DGQAN: Dual Graph Question-Answer Attention Networks for Answer Selection

Haitian Yang
Institute of Information Engineering, Chinese Academy of Sciences
School of Cyber Security, University of Chinese Academy of Sciences
yanghaitian@iie.ac.cn

Xuan Zhao
Department of Electrical Engineering Computer Science, Lassonde School of Engineering,
York University
xuanzhao@eecs.yorku.ca

Yan Wang★
Institute of Information Engineering, Chinese Academy of Sciences
School of Cyber Security, University of Chinese Academy of Sciences
wangyan@iie.ac.cn

Min Li
Institute of Information Engineering, Chinese Academy of Sciences
School of Cyber Security, University of Chinese Academy of Sciences
limin@iie.ac.cn

Wei Chen
School of Information Management and Engineering
Shanghai University of Finance and Economics
justcw@126.com

Weiqing Huang
Institute of Information Engineering, Chinese Academy of Sciences
School of Cyber Security, University of Chinese Academy of Sciences
huangweiqing@iie.ac.cn

ABSTRACT

Community question answering (CQA) becomes increasingly prevalent in recent years, providing platforms for users with various backgrounds to obtain information and share knowledge. However, the redundancy and lengthiness issues of crowd-sourced answers limit the performance of answer selection, thus leading to difficulties in reading or even misunderstandings for community users. To solve these problems, we propose the dual graph question-answer attention networks (DGQAN) for answer selection task. Aims to fully understand the internal structure of the question and the corresponding answer, firstly, we construct a dual-CQA concept graph with graph convolution networks using the original question and answer text. Specifically, our CQA concept graph exploits the correlation information between question-answer pairs to construct two sub-graphs (QSubject-Answer and QBody-Answer), respectively. Further, a novel dual attention mechanism is incorporated to model both the internal and external semantic relations among questions and answers. More importantly, we conduct experiment to investigate the impact of each layer in the BERT model. The experimental results show that DGQAN model achieves state-of-the-art performance on three datasets (SemEval-2015, 2016, and 2017), outperforming all the baseline models.

CCS CONCEPTS

• **Information systems → Question answering**.

KEYWORDS

Dual Graph Attention, Answer Selection, Community Question Answering

★Yan Wang is the corresponding author.

SIGIR '22, July 11–15, 2022, Madrid, Spain.

ACM ISBN 978-1-4503-8732-3/22/07.
https://doi.org/10.1145/3477495.3532084

ACM Reference Format:
Haitian Yang, Xuan Zhao, Yan Wang★, Min Li, Wei Chen, and Weiqing Huang. 2022. DGQAN: Dual Graph Question-Answer Attention Networks for Answer Selection. In *Proceedings of the 45th Int'l ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '22), July 11–15, 2022, Madrid, Spain.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3477495.3532084

1 INTRODUCTION

Community Question Answering (CQA) [1–4] has become popular in recent years, which provides not only an interactive experience but also a fast retrieve in many fields. The typical Question Answering platforms, such as Zhihu, StackOverflow, Quora, etc., are fairly open so that users can search for the information they are interested in, post questions of their own concerns, and provide answers to topics that they are eager to share. Due to the various backgrounds of users, answers from different users show a huge variance in quality. Specifically, some answers are related to the topic, some answers are completely desirable, while others may completely deviate from the actual intention of users. Therefore, it is time-consuming for users to view all candidate answers and select the most relevant one. Answer selection [5–9] aims to find the most relevant answer in the repository and reduce users' time going through all candidate answers. Answer selection can provide a better experience for CQA users, and more importantly, can solve some unsettled but similar questions with different representations, which actually have been answered before.

A typical CQA example is shown in Table 1. Answer 1 is a good answer because it gives informative details, such as "check it to the traffic dept", which is very useful to solve the question. Answer 2 is a bad answer, because it does not provide much useful information, even it is not completely unrelated to the question.

The example demonstrates a common characteristic of questions in CQA, i.e., a question often consists of two parts - *question subject* and *question body*. Indeed, not every question in CQA contains these two parts. In this study, we focus on questions with both question subject and question body. Question subject is a summary of a question, often containing and emphasizing the important

Table 1: An example question and its related answers.

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 accident/s 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 :)

words. While question body is a detailed description of the question, including both the critical information and some extended information (illustrated in Table 1). It is worth noting that for both question and answer, there is a vast amount of redundancy, which deteriorates the performance of answer selection solutions.

The answer selection task is challenging mainly due to two reasons. First of all, there are many low-informative words (for example, auxiliary verbs), a large number of synonyms, and syntactic transformation, which make the answer selection task very difficult. For instance, the questions "What are better ways to look for nice local restaurants?" and "How do you search for great restaurants along your route?" both mean seeking good places to eat food, but words and syntactic representations in these two samples are not identical at all. Secondly, many researchers focus on treating questions and answers in the same way, ignoring the redundancy and noise in answers [10]. Often only a part of the content can provide useful information, inevitably resulting in severe deviation.

To overcome these difficulties, we propose DGQAN – Dual Graph Question & Answer Attention Networks, which investigate the interactions among question subjects, question bodies, and answers, adopting these important information for answer selection in CQA. Firstly, we introduce our proposed CQA Concept Graph, constructing QSubject-Answer Graph and QBody-Answer Graph separately. The CQA Concept Graph ensures that better representations illustrating the interactions of question-subject, question-body, and answer are obtained. Secondly, we apply the QSubject-Answer attention and QBodyAnswer attention mechanism to extract multiple interactive features. Finally, we integrate our constructed graphs and extracted features under our fusion framework to achieve answer selection.

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

- We present the CQA Concept Graph, consisting of two parts: QSubject-Answer Graph and QBody-Answer Graph. CQA Concept Graph provides richer representations of CQA text. To the best of our knowledge, this is the first work to explore a solution of answer selection task via multi-graph networks.
- We use QSubject-Answer and QBody-Answer attentions simultaneously to capture more interactive features. Further, we fuse the constructed graphs and extracted features under our dual attention framework to obtain useful information both locally and globally.
- We explore the effect of different BERT layers (including one, multiple, and full layers of Transformer) on public dataset, regarding to the graph neural network and the dual attention components.
- We use three representative datasets, SemEval-2015, SemEval-2016, and SemEval-2017 to evaluate our proposed model. The results show that our DGQAN achieves excellent performance on these datasets, outperforming all baseline models.

2 RELATED WORK

2.1 Answer selection

The answer selection task in the early years highly relies on feature engineering, linguistics tools and other external resource. Nakov et al. [11] studied a wide range of feature types, including similarity feature, content feature, meta-feature, and other features which are automatically extracted from the SemEval CQA model. Tran et al. [12] applied the topic-model-based features and word-embedding-based features for the answer ranking task. Filice et al. [13] designed various heuristic features and thread-based features, which can also provide better selections.

