S}^{T}_{Answer2Subject} \cdot A^{subject} \tag{12}$$

where $w'^{T}_{subject}$ is a parameter.

We can similarly obtain $A_{Body2Answer}$ and $A_{Answer2Body}$ as aboves.

3.7 Adaptive Co-attention

Inspired by previous works [18], we use adaptive co-attention, including question-guided co-attention and answer-guided co-attention to capture interaction between questions and their corresponding answers. Then we propose to use a gated fusion module to adaptively fuse features. In addition, a filtration gate is adopted to filter out useless information of question bodies so as to

reduce the noise in question-answer fused features. Through cross attention layer, $A_{Subject2Answer_i}$ represents the i-th cross-attention word in a question subject while $A_{Body2Answer}$ is a cross-attention question body matrix. A single-layer neural network with a softmax function are used to generate attention distribution of question subjects over their question bodies.

$$z_i = tanh(W_{A_{B2A}} \cdot A_{Body2Answer} \oplus (W_{A_{S2A_i}} \cdot A_{Subject2Answer_i} + b_{A_{S2A_i}})) \quad (13)$$

$$\alpha_i = softmax(W_{\alpha_i} \cdot z_i + b_{\alpha_i}) \quad (14)$$

Here, $W_{A_{B2A}}$,$W_{A_{S2A_i}}$,W_{α_i},$b_{A_{S2A_i}}$ and b_{α_i} are parameters.

We use $\oplus$ to represent the concatenation of a question-body feature matrix and a question-subject feature vector. The concatenation between a matrix and a vector is achieved by connecting each column of the matrix to the vector.

α_i is the attention distribution, namely, the attention weight of each word in a question body. The i-th word in a question body related to the question subject can be computed by the following operation.

$$A'_{Body2Answer_i} = A_{Body2Answer} \cdot \alpha_i \quad (15)$$

Next, we use the new word vector of a question body $A'_{Body2Answer_i}$ to obtain the attention matrix of the question body.

$$\gamma_i = tanh(W_{A_{S2A}} \cdot A_{Subject2Answer} \oplus (W_{A_{B2A_i}} \cdot A'_{Body2Answer_i} + b_{A_{B2A_i}})) \quad (16)$$

$$\beta_i = softmax(W_{\beta_i} \cdot \gamma_i + b_{\beta_i}) \quad (17)$$

Then, we obtain a new representation of the question subject related to the question body.

$$A'_{Subject2Answer_i} = A_{Subject2Answer} \cdot \beta_i \quad (18)$$

Afterwards, we propose to use a gated fusion module to fuse the new question subject with the new question body. Details are demonstrated as follows:

$$A''_{Body2Answer_i} = tanh(W_{B2A_i} \cdot A'_{Body2Answer_i} + b_{B2A_i}) \quad (19)$$

$$A''_{Subject2Answer_i} = tanh(W_{S2A_i} \cdot A'_{Subject2Answer_i} + b_{S2A_i}) \quad (20)$$

$$g_i = \sigma(W_{g_i} \cdot (A''_{Subject2Answer_i} \oplus A''_{Body2Answer_i}) \quad (21)$$

$$v_i = g_i \cdot A''_{Body2Answer_i} + (1 - g_i) \cdot A''_{Subject2Answer_i} \quad (22)$$

where W_{B2A_i}, W_{S2A_i}, b_{B2A_i} and b_{S2A_i} are parameters, σ is the logistic sigmoid activation, g_i is the gate fusion model applied to the new question-subject vector $A''_{Subject2Answer_i}$ and the new question-body vector $A''_{Body2Answer_i}$, and v_i is the fusion features incorporating question-subject information and question-answer information.

Since the fusion features v_i originates from two kinds of information which may cause extra noise, we use a filtration gate to combine fusion features with

original features. The filtration gate is a scalar in the range of $[0, 1]$. When the fusion feature is helpful to improve the model performance, the filtration gate is 1; otherwise, the value of the filtration gate is set to 0. The filtration gate s_i and the question-information-enhanced features of new questions $question\_new_i$ are defined as follows:

$$s_i = \sigma(W_{A_{S2A_i}} \cdot A_{Subject2Answer_i} \oplus W_{A_{B2A_i}} \cdot A_{Body2Answer_i} \oplus (W_{v_i,s_i,b_i} \cdot v_i + b_{v_i,s_i,b_i})) \tag{23}$$

$$u_i = s_i \cdot tanh(W_{v_i} \cdot v_i + b_{v_i}) \tag{24}$$

$$question\_new_i = w_{question\_new_i} \cdot (A_{Subject2Answer_i} \oplus A_{Body2Answer_i} \oplus u_i) \tag{25}$$

where W_{v_i,s_i,b_i}, W_{v_i}, b_{v_i,s_i,b_i} and b_{v_i} are parameters, and u_i is the filtered fusion features.

