Knowledge-Based Systems 171 (2019) 106–119

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

Advanced community question answering by leveraging external knowledge and multi-task learning

Min Yang [a], Wenting Tu [b,*], Qiang Qu [a], Wei Zhou [c], Qiao Liu [d], Jia Zhu [e]

[a] Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen, China
[b] Department of computer science, Shanghai University of Finance and Economics, Shanghai, China
[c] School of Big Data and Software Engineering, Chongqing University, Chongqing, China
[d] Department of computer science, University of Electronic Science and Technology of China, Chengdu, China
[e] School of Computing Science, South China Normal University, Guangzhou, China

ARTICLE INFO

Article history:
Received 22 June 2018
Received in revised form 3 February 2019
Accepted 6 February 2019
Available online 27 February 2019

Keywords:
Community question answering
Knowledge base
Multitask learning
Multi-head attention
Interactive attention

ABSTRACT

Community question answering (CQA) is an important but challenging task. Meantime, as the theory of deep learning develops, remarkable progress has been made by deep neural networks. This paper studies an advanced deep neural network that not only uses external knowledge to learn better representations of questions and answers but also improves representation learning by considering question categorization as an auxiliary task. Specifically, we propose a novel Multi-task and Knowledge enhanced Multi-head Interactive Attention network for Community Question Answering (MKMIA-CQA). It contains a document modeling module responsible for utilizing external commonsense knowledge to help identify background information (entity mentions and their relations) and filter out noise information from the long text which has complicated semantic and syntactic structures. Moreover, the model is trained in a multi-task manner. It regards community question answering as the primary task and question categorization as the auxiliary task, which aims to learn a category-aware encoder and improve the quality of locating the salient information of a long question. The experimental results on three widely used CQA datasets demonstrate that our model achieves impressive results compared to other strong competitors.

1. Introduction

Community question answering (CQA) platforms, e.g. Yahoo! Answer, Quora and Stack Overflow, have gained increasing popularity for people to share their knowledge and learn from each other. The new questions and corresponding answers are expanding rapidly since these forums are quite open and typically have few restrictions on who can submit questions or answers. These questions and answers are valuable to individual users for seeking the suitable answers to their questions. Unfortunately, reading through all the provided answers is difficult, if not impossible, especially for the answers that are lengthy and have low readability. It is therefore essential and of practical importance to propose automated techniques to pick out the best suitable answers for the given questions.

Most answer selection tasks such as WikiQA [1] and InsuranceQA [2] deal with simple single sentence queries whose answers are simple facts within one sentence. These questions are direct and rarely contain noise [3]. However, there are several key differences between a general answer selection task and a CQA task. First, a question in CQA is composed of two parts: (i) a question title that briefly summaries the question, and (ii) a body that describes the question in detail. Second, questions and answers in CQA generally consist of multiple sentences, in which some sentences are noises and do not provide meaningful information. In addition, these questions and answers are often informal, which contains abbreviations, typos, grammatical mistakes, etc. A typical example question and two possible answers in a community forum are exhibited in Table 1.

Earlier efforts in CQA focus on designing a set of features such as translation features, frequency features, structural features and community features, to rank the answers by explicitly modeling the correlations between text fragments in questions and answers. However, these methods require a large amount of manual work and domain expertise. The feature engineering is labor intensive and almost reaches its performance bottleneck. Inspired by the recent success of deep learning techniques in

* Corresponding author.
E-mail addresses: min.yang1129@gmail.com (M. Yang), tu.wenting@mail.shufe.edu.cn (W. Tu), qiang@siat.ac.cn (Q. Qu), zhouwei@cqu.edu.cn (W. Zhou), qliu@uestc.edu.cn (Q. Liu), jzhu@m.scnu.edu.cn (J. Zhu).

https://doi.org/10.1016/j.knosys.2019.02.006
0950-7051/

computer vision and natural language processing, many deep neural networks such as convolutional neural network (CNN) [4] and long short-term memory (LSTM) [5] have been proposed to automatically select answers in CQA. The key idea behind deep neural networks is to encode the input sentences as vector representations. Based on the representations, an output layer is utilized to provide the matching score of the question–answer pair. These methods can capture the semantic regularities in language and avoid feature engineering. Instead of learning the representations of the question and the answer separately, some recent studies exploit attention mechanisms to learn the interaction information between questions and answers, which can better focus on relevant parts of the input.