In recent years, deep learning models (for example, Yang et al. [14], Deng et al. [5]) are proved to be a relatively more effective method. Deep learning methods can automatically capture various features through multi-layer networks instead, avoiding complex feature engineering. Shen et al. [15] used translation matrices to learn word representations, then calculated the relevance score of question-answer pairs in the repository. Tay et al. [16] proposed temporal gates, learning the representations of sequence pairs, which take interactive information of question-answer pairs into consideration. Wu et al. [17] proposed question condensing networks using the relationship between questions and answers to align question-answer pairs. Zhou et al. [18] introduced a recurrent convolutional neural network which combines the characteristics of the recurrent neural network and the convolutional neural network to capture the semantic correlations within each answer.

Although these deep learning models have achieved better performance, none of these explored question subjects, question bodies, and answers, respectively. In order to further improve the performance of the answer selection task, we propose an approach that utilizes dual graph question-answer attention networks (DGQAN) for Answer Selection. Specifically, we first organize the original question and answer text into a CQA concept graph with graph

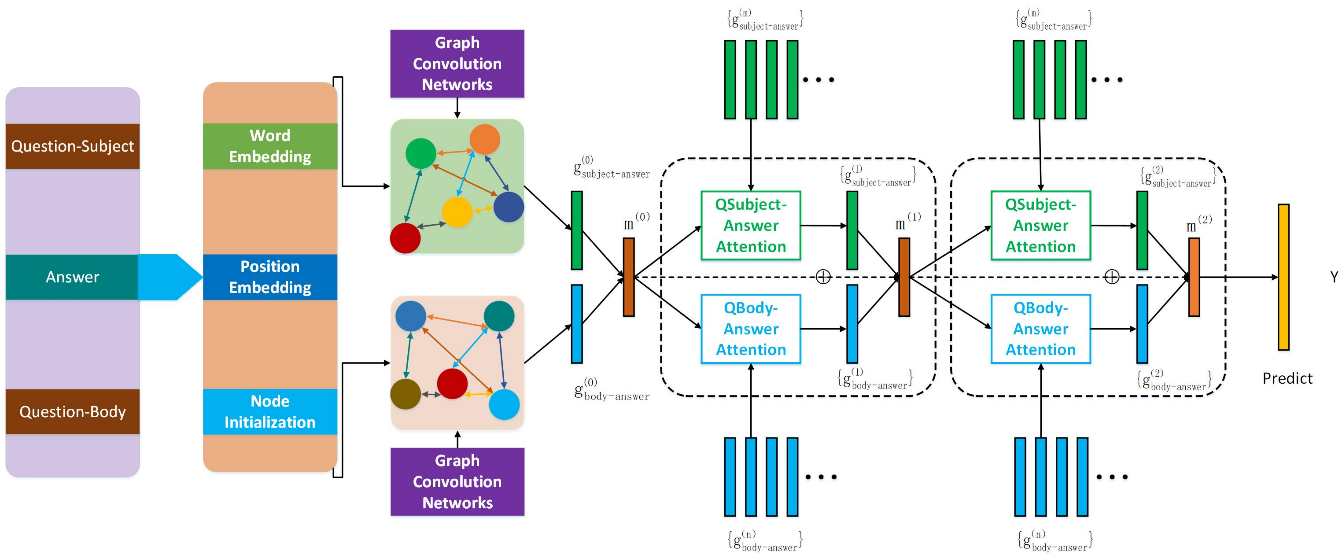

Figure 1: Overall structure of Our Proposed Model DGQAN

convolution networks to better understand the internal structure of the question and the correlation answer, which exploits the correlation information between question-answer pairs to bulid the QSubjetc-Answer Graph and the QBody-Answer Graph separately. Further, a novel dual attentive mechanism is incorporated to model the semantic internal and external relations among questions as well as answers.

2.2 Answer Ranking

Answer ranking [19] task in retrieval-based question answering systems can be solved using answer selection approaches. Especially with the breakthroughs brought by deep learning technology, answer selection schemes are more and more used to solve the task of answer ranking. Some methods apply attention architectures [20] or pre-trained models [21], such as ElMo [22] or BERT [23] to model semantic correlations. Some papers explore bringing in different sources to enhance performance while alleviating redundancy [6]. These methods have contributed many classic solutions to CQA, but they do not leverage the real-world background knowledge into the task.

2.3 Graph Neural Network

Some recent studies utilize Graph Neural Network (GNN) in natural language processing (NLP) tasks, constructing text-based graphs to model structural relationships. Compared to other deep learning techniques, GNN shows incomparable advances - high interpretability and significant performance. Wang et al. [24] proposed a method for cross-lingual knowledge graph alignment via Graph Convolutional Networks (GCN). Their method trains the GCN to embed entities for each language into a unified vector space. Combining graph-based ranking methods and neural network-based methods, Sun et al. [25] presented a GCN encoder for diverse key phrases that can effectively capture document-level word salience. In this paper, our motivation of using graph neural networks is that graph-based structures can better represent the correlations of different concepts in question-answer pairs and capture deep semantic information of the entire text.

3 THE PROPOSED MODEL

3.1 Task Description

The answer selection task of community question answering can be described as a tuple of four elements (S, B, A, y), where $\mathrm{S} = \left[s^1, s^2, \ldots, s^g\right]$ represents the question subject whose length is g, $\mathrm{B} = \left[b^1, b^2, \ldots, b^m\right]$ stands for the question body whose length is m, $\mathrm{A} = \left[a^1, a^2, \ldots, a^n\right]$ denotes the answer corresponding to a question whose length is n, and $\mathrm{y} \in \mathrm{Y}$ conveys the relevance degree of a question-answer pair. More detailed, Y = {Good, Bad}, which *Good* represents that the answer can provide a definitely relevant solution for the given question and "*Bad*" means the answer is irrelevant to the question or useless to users. Generally, the function of our model DGQAN is to assign a label to each answer based on the conditional probability $\Pr(y \mid S, B, A)$ according to the given set $\{\mathrm{S}, \mathrm{B}, \mathrm{A}\}$ to solve the answer selection task in Community Question Answering.

3.2 Overview of Proposed Model

The structure of our approach is shown in Figure 1. First, we explain the construction of our CQA Concept Graph, node Initialization, and the overall of our Graph Convolution Networks; next, we describe the QSubject-Answer attention and QBody-Answer attention mechanism, which are applied to extract multiple features; finally, we fuse our constructed graphs and extracted attentions through our fusion framework to achieve answer selection.