We can similarly derive the answer-information-enhanced features of new answers $answer\_new_i$ as aboves.

3.8 Interaction and Prediction Layer

Inspired by previous works [19,20], we combine the new representation of question subjects with the new representation of question bodies to further enhance local semantic information. To be specific:

$$Q_i^m = [u_i^{subject}; u_i^{body}; question\_new_i; u_i^{subject} + u_i^{body} - question\_new_i; u_i^{subject} \odot u_i^{body} \odot question\_new_i] \tag{26}$$

$$A_j^n = [u_j^{answer}; answer\_new_j; u_j^{answer} - answer\_new_j; u_j^{answer} \odot answer\_new_j] \tag{27}$$

where $[\cdot;\cdot;\cdot;\cdot]$ is the concatenation operation of vectors, and $\odot$ indicates element-wise product.

Next, we train our model with bidirectional GRU (Bi-GRU) to acquire context information between questions and their corresponding answers. The detailed acquisition methods are shown as follows.

$$Q_i^v = BiGRU(Q_i^m, Q_{i-1}^v, Q_{i+1}^v) \tag{28}$$

$$A_j^v = BiGRU(A_j^n, A_{j-1}^v, A_{j+1}^v) \tag{29}$$

After that, we use max pooling and mean pooling of $Q^v = (Q_1^v, Q_2^v, \cdots, Q_m^v)$ and $A^v = (A_1^v, A_2^v, \cdots, A_n^v)$ to acquire fixed-length vectors. Specifically, we compute max pooling vectors r_Q^{max} and r_A^{max} as well as mean pooling vectors r_Q^{mean} and r_A^{mean}. Then, we concatenate these vectors to get the global representation r. The details are indicated as follows.

$$r_Q^{mean} = \Sigma_{i=1}^m \frac{Q_i^v}{m} \tag{30}$$

$$r_Q^{max} = max_{i=1}^m Q_i^v \tag{31}$$

$$r_A^{mean} = \Sigma_{i=1}^{n} \frac{A_j^v}{n} \tag{32}$$

$$r_A^{max} = max_{i=1}^{n} A_j^v \tag{33}$$

$$r = [r_Q^{mean}; r_Q^{max}; r_A^{mean}; r_A^{max}] \tag{34}$$

Finally, we pass the global representation r to the prediction layer which is consisted of a multi-layer perceptron (MLP) classifier to determine whether the semantic meaning of the give question-answer pair is equivalent or not.

$$\nu = tanh(W_r \cdot r + b_r) \tag{35}$$

$$\hat{y} = softmax(W_\nu \cdot \nu + b_\nu) \tag{36}$$

Here, W_r, b_r, W_ν, and b_ν are trainable parameters. The entire model is trained end-to-end.

4 Experimental Setup

4.1 DataSet

The two corpora using to train and evaluate our model are CQA datasets SemEval2015 and SemEval2017, both containing two parts, questions and their corresponding answers. Each question comprises a brief title and informative descriptions. Detailed statistics of two corpora are shown in Table 2 and Table 3.

Table 2. Statistical information of SemEval2015 corpus

	Train	Dev	Test
Number of questions	2376	266	300
Number of answers	15013	1447	1793
Average length of subject	6.36	6.08	6.24
Average length of body	39.26	39.47	39.53
Average length of answer	35.82	33.90	37.33

4.2 Training and Hyper Parameters

For the text preprocessing procedure, we exert NLTK toolkit over each question and its corresponding answers, including capitalization conversion, stemming, removal of stop words, etc. After preprocessing, we train two datasets with GloVe proposed by Pennington [11] to obtain 300-dimensional initialized word vectors and set the number of out-of-vocabulary(OOV) words to zero. Adam Optimizer is chosen for optimization with the first momentum coefficient β_1 0.9 and the second momentum coefficient β_2 0.999. In addition, initial learning rate is set to $[1 \times 10^{-4}, 4 \times 10^{-6}, 1 \times 10^{-5}]$, L2 regularization value is set to $[1 \times 10^{-5}, 4 \times 10^{-7}, 1 \times 10^{-6}]$, and batch-size is set to [64, 128, 256]. We select parameters which perform best on validation sets and evaluate model performance on test sets.