Despite the effectiveness of previous studies, community question answering in real-world remains a challenge. (i) The commonsense knowledge or factual background knowledge in the knowledge bases (KBs), such as Freebase [6], BabelNet [7], YAGO [8], DBpedia [9], and ConceptNet [10], provide rich information of entities and relations between entities, and highlight the parts that are important to text matching. Therefore, the external background knowledge is helpful for CQA, especially when the questions and answers are long. Considering the example in Table 1, conventional CQA methods may assign a higher score to the negative answer than the positive answer, since the negative answer is more similar to the given question at the word level. However, with the background knowledge, we can correctly identify the positive answer based on the relative facts contained in the KB such as (exit_permit, issued_by, labor_department). Despite its usefulness, to our best knowledge, the external commonsense knowledge from KBs or WordNet receives little attention in recent deep neural networks in CQA. (ii) A model which does not recognize the category of the input text could learn a text encoder missing the salient information in the text. Yet, an improved CQA system whose document modeling component have been co-trained with question categorization could learn a more satisfied category-aware question representation for accurately and correctly representing the given question.

The motivation of this paper is to leverage the external background knowledge and question category information to help the comprehension of the question/answer and thus improve the performance of the CQA system. We propose a novel Multi-task and Knowledge enhanced Multi-head Interactive Attention network for Community Question Answering (MKMIA-CQA). Our main idea is to (i) leverage the category information of question via multi-task learning and (ii) incorporate the external background knowledge into the matching process by interactively learning the attention in the contexts and the entity mentions in KB, and generating the context and knowledge representations for the questions/answers separately.

MKMIA-CQA is a multi-task system, in which a document modeling module is shared across tasks. A question categorization model, the first subtask in MKMIA-CQA, is trained to predict the category information of the given question. It helps the semantic analysis and comprehension of the input documents by locating the salient information of questions and learning robust question-aware representations of questions. In the second subtask, the representations of questions and answers are composed to perform CQA. The main purpose of our multi-task model is to strengthen the representation learning of questions, and safeguard the performance of CQA.

Specifically, MKMIA-CAQ is performed by three modules: *initial representation learning*, *interactive-learning based representation learning*, and *multi-task learning*. In *initial representation learning*, MKMIA-CQA employs GRU layers to learn the hidden representations of the question title, the question description and the answer. In order to reduce the impact of redundancy and noises in both long questions and answers, the concise question subject is used as the attention source to capture the important information of the question body and the answer, which learns the *initial context representations* of the questions and answers. Meanwhile, we also learn the distributed representations of entities in KB by DeepWalk [11] which is one of the best performers in graph embedding. The embeddings of the extracted entity mentions from KB are inputted into a convolutional layer and an attention layer to form the *initial knowledge representations* of the questions and answers. In *interactive-learning based representation learning*, a multi-head interactive attention mechanism is designed to capture the correlation of the context representations and the knowledge representations of question/answer, which makes use of the interactive information from contexts of question/answers and the knowledge base to supervise the modeling of each other. In *multi-task learning module*, the context representations and knowledge representations of questions/answers are concatenated to jointly perform question categorization and community question answering.

Compared with the prior CQA methods, the main contributions of this work are five-fold.

- *The main innovation of this paper* is to leverage external knowledge from KB to capture background information of the questions and answers for effective community question answering. To the best of our knowledge, we are the first to incorporate external commonsense knowledge from KB into the deep neural networks to improve the performance of CQA.
- We leverage question categorization task to learn better question representations for CQA, improving the quality of locating salient information of the questions.
- We make use of the interactive guidance between the external knowledge and the contexts of questions/answers by adopting an interactive attention mechanism. It is helpful to capture important information from both context and knowledge representations of questions/answers.
- Multi-head attention is adopted to model the overall semantics of the text, which helps to capture the important information from different representation subspaces at different positions.
- Extensive experiments have been conducted to evaluate the performance of our model. The experimental results demonstrate that MKMIA-CQA achieves substantial improvements over the compared methods.

The remainder of this paper is structured as follows. Section 2 reviews and discusses the related work. In Section 3, we fully describe the proposed MKMIA-CQA model. The experimental setup is introduced in Section 4, including the evaluation datasets, the compared methods, the automatic evaluation metrics, and the implementation details. Section 5 shows the evaluation results and analysis. Section 6 makes the conclusions and discusses the future work.

2. Related work

The goal of community question answering task is to automatically pick out the most relevant answers from many possible answers from community question answering forums. It is a challenging problem that has been receiving much attention from the natural language processing community in recent years. In this section, we primarily review the related work in community question answering and knowledge base applications.

Table 1
Example of one question and its two answers from SemEval-2017.

Question subject	How to go home without exit permit.
Question body	My sponsor is not willing to give me exit permit in the fear of my not return. I am about 3 years at Qatar; and I want to go home on a vacation before Eid. Is there any way to go home without the exit permit from sponsor? I want