3.3 Question Answer Graph Construction

We describe the construction of our CQA Concept Graph, which consists of two parts, QSubject-Answer Graph and QBody-Answer

Algorithm 1 CQA Concept Graph

Require: The question subject Text **S** and answer text **A** (similar way to question body), weight calculation function Φ
Combine **S** and clause **A** to get a set of sentences
Segment **S**, and **A** into words
Use keyword detection and named entity recognition to generate concepts Ω
for *sentences* **do**
  if s_i contains $\omega \in \Omega$ **then**
    Assign s_i to vertex v_k
  else
    Assign s_i to vertex v_{empty}
  end if
end for
for vertex v_i and v_j **do**
  Obtain edge weight: $\omega_{i,j} = \Phi(v_i, v_j)$

Graph. We construct our QSubject-Answer Graph based on the relationship between question-subject and answer. Similarly, we apply the process to question-body and answer to construct QBody-Answer Graph. Thus, two different graphs are obtained separately, demonstrated in Algorithm 1.

We define the CQA concepts as follows: phrases/words of question and answer entities; keywords that are vital components of the text. Answers from online platforms often contain a considerable amount of noise, arguably some sentences in the answer are even irrelevant to the question itself.

Given an input question-subject S and an answer A, firstly, we perform word segmentation and named entity recognition (NER) for the text with a pre-trained model BERT-CRF [26]. Secondly, we apply keyword extraction with TextRank [27] to obtain keywords. Thus, we can obtain the concepts of CQA, associating each sentence in the question-subject and answer to its corresponding concepts. Specifically, we assign the sentence to the concept ω, when ω appears in the sentence. A single sentence can be connected with more than one concept, which may implicitly indicate the correlation between concepts. We assign sentences that do not contain any of the concepts with an "empty" vertex. The sentences and the concept consist of the vertex V_k in the question and answer concept graph. We represent each vertex by the concatenation of the concept and sentence words in the question-subject and answer. Further, we construct the edges between vertices through a set of approaches. Whereas, the more sentences are mentioned by two concepts together, the closer these two concepts are. For this reason, we adopt a structure-based method in this paper. Specifically, when vertices V_i and V_j share at least one sentence, we add an edge $E_{(i,j)}$ between them, and its weight is obtained with the number of shared sentences.

3.4 Node Initialization

We encode the vertex in our CQA Concept Graph with vector u_i. First, we use a multi-head self-attention-based vertex encoder, which consists of two modules - the embedding module and the self-attention module. For both words and concepts, we apply the BERT as pre-trained embedding through a sharing embedding lookup table to represent word information. The regular words refer to words other than concept words. We add absolute and relative positional embedding p_i^{absolute}, p_i^{relative} to represent the position information. p_i^{absolute} is utilized to encoder the absolute locations of the words and concepts in the answer. To acquire more accurate relative position embedding, we set the concept in front of the word sequence. Consequently, the relative position embedding of the concept share the same embedding p_0. We add the word embedding w_i, position embedding p_i^{absolute} and p_i^{relative} to get the final embedding u_i, denoted as below:

$$u_i = w_i + p_i^{absolute} + p_i^{\text{relative}} \tag{1}$$

After that, we feed u_i into the self-attention module to obtain the hidden representation a_i of each word. The self-attention can explicitly model the interactions among words to capture the context of the vertex. We calculate the hidden representation of the self-attention layer using equation (2) to equation (4), where Q, k, and V present the query, key, and value vectors, respectively.

$$\text{Attention}(Q, K, V) = \text{softmax}\left(QK^T\right)V \tag{2}$$

$$\text{MultiHead }(Q, K, V) = [\text{ head }_1; \cdots ; \text{ head }_h] W^o \tag{3}$$

$$\text{head}_i = \text{ Attention }\left(QW_i^{Q_s}, QW_i^{K_s}, QW_i^{V_s}\right) \tag{4}$$

The concept ω contains critical information of vertex, hence, we use the representation of the concept a_0 to represent the entire vertex.

3.5 Graph Convolution Networks

We feed the vertex v_i into a graph encoder after obtaining the hidden vectors, which explicitly models the graph structure of the constructed concept graph. We use an implementation of the GCN model following [28]. We denote the adjacency matrix of the interaction graph as $A \in R^{N \times N}$, where $A_{ij} = w_{ij}$, and D is a diagonal matrix.

$$H^{l+1} = \sigma\left(\tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}} H^l W^l\right) \tag{5}$$

$$\tilde{A} = A + I_N \tag{6}$$

where I_N is the identity matrix, $\tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}}$ is the normalized adjacency matrix, and W^l is a learnable weight matrix. We also add residual connections between layers to avoid over-smoothing.

$$g^{l+1} = H^{l+1} + H^l \tag{7}$$

The output of our GCN is g^k. The last input layer of QSubject-Answer Graph is denoted as $g^k_{\text{sub-ans}}$. Similarly, we obtain the last input layer of QBody-Answer Graph, denoted as $g^k_{\text{body-ans}}$.

3.6 Dual Attention Mechanisms

Dual attention mechanisms are the groundwork of our network DGQAN, which captures QSubject-Answer and QBody-Answer attentions simultaneously to achieve multi-features. Here, we explain the details of our networks, depicting the running of underlying attention mechanism step by step. Similarly, QSubject-Answer attention and QBody-Answer attention operate separately. QSubject-Answer attention focuses on key words in each input sentence to obtain context vectors, denoted as:

$$g_{\text{sub-ans}}^{(k)} = \text{SA\_ Att}\left(\left\{g_{\text{sub-ans}}^{t}\right\}_{t=1}^{T},\ \mathrm{m}_{\mathrm{g}}^{(k-1)}\right) \tag{8}$$

where $g_{\text{sub-ans}}^{(k)}$ is the context vector, $\mathrm{m}_{\mathrm{g}}^{(k-1)}$ is the memory vector.

$$\mathrm{h}_{\mathrm{g},t}^{(k)} = \tanh\left(\mathrm{W}_{\mathrm{g}}^{(k)} g_{\text{sub-ans}}^{t}\right) \odot \tanh\left(\mathrm{W}_{\mathrm{g,m}}^{(k)} \mathrm{m}_{\mathrm{g}}^{(k-1)}\right) \tag{9}$$

$$\alpha_{\mathrm{g},t}^{(k)} = \operatorname{softmax}\left(\mathrm{W}_{\mathrm{g,h}}^{(k)} \mathrm{h}_{\mathrm{g},t}^{(k)}\right) \tag{10}$$

$$g_{\text{sub-ans}}^{(k)} = \sum_{t} \alpha_{\mathrm{g},t}^{(k)} g_{\text{sub-ans}}^{t} \tag{11}$$

where $\left\{\alpha_{\mathrm{g},t}^{(k)}\right\}_{t=1}^{T}$ is attention weights, $\mathrm{W}_{\mathrm{g}}^{(k)}$, $\mathrm{W}_{\mathrm{g,m}}^{(k)}$, and $\mathrm{W}_{\mathrm{g,h}}^{(k)}$ are the network parameters, $\mathrm{h}_{\mathrm{g},t}^{(k)}$ is a hidden state. In the similar way, we acquire $g_{\text{body-ans}}^{(k)}$. Attention weights is acquired via 2-layer feed-forward neural network. From the above equations, we also can observe the context vector is the weighted averages of attention weights. Moreover, this textual attention mechanism realizes end-to-end training without additional layers.