Table 3. Statistical information of SemEval2017 corpus

	Train	Dev	Test
Number of questions	5124	327	293
Number of answers	38638	3270	2930
Average length of subject	6.38	6.16	5.76
Average length of body	43.01	47.98	54.06
Average length of answer	37.67	37.30	39.50

4.3 Results and Analysis

Results and Analysis on SemEval 2015 Dataset. We adopt two evaluation metrics, F1 and Acc (accuracy) to compare performance of AMQAN with other seven answer selection models on SemEval 2015 dataset as shown in Table 4 and Table 5.

As shown in Table 5, not all deep learning models can outperform conventional machine learning models. For example, machine learning model JAIST increase by 1.73% in F1 and 0.7% in Acc than deep learning model BGMN. Then, Graph-cut and FCCRF outperform BGMN, JAIST, and HITSZ-ICRC on F1 and Acc, which prove that feature-engineering-based model with effective features sometimes have better performance in the answer selection task.

Most importantly, the experiment results show that our model AMQAN has the best performance on this dataset, outperforming the current state-of-the-art model (7) by 0.48% in F1 and 1.28% in Acc ($p < 0.05$ on student t-test), which can be mainly attributed to multi-attention mechanisms. Specifically, self-attention focuses on modeling temporal interaction in long sentences, finding the most relevant words in answers to their corresponding questions. With cross attention between question subjects and answers, question bodies and answers, AMQAN can successfully identify words that represent the correlation between questions and answers. In adaptive co-attention module, question-driven attention and answer-driven attention are interacted to precisely capture the semantic meaning between questions and answers.

Results and Analysis on SemEval 2017 Dataset. We also adopt three evaluation metrics, F1, Acc (accuracy) and MAP (Mean Average Precision) to compare performance of AMQAN with other eight answer selection models, including two embeddings on SemEval 2017 dataset as shown in Table 6 and Table 7.

From Table 7, we can also find deep learning models not always achieve unsurpassable results. For example, model (1), (2) outperform two deep learning models (4) and (5) in MAP. Next, model (5) and (6) which treat question subjects and question bodies separately outperform model (4) which takes question subjects and question bodies as an entirety because mechanical connection of question-subjects and question-bodies introduces too much noise, resulting in

Table 4. Descriptions of answer selection models on SemEval 2015 dataset

Model	Reference and description
(1) JAIST [21]	Use SVM to incorporate various kinds of features, including topic model based features and word vector features
(2) HITSZ-ICRC [22]	Propose ensemble learning and hierarchical classification to classify answers
(3) Graph-cut [23]	Model the relationship between answers in the same question thread, based on the idea that similar answers should have similar labels
(4) FCCRF [4]	Apply local learned classifiers and fully connected CRF to make global inference so as to predict the label of each individual node more precisely
(5) BGMN [24]	Use the memory mechanism to iteratively aggregate more relevant information in order to identify the relationship between questions and answers
(6) CNN-LSTM-CRF [25]	Propose multilingual hierarchical attention networks for learning document structures
(7) QCN [1]	A Question Condensing Network focusing on the similarity and disparities between question-subjects and question-bodies for answer selection in Community Question Answering, which is the state-of-the-art model for Answer Selection
(8) AMQAN (ours)	An adaptive multi-attention question-answer network, outperforming all current answer selection models

Table 5. Comparisons on the SemEval 2015 dataset

Model	F1	Acc
(1) JAIST	0.7896	0.7910
(2) HITSZ-ICRC	0.7652	0.7611
(3) Graph-cut	0.8055	0.7980
(4) FCCRF	0.8150	0.8050
(5) BGMN	0.7723	0.7840
(6) CNN-LSTM-CRF	0.8222	0.8224
(7) QCN	0.8391	0.8224
(8) AMQAN (ours)	**0.8439**	**0.8352**

the performance degradation. Then the comparison between model (7) and (8) indicates that using task-specific embeddings and character embeddings both contribute to model performance, especially help attenuate OOV [29] problems