3.7 Model Fusion Mechanism

Our fusion framework is demonstrated in Figure 1. This section describes the fusion of $g_{\text{sub-ans}}^{(k)}$ and $g_{\text{body-ans}}^{(k)}$, illustrated in equation (13). $m^{(k)}$ is a fusion memory vector, which stores the information both from the QSubject-Answer Graph and QBody-Answer Graph that has been accumulated until step k. It is recursively updated by

$$\mathrm{m}^{(k)} = \mathrm{m}^{(k-1)} + g_{\text{sub-ans}}^{(k)} \odot g_{\text{body-ans}}^{(k)} \tag{12}$$

where $g_{\text{sub-ans}}^{(k)}$ and $g_{\text{body-ans}}^{(k)}$ are the QSubject-Answer Graph and QBody-Answer Graph context vectors obtained from equations (11) and (8), respectively.

This fusion mechanism ensures that QSubject-Answer Graph and QBody-Answer Graph attention are guided simultaneously, which means that under our framework $m^{(k)} = m_{g_{\text{sub-ans}}}^{(k)} = m_{g_{\text{body-ans}}}^{(k)}$, therefore, two attention mechanisms can achieve close collaboration.

The initial storage vector $m^{(0)}$ is defined based on the global context vectors $g_{\text{sub-ans}}^{(0)}$ and $g_{\text{body-ans}}^{(0)}$:

$$\mathrm{m}^{(0)} = g_{\text{sub-ans}}^{(0)} \odot g_{\text{body-ans}}^{(0)} \tag{13}$$

$$g_{\text{sub-ans}}^{(0)} = \tanh\left(\mathrm{P}^{(0)} \frac{1}{G} \sum_{g} g_{\text{sub-ans}}^{g}\right) \tag{14}$$

$$g_{\text{body-ans}}^{(0)} = \tanh\left(\mathrm{P}^{(0)} \frac{1}{M} \sum_{m} g_{\text{body-ans}}^{m}\right) \tag{15}$$

During the processing, dual-attention (equation (6) and (10)) and memory (equation (14)) are updated k times. Eventually, we predict the detection results using a single-layer softmax classifier with cross-entropy loss, where the input is the final memory $m^{(K)}$:

$$p_i = softmax(W_f \cdot m^{(K)} + b) \tag{16}$$

where $p_i \in R^2$ represents whether the input log sequence is normal or not, and $W_f \in R^{2\times c}$, $b \in R^2$ are parameters to be learned.

3.8 Training Objective

The lost function of this answer selection task is a standard cross-entropy:

$$J(\theta) = -\sum_{i=1}^{N} \left[y_i \cdot \log p_i + (1 - y_i) \cdot \log(1 - p_i)\right] + \lambda \cdot R \tag{17}$$

where N is the size of training dataset, y_i is the true label for QA pair sequence i, $\theta = \{M_{i,j}, U, V, g, W_f, b\}$ are model parameters, $R = \|\theta\|_{L2}$ regularization term, and λ is a hyper-parameter measuring the weight of regularization term.

4 EXPERIMENTAL SETUP

4.1 Experimental Datasets

We use three CQA corpora (dataset of SemEval-2015 Task 3 [29], SemEva-2016 Task 3 [11], and SemEval-2017 Task 3 [30]) to train and evaluate our model. These corpora aim at the answer selection task in CQA by identifying good answers vs. potentially useful answers vs. bad or useless answers. Each dataset consists of a list of questions comprising a brief title and more informative descriptions, and the corresponding answers evaluated as good, potentially useful or bad. The statistics of three corpora are shown in Table 2.

4.2 Compared Methods

We evaluate and compare DGQAN with several state-of-the-art baseline models, including AP [31], ARC-I [32], ARC-II [32], ConvKN [33], DFFN [34], AI-CNN [10], IIT-UHH [35], Beihang-MSRA [36], Kelp [37], ECNU [38], UIA-LSTM-CNN [39], JAIST [12], BGMN [40], CNN [41], LSTM [42], Bi-LSTM-attention [42], CNN-LSTM-CRF [43], AP-LSTM [44], **KMAMA [14]** and **AUANN [45]**.

4.3 Training and hyper-parameters

We fix all the hyper-parameters applied to our model. Specifically, we use the basic version of BERT as the pre-trained embeddings in our experiments. The number of dual attention steps K is set to 2, which empirically shows the best performance. The question-subject, question-body, and answer are divided into fixed-length textual sequences to facilitate network processing. The algorithm we choose for optimization is Adam Optimizer with the first momentum coefficient β_1= 0.9 and the second momentum coefficient β_2=0.999. The initial learning rate, L2 regularization parameters, and batch size of the model are set to $\left[1 \times 10^{-9}, 4 \times 10^{-5}, 1 \times 10^{-7}\right]$, $\left[1 \times 10^{-6}, 4 \times 10^{-7}, 1 \times 10^{-7}\right]$, and [64, 128, 256], respectively. We use the best parameters on development sets, and evaluate the performance of our model on test sets. The whole experiments are implemented with Pytorch on NVIDIA TESLA V100 GPU.

5 EXPERIMENTAL RESULT

In this section, we elaborate the experimental setup and analyze the experimental results, aiming to answer:

RQ1: Can DGQAN achieve better answer selection performance than the state-of-the-art methods in answer selection task?

RQ2: How do the key model components and information types applied in DGQAN contribute to the overall performance?

RQ3: Can DGQAN more effective than other baseline methods in a specific case?

Table 2: Statistics of three CQA datasets

Method	SemEval-2015			SemEval-2016			SemEval-2017		
	Train	Dev	Test	Train	Dev	Test	Train	Dev	Test
#of Ques	2,376	266	300	4,897	244	327	5,124	327	293
#of Ans	15,013	1,447	1,793	36,198	2,440	3,270	38,638	3,270	3,930
Avg.sub.len	6.36	6.08	6.24	6.40	6.04	6.17	6.38	6.16	5.76
Avg.body.len	39.26	39.47	39.53	43.29	46.88	49.77	43.01	47.98	54.06
Avg.ans.len	35.82	33.90	37.33	37.76	36.18	37.27	37.67	37.30	39.50

RQ4: How does the size of these parameters hidden state dimension of dual attention, answer length, question subject length, and question body length in DGQAN affect the performance of the entire model?

RQ5: How do the different BERT layers used in DGQAN contribute to the overall performance?

5.1 Model Comparisons (RQ1)

To analyze the effectiveness of our model, we take some current competitive methods as baselines on the above three datasets.

As shown in Table 3, **AUANN** [45] and **KMAMA** [14] are strongest baseline models of the answer selection task. The experiment results show that our model DGQAN achieves best performance on three datasets. For instance, on the SemEval-2015 dataset, our model outperforms the previous best baseline model KMAMA [14] by 2.2% in F1 and 1.81% in Acc ($p < 0.05$ on student t-test). On the SemEval-2016 dataset, our model outperforms the previous best baseline model AUANN [45] by 3.93% in F1, 5.09% in Acc, and 4.86% in MAP ($p < 0.05$ on student t-test). On the SemEval-2017 dataset, our model outperforms the previous best baseline model AUANN [45] by 1.1% in F1, 1.37% in Acc, and 1.62% in MAP ($p < 0.05$ on student t-test). We attribute the performance enhancement to the attention mechanism applied in this paper. Specifically, graph convolution networks between question subjects and answers help our model estimate the relevance of subject-answer pairs and body-answer pairs. In the similar way, graph convolution networks between question bodies and answers also contribute. Hence, our method can effectively capture crucial interaction semantic features between questions and answers. QSubject-Answer inner attention focuses on assigning different weights to words, finding important words in question subjects. Dual graph attention mechanism aims at capturing useful features at word-level and sentence-level to indicate important words in a sentence and important sentence in a document (i.e., a question body or an answer). Therefore, our method can fully utilize the deep semantic information to solve the task.

5.2 Ablation Study (RQ2)

To thoroughly figure out the effect of each key model component, we carry out a series of ablation study to decompose the whole model into five derived models. To validate our model more effectively, we conduct ablation study in three datasets (SemEval-2015, SemEval-2016, and SemEval-2017). The results are shown in Figure 5.

Model (1): without position embedding – The entire model excludes position embedding in the embedding layer.

Model (2): only use question-subject and answer – The entire model only uses question-subject and answer.

Model (3): only use question-body and answer – The entire model only uses question-body and answer.

Model (4): without graph convolution networks – The entire model excludes the graph convolution networks.

Model (5): without dual attention – The entire model excludes the dual attention.

From Figure 5, we observe that the approach **Model (1)** has an obvious performance decay on all three datasets. These results demonstrate that position information is indispensable, which plays a significant role in enhancing the performance of the answer selection model.

To verify the core components of our proposed model, we design four variants to make comparisons. Firstly, comparison between **Model (2)** and **Model (3)** shows that the performance of **Model (3)** increases by 0.96%, 0.77%, and 1.07% ($p < 0.05$ on student t-test), in terms of Acc, on the SemEval-2017, SemEval-2016, and SemEval-2015 dataset, respectively. The main reason could be that question bodies often are more lengthy than question subjects, hence, containing more semantic information. In other words, question bodies can express the semantic meaning of questions more accurately than question subjects.

Secondly, we compare **our model (DQGAN)** with **Model (4)** to demonstrate the importance of the graph neural network component in our method. The performance of **Model (4)** reduces by 6.84%, 7.11%, and 7.32% ($p < 0.05$ on student t-test), in Acc on the SemEval-2017, 2016, and 2015 dataset, respectively. This is mainly because the graph neural network can better capture the semantic correlation between question subject and answer, as well as question body and answer. Therefore, when this part of the model is excluded, the performance is inevitably degraded.

Thirdly, we conduct a comparison between **Model (5)** and **our model (DQGAN)**. The performance of **Model (5)** reduces by 6.76%, 6.93%, and 6.94% ($p < 0.05$ on student t-test), in Acc on the SemEval-2017, 2016, and 2015 dataset, respectively. These results demonstrate that the dual attention mechanism is an essential part of our proposed method. This mainly because the dual attention mechanism can better capture the semantic correlation between the QSubject-Answer Graph and the QBody-Answer Graph. Hence, excluding this part could deteriorate the overall model performance.

Finally, comparison between **Model (4)** and **Model (5)** shows that the performance of **Model (4)** reduces by 0.08%, 0.18%, and

Table 3: Performance of different approaches on the three CQA datasets

Model	SemEval-2015		SemEval-2016			SemEval-2017		
	Acc	F1	Acc	MAP	F1	Acc	MAP	F1
AP [31]	-	-	75.47	77.12	71.72	-	-	-
ARC-I [32]	-	-	74.07	77.05	69.50	-	-	-
ARC-II [32]	-	-	75.26	77.98	71.64	-	-	-
ConvKN [33]	-	-	75.54	77.66	66.16	-	-	-
DFFN [34]	-	-	76.67	82.34	66.22	-	-	-
AI-CNN [10]			76.87	80.14	73.03	78.24	88.33	77.75
IIT-UHH [35]	-	-	-	-	-	72.70	86.88	73.94
Beihang-MSRA [36]	-	-	-	-	-	51.98	88.24	68.40
Kelp [37]	81.96	80.73	75.11	79.19	64.36	73.89	88.43	69.87
ECNU [38]	-	-	74.31	77.28	66.72	78.43	86.72	77.67
UIA-LSTM-CNN [39]	-	-	78.17	80.86	68.44	77.13	87.96	76.45
AUANN [45]	-	-	**80.49**	**82.76**	**74.46**	**78.46**	**89.59**	**79.81**
JAIST [12]	79.10	78.96	-	-	-	73.78	87.24	68.04
BGMN [40]	81.24	80.22	-	-	-	74.75	87.68	75.39
CNN [41]	77.33	76.92	-	-	-	73.22	86.21	72.14
LSTM [42]	76.21	75.15	-	-	-	74.05	86.28	73.45
Bi-LSTM-attention [42]	81.12	79.09	-	-	-	76.60	88.05	74.82
CNN-LSTM-CRF [43]	82.15	81.33	-	-	-	77.18	87.66	77.04
AP-LSTM [44]	79.45	79.06	-	-	-	77.64	87.82	76.82
KMAMA [14]	**86.98**	**85.45**	-	-	-	**82.32**	**90.76**	**81.15**
DGQAN (ours)	**88.79**	**87.65**	**85.58**	**87.62**	**78.39**	**83.69**	**92.38**	**82.25**