Table 6. Descriptions of answer selection models on SemEval 2015 dataset

Model	Reference and description
(1) KeLP [26]	Use syntactic tree kernels with relational links between questions and answers, and apply standard text similarity measures linearly combined with the tree kernel
(2) Beihang-MSRA [27]	Use gradient boosted regression trees to combine traditional linguistic features and neural network-based matching features
(3) ECUN [28]	Use traditional features and a convolutional neural network to represent question-answer pairs
(4) LSTM	A LSTM-based model applied on question subjects and question bodies respectively, and concatenate these two results to obtain final representation of questions
(5) LSTM-subject-body	Use the memory mechanism to iteratively aggregate more relevant information in order to identify the relationship between questions and answers
(6) QCN [1]	A Question Condensing Network focusing on the similarity and disparities between question-subjects and question-bodies for answer selection in CQA, which is the state-of-the-art model for Answer Selection
(7) w/o task-specific word embeddings	Word embeddings are initialized with the 300-dimensional GloVe trained on Wikipedia 2014 and Gigaword 5
(8) w/o character embeddings	Word embeddings are initialized with the 600-dimensional GloVe trained on the domain-specific unannotated corpus
(9) AMQAN (ours)	An adaptive multi-attention question-answer network, outperforming all current answer selection models

Table 7. Comparisons on the SemEval 2017 dataset

Model	MAP	F1	Acc
(1) KeLP	0.8843	0.6987	0.7389
(2) Beihang-MSRA	0.8824	0.6840	0.5198
(3) ECNU	0.8672	0.7767	0.7843
(4) LSTM	0.8632	0.7441	0.7569
(5) LSTM-subject-body	0.8711	0.7450	0.7728
(6) QCN	0.8851	0.7811	0.8071
(7) w/o task-specific word embeddings	0.8896	0.7832	0.8085
(8) w/o character embeddings	0.8756	0.7768	0.7978
(9) AMQAN (ours)	**0.8925**	**0.7868**	**0.8132**

since CQA text is non-standard with abbreviations, typos, emoticons, and grammatical mistakes, etc.

The most important is our proposed model AMQAN increase by 0.74% in MAP, 0.57% in F1, and 0.61% in Acc ($p < 0.05$ on student t-test) compared with the best model (6), for the reason that AMQAN studies the relationship of question-bodies, question-subjects, and answers, fully using self-attention, cross-attention and adaptive co-attention, so that it greatly enhances the performance of the answer selection task.

5 Conclusion

In this study, we propose AMQAN, an adaptive multi-attention question-answer networks for answer selection, which take the relationship between questions and answers as important information for answer selection. In order to effectively answer both factoid and non-factoid questions with various length, our model AMQAN applies deep attention mechanism at word, sentence, and document level, utilizing characteristics of linguistic knowledge to explore the complex relationship between different components. Further, we use multi-attention mechanism like self-attention, cross attention and adaptive co-attention to comprehensively capture more interactive information between questions and answers. In general, AMQAN outperforms all baseline models, achieving significantly better performance than all current state-of-the-art answer selection methods. In the future, our research group will mainly focus on improving the computing speed of AMQAN to further level up the performance of our solution.

References

1. Wu, W., Sun, X., Wang, H., et al.: Question condensing networks for answer selection in community question answering. In: Meeting of the Association for Computational Linguistics, pp. 1746–1755 (2018)
2. Yao, X., Van Durme, B., Callisonburch, C., et al.: Answer extraction as sequence tagging with tree edit distance. In: North American Chapter of the Association for Computational Linguistics, pp. 858–867 (2013)
3. Tran, Q.H., Tran, D.V., Vu, T., Le Nguyen, M., Pham, S.B.: JAIST: Combining multiple features for Answer Selection in Community Question Answering. In: North American Chapter of the Association for Computational Linguistics, pp. 215–219 (2015)
4. Joty, S., Marquez, L., Nakov, P.: Joint learning with global inference for comment classification in community question answering. In: North American Chapter of the Association for Computational Linguistics, pp. 703–713 (2016)
5. Nakov, P., Marquez, L., Moschitti, A., et al. SemEval-2016 task 3: community question answering. In: North American Chapter of the Association for Computational Linguistics, pp. 525–545 (2016)
6. Filice, S., Croce, D., Moschitti, A., et al.: KeLP at SemEval-2016 task 3: Learning semantic relations between questions and answers. In: North American Chapter of the Association for Computational Linguistics, pp. 1116–1123 (2016)