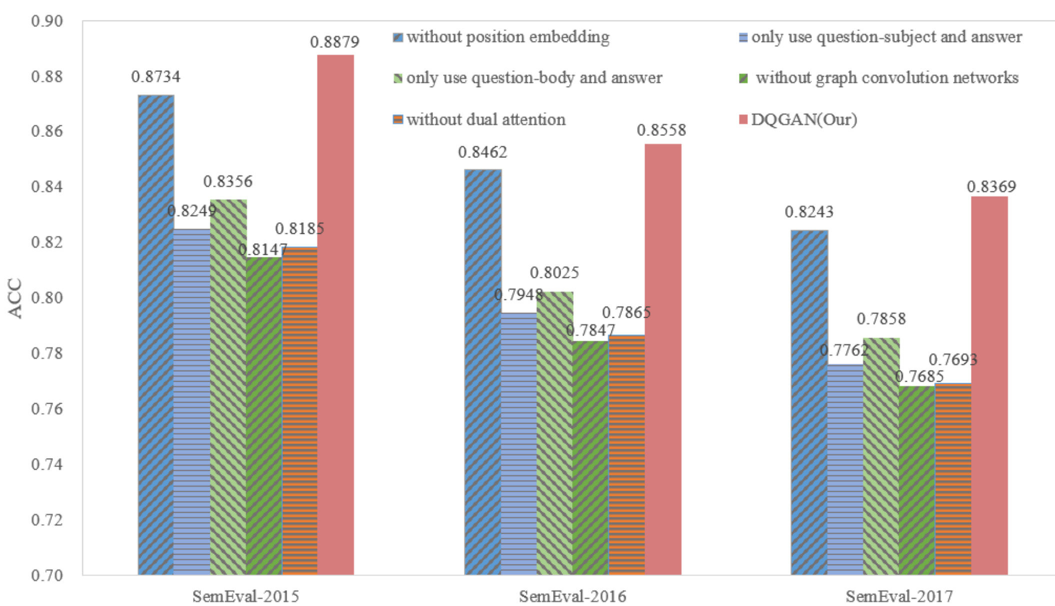

Figure 2: Ablation study on SemEval-2015, SemEval-2016, and SemEval-2017 dataset

0.38% ($p < 0.05$ on student t-test), in Acc on the SemEval-2017, 2016, and 2015 dataset, respectively. These results mean that excluding the graph neural network deteriorate the model slightly more than excluding the dual attention. These results demonstrate that though both the graph neural network component and the dual attention component make great contributions to our approach, the graph neural network component could be more important to our method slightly.

To sum up, our proposed model uses graph neural network to study the relationship among question subjects, question bodies, and answers. Also, our model adopt dual attention mechanism to fully capture interactive features in sequences at word-level and sentence-level to furthest enhance the performance of the answer selection approach.

5.3 Case Study (RQ3)

To investigate how our proposed model DQGAN works on real cases, we list a typical example from SemEval-2017 in Table 4. This sample is incorrectly predicted by well-performed models ECNU and IIT-UHH, while is correctly predicted by our model. We observe that ECNU and IIT-UHH tend to assign a higher score to the "Negative Answer" than the "Positive Answer", because the "Negative Answer" is more similar to the given Question at the word level, such as "passport" and "VISA". Different from them, our model can better obtain the deeper semantic information of question subject and answer by building CQA Concept Graph. Also, our method can further learn the hidden semantic relationships between question-answer pair by attention mechanism. Thus, our model show better results.

5.4 Parameter Sensitivity (RQ4)

In this section, we evaluate the impact of some parameters, such as hidden state dimension of dual attention, length of answers (the words of each answer), length of question subject (the words of question subject), and length of question body (the words of each question body) on all three datasets (SemEval-2015, SemEval-2016, and SemEval-2017 datasets).

Firstly, we investigate the impact of hidden state dimension of dual attention with results shown in Figure 3. We can observe that on three datasets, the MAP of our model shows an upward trend when the dimension size is less than 300. Especially achieving highest when the dimension size is exactly 300, which indicates that a large dimension size could contribute to model performance. However, when the dimension size is larger than 300, the MAP of the model drops on both development and test sets, possibly due to insufficient training data.

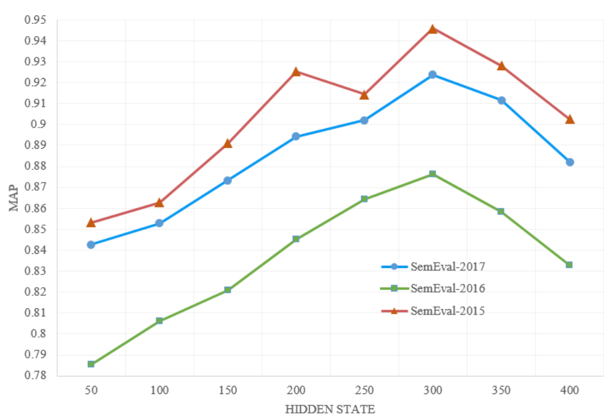

Figure 3: Performance of model DGQAN influenced by different hidden state dimensions of dual attention

Secondly, we compare the performance of our model using answers with different lengths on three datasets. As illustrated in Figure 4, our model achieves the best MAP, when the length of answers is 100. Because shorter answers may lose useful information while longer answers may introduce extra noises.

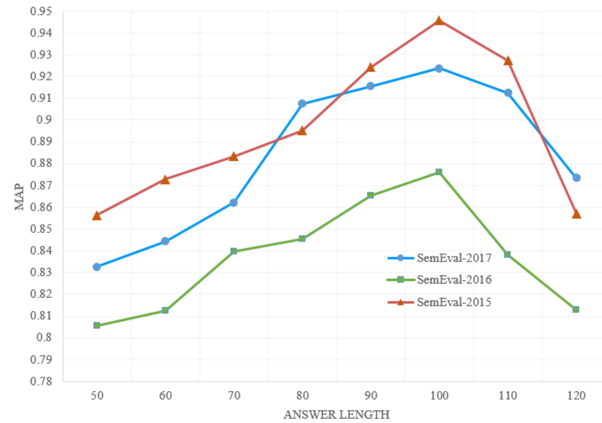

Figure 4: Performance of model DGQAN influenced by answers with different lengths

Thirdly, we further compare the performance of our model using question subject of different lengths on three datasets. As illustrated in Figure 5, our model achieves the best accuracy, when the length of question subject is 30. Because shorter question subject may lose useful information while longer question subject may introduce extra noises.

Figure 5: Performance of model DGQAN influenced by question subject with different lengths

Fourthly, we compare the performance of our model using question body with different lengths on three datasets (SemEval-2015, SemEval-2016, and SemEval-2017). As illustrated in Figure 6, our model achieves the best accuracy with the length of answers to 130. By comparing Figure 6 and Figure 5, we note that the highest performance is achieved when Figure 6 shows a longer text length than that of Figure 5. This main reason could be that the question-body is often a supplement and explanation of the question-subject.

Table 4: Case Study

Question:	Hi, can anybody tell me how I could get 3 visit VISA for my family? Suggestion for hotel names and contact numbers will be highly appreciated. Thank you!
Positive Answer:	At the immigration department. Prepare the application and documents and submit them.
User's Historical Answer:	It depends on your VISA. If you entered Qatar with the Visit VISA, you don't need an exit permit. If you came on Business VISA and stay more than 1 month, you will need an exit permit.
Negative Answer:	...I decided to renew my passport. The problem is my renewed passport NO. will not be the same with my passport NO. indicated in my VISA. Would the Embassy still visa-stamp on my new passport.