7. Wang, M., Smith, N.A., Mitamura, T.: What is the jeopardy model? a quasisynchronous grammar for QA. In: EMNLP-CoNLL (2007)
8. Heilman, M., Smith, N.A.: Tree edit models for recognizing textual etailments, paraphrases, and answers to questions. In: Proceedings of NAACL (2010)
9. Murdock, V., Croft, W.B.: Simple translation models for sentence retrieval in factoid question answering. In: Proceedings of the SIGIR-2004 Workshop on Information Retrieval For Question Answering (IR4QA), pp. 31–35 (2004)
10. Zhou, G., Cai, L., Zhao, J., et al.: Phrase-based translation model for question retrieval in community question answer archives. In: Meeting of the Association for Computational Linguistics, pp. 653–662 (2011)
11. Pennington, J., Socher, R., Manning, C. D., et al.: Glove: global vectors for word representation. In: Empirical Methods in Natural Language Processing, pp. 1532–1543 (2014)
12. Kim, Y., Jernite, Y., Sontag, D., et al.: Character-aware neural language models. In: National Conference on Artificial Intelligence pp. 2741–2749 (2016)
13. Chen, Q., Zhu, X., Ling, Z., et al.: Neural natural language inference models enhanced with external knowledge. In: Meeting of the Association for Computational Linguistics, pp. 2406–2417 (2018)
14. Lin, Z.: A structured self-attentive sentence embedding. In: International Conference on Learning Representations (ICLR) (2017)
15. Weissenborn, D., Wiese, G., Seiffe, L., et al.: Making neural QA as simple as possible but not simpler. In: Conference on Computational Natural Language Learning, pp. 271–280 (2017)
16. Yu, A.W.: QANet: combining local convolution with global self-attention for reading comprehension. In: International Conference on Learning Representations (ICLR) (2018)
17. Seo, M., Kembhavi, A., Farhadi, A., Hajishirzi, H.: Bidirectional attention flow for machine comprehension. In: International Conference on Learning Representations (ICLR) (2017)
18. Lu, J., Xiong, C., Parikh, D., et al.: Knowing when to look: adaptive attention via a visual sentinel for image captioning. In: Computer Vision and Pattern Recognition, pp. 3242–3250 (2017)
19. Chen, Q., Zhu, X., Ling, Z., et al.: Enhanced LSTM for natural language inference. In: Meeting of the Association for Computational Linguistics, pp. 1657–1668 (2017)
20. Mou, L., Men, R., Li, G., et al.: Natural language inference by tree-based convolution and heuristic matching. In: Meeting of the Association for Computational Linguistics, pp. 130–136 (2016)
21. Tran, Q.H., Tran, V.D., Vu,T.T., et al.: JAIST: Combining multiple features for answer selection in community question answering. In: North American Chapter of the Association for Computational Linguistics, pp. 215–219 (2015)
22. Hou, Y., Tan, C., Wang, X., et al. HITSZ-ICRC: exploiting classification approach for answer selection in community question answering. In: North American Chapter of the Association for Computational Linguistics, pp. 196–202 (2015)
23. Joty, S.R., Barroncedeno, A., Martino, G.D., et al.: Global thread-level inference for comment classification in community question answering. In: Empirical Methods in Natural Language Processing, pp. 573–578 (2015)
24. Wei, W., Houfeng, W., Sujian, L.: Bidirectional gated memory networks for answer selection. In: Chinese Computational Linguistics and Natural Language Processing Based on Naturally Annotated Big Data, LNAI 10565, Springer, Cham pp. 251–262 (2017)

25. Xiang, Y., Zhou, X., Chen, Q., et al.: Incorporating label dependency for answer quality tagging in community question answering via CNN-LSTM-CRF. In: International conference on computational linguistics, pp. 1231–1241 (2016)
26. Filice, S., Martino, G.D., Moschitti, A., et al.: KeLP at SemEval-2017 task 3: learning pairwise patterns in community question answering. In: Meeting of the Association for Computational Linguistics, pp. 326–333 (2017)
27. Feng, W., Wu, Y., Wu, W., et al.: Beihang-MSRA at semEval-2017 task 3: a ranking system with neural matching features for community question answering. In: Meeting of the Association for Computational Linguistics, pp. 280–286 (2017)
28. Wu, G., Sheng, Y., Lan, M., et al.: ECNU at semEval-2017 task 3: using traditional and deep learning methods to address community question answering task. In: Meeting of the Association for Computational Linguistics, pp. 365–369 (2017)
29. Zhang, X., Li, S., Sha, L., et al.: Attentive interactive neural networks for answer selection in community question answering. In: National Conference on Artificial Intelligence, pp. 3525–3531 (2017)