Consequently, more noise is introduced and longer sequences are required to capture better semantic information.

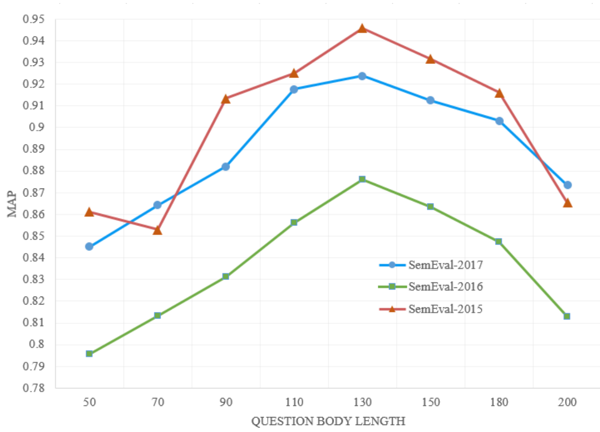

Figure 6: Performance of model DGQAN influenced by question body with different lengths

5.5 Analysis of Different BERT Layers (RQ5)

Adaptation at Different Layers: To understand the impact of each layer in the BERT model beyond the current knowledge, in this paper, we apply the graph neural network and the dual attention components, while investigating the capabilities of different Transformer layers of BERT. Specifically, we use single-layer Transformer, multiple-layer Transformer, and full-layer Transformer, respectively. Given a single layer, we conduct our investigation after the k-th layer, $k \in \{1, 3, 6, 9, 12\}$; while given multiple layers, we choose the layer of {1,3}, {1,3,6}, {1,3,6,9}, respectively. For the full-layer Transformer, we operate the analysis after each layer is represented. The experimental on SemEval-2017 dataset results are demonstrated in Table 5.

We can observe that the shallow layer is more conducive to obtaining better performance. The reason could be that the shallow layer promotes a more layered interaction between the textual features and the BERT. Further, we can note that applying the graph neural network and the dual attention components to multiple layers of BERT deteriorates the performance. One possible reason could be that multi-layer integration can lead to over-fitting.

Table 5: Results of variations of our model at different layers of BERT model. *one*, *multi*, *full* mean applying the graph neural network and the dual attention components to one layer, multiply layers, and full layers of Transformer in BERT.

Layer	*one*				
	1	3	6	9	12
F1	0.8043	0.8225	0.7968	0.7854	0.7698
Layer	*multi*			*full*	
	1,3	1,3,6	1,3,6,9	full	
F1	0.8136	0.8117	0.8024	0.7469	

Tuning BERT or Not: Intuitively, it is faster to integrate textual features into BERT without fine-tuning. However, the question features and the answer features encoded by BERT are distinctive. In order to evaluate the impact of fine-tuning and non-fine-tuning, we conduct experiments while the BERT parameters are fine-tuned and are not fine-tuned, respectively. The results show that without fine-tuning BERT, the f1 score of the experimental evaluation index drops by 0.2383 (from 0.8225 to 0.5842), indicating that fine-tuning BERT is more suitable for the answer selection task.

6 CONCLUSION

Community question answering (CQA) platforms attract many attention during the past decade. One can freely ask any question and expect some useful and explicit answers. However, it takes efforts to go through all possible answers and select the most relevant one. To automate this process, we propose Dual Graph Question-Answer Attention Networks for the task of answer selection, which take the relationship between questions and answers as important information. We use graph convolution networks between question subjects and answers, question bodies and answers to capture interactive features. Moreover, QSubject-Answer inner attention on question subjects and QBody-Answer inner attention on question bodies and answers help to assign different weights to each word so as to determine important words in a sentence. We conduct extensive experiments on three real-world datasets collected from representative CQA platforms and the results demonstrate the superiority of the proposed model, validating the contributions of the key model components. Also, we conduct experiment to investigate the impact of each layer in the BERT model. The results show that the shallow layer is more conducive to obtaining better performance.

REFERENCES

[1] Chaogang Fu. User intimacy model for question recommendation in community question answering. *Knowledge-Based Systems*, 188:104844, 2020.

[2] Dipankar Kundu, Rajat Kumar Pal, and Deba Prasad Mandal. Topic sensitive hybrid expertise retrieval system in community question answering services. *Knowledge-Based Systems*, 211:106535, 2021.

[3] Zizheng Lin, Haowen Ke, Ngo-Yin Wong, Jiaxin Bai, Yangqiu Song, Huan Zhao, and Junpeng Ye. Multi-relational graph based heterogeneous multi-task learning in community question answering. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 1038–1047, 2021.

[4] Nouha Othman, Rim Faiz, and Kamel Smaïli. Learning english and arabic question similarity with siamese neural networks in community question answering services. *Data & Knowledge Engineering*, page 101962, 2021.

[5] Yang Deng, Wai Lam, Yuexiang Xie, Daoyuan Chen, Yaliang Li, Min Yang, and Ying Shen. Joint learning of answer selection and answer summary generation in community question answering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 7651–7658, 2020.

[6] Yang Deng, Yuexiang Xie, Yaliang Li, Min Yang, Wai Lam, and Ying Shen. Contextualized knowledge-aware attentive neural network: Enhancing answer selection with knowledge. *ACM Trans. Inf. Syst.*, 40:2:1–2:33, 2022.

[7] Zhixing Tian, Yuanzhe Zhang, Xinwei Feng, Wenbin Jiang, Yajuan Lyu, Kang Liu, and Jun Zhao. Capturing sentence relations for answer sentence selection with multi-perspective graph encoding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 9032–9039, 2020.

[8] Haitian Yang, Chonghui Zheng, Xuan Zhao, Yan Wang, Zheng Yang, Chao Ma, Qi Zhang, and Weiqing Huang. Bertdan: Question-answer dual attention fusion networks with pre-trained models for answer selection. In *International Conference on Neural Information Processing*, pages 520–531. Springer, 2021.

[9] Haitian Yang, Weiqing Huang, Xuan Zhao, Yan Wang, Yuyan Chen, Bin Lv, Rui Mao, and Ning Li. Amqan: adaptive multi-attention question-answer networks for answer selection. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 584–599. Springer, 2020.

[10] Xiaodong Zhang, Sujian Li, Lei Sha, and Houfeng Wang. Attentive interactive neural networks for answer selection in community question answering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31, 2017.

[11] PreslavNakov LluısMarquez AlessandroMoschitti, WalidMagdy HamdyMubarak AbedAlhakimFreihat, James Glass, and Bilal Randeree. Semeval-2016 task 3: Community question answering. *Proceedings of SemEval*, pages 525–545, 2016.

[12] Quan Hung Tran, Duc-Vu Tran, Tu Vu, Minh Le Nguyen, and Son Bao Pham. Jaist: Combining multiple features for answer selection in community question answering. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 215–219, 2015.

[13] Simone Filice, Giovanni Da San Martino, and et al. Kelp at semeval-2017 task 3: Learning pairwise patterns in community question answering. In *(SemEval-2017)*, pages 326–333, 2017.

[14] Min Yang, Lei Chen, Xiaojun Chen, Qingyao Wu, Wei Zhou, and Ying Shen. Knowledge-enhanced hierarchical attention for community question answering with multi-task and adaptive learning. In *IJCAI*, pages 5349–5355, 2019.

[15] Yikang Shen, Wenge Rong, Nan Jiang, Baolin Peng, Jie Tang, and Zhang Xiong. Word embedding based correlation model for question/answer matching. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31, 2017.

[16] Yi Tay, Luu Anh Tuan, and Siu Cheung Hui. Cross temporal recurrent networks for ranking question answer pairs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[17] Wei Wu, Xu Sun, and Houfeng Wang. Question condensing networks for answer selection in community question answering. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1746–1755, 2018.

[18] Xiaoqiang Zhou, Baotian Hu, and et al. Recurrent convolutional neural network for answer selection in community question answering. *Neurocomputing*, 274:8–18, 2018.

[19] Wenxuan Zhang, Yang Deng, and Wai Lam. Answer ranking for product-related questions via multiple semantic relations modeling. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, pages 569–578, 2020.

[20] Haitian Yang, Weiqing Huang, Xuan Zhao, Yan Wang, Yuyan Chen, Bin Lv, Rui Mao, and Ning Li. Amqan: adaptive multi-attention question-answer networks for answer selection. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 584–599. Springer, 2020.

[21] Siddhant Garg, Thuy Vu, and Alessandro Moschitti. Tanda: Transfer and adapt pre-trained transformer models for answer sentence selection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 7780–7788, 2020.

[22] Seunghyun Yoon, Franck Dernoncourt, Doo Soon Kim, Trung Bui, and Kyomin Jung. A compare-aggregate model with latent clustering for answer selection. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 2093–2096, 2019.

[23] Dongfang Li, Yifei Yu, Qingcai Chen, and Xinyu Li. Bertsel: Answer selection with pre-trained models. *ArXiv*, abs/1905.07588, 2019.

[24] Zhichun Wang, Qingsong Lv, Xiaohan Lan, and Yu Zhang. Cross-lingual knowledge graph alignment via graph convolutional networks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 349–357, 2018.

[25] Zhiqing Sun, Jian Tang, Pan Du, Zhi-Hong Deng, and Jian-Yun Nie. Divgraphpointer: A graph pointer network for extracting diverse keyphrases. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 755–764, 2019.

[26] Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186, 2019.

[27] Rada Mihalcea and Paul Tarau. Textrank: Bringing order into text. In *Proceedings of the 2004 conference on empirical methods in natural language processing*, pages 404–411, 2004.

[28] Thomas Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *ArXiv*, abs/1609.02907, 2017.

[29] Preslav Nakov, Lluís Màrquez, Walid Magdy, Alessandro Moschitti, Jim Glass, and Bilal Randeree. Semeval-2015 task 3: Answer selection in community question answering. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 269–281, 2015.

[30] Preslav Nakov, Doris Hoogeveen, Lluís Màrquez, Alessandro Moschitti, Hamdy Mubarak, Timothy Baldwin, and Karin Verspoor. Semeval-2017 task 3: Community question answering. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 27–48, 2017.

[31] Cícero Nogueira dos Santos, Ming Tan, Bing Xiang, and Bowen Zhou. Attentive pooling networks. *ArXiv*, abs/1602.03609, 2016.

[32] Baotian Hu, Zhengdong Lu, Hang Li, and et al. Convolutional neural network architectures for matching natural language sentences. In *Advances in neural information processing systems*, pages 2042–2050, 2014.

[33] Alberto Barrón-Cedeno, Daniele Bonadiman, Giovanni Da San Martino, Shafiq Joty, Alessandro Moschitti, Fahad Al Obaidli, Salvatore Romeo, Kateryna Tymoshenko, and Antonio Uva. Convkn at semeval-2016 task 3: Answer and question selection for question answering on arabic and english fora. In *Proceedings of the 10th international workshop on semantic evaluation (SemEval-2016)*, pages 896–903, 2016.

[34] Sai Praneeth Suggu, Kushwanth Naga Goutham, Manoj Chinnakotla, and Manish Shrivastava. Hand in glove: deep feature fusion network architectures for answer quality prediction in community question answering. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1429–1440, 2016.

[35] Titas Nandi, Chris Biemann, and et al. Iit-uhh at semeval-2017 task 3: Exploring multiple features for community question answering and implicit dialogue identification. In *(SemEval-2017)*, pages 90–97, 2017.

[36] Wenzheng Feng and Yu andet al. Wu. Beihang-msra at semeval-2017 task 3: A ranking system with neural matching features for community question answering. In *(SemEval-2017)*, pages 280–286, 2017.

[37] Simone Filice, Danilo Croce, and et al. Kelp at semeval-2016 task 3: Learning semantic relations between questions and answers. In *(SemEval-2016)*, pages 1116–1123, 2016.

[38] Yu Wu, , Wenzheng Feng, and et al. Beihang-msra at semeval-2017 task 3: A ranking system with neural matching features for community question answering. In *(SemEval-2017)*, pages 280–286, 2017.

[39] Jiahui Wen, Jingwei Ma, Yiliu Feng, and Mingyang Zhong. Hybrid attentive answer selection in cqa with deep users modelling. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[40] Wei Wu, Houfeng Wang, and Sujian Li. Bi-directional gated memory networks for answer selection. In *Chinese Computational Linguistics and Natural Language Processing Based on Naturally Annotated Big Data*, pages 251–262. Springer, 2017.

[41] Lei Yu, Karl Moritz Hermann, Phil Blunsom, and Stephen G. Pulman. Deep learning for answer sentence selection. *ArXiv*, abs/1412.1632, 2014.

[42] Ming Tan, Bing Xiang, and Bowen Zhou. Lstm-based deep learning models for non-factoid answer selection. *ArXiv*, abs/1511.04108, 2015.

[43] Yang Xiang, Xiaoqiang Zhou, Qingcai Chen, Zhihui Zheng, Buzhou Tang, Xiaolong Wang, and Yang Qin. Incorporating label dependency for answer quality tagging in community question answering via cnn-lstm-crf. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1231–1241, 2016.

[44] Cícero Nogueira dos Santos, Ming Tan, Bing Xiang, and Bowen Zhou. Attentive pooling networks. *ArXiv*, abs/1602.03609, 2016.

[45] Yuexiang Xie, Ying Shen, Yaliang Li, Min Yang, and Kai Lei. Attentive user-engaged adversarial neural network for community question answering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 9322–9329, 2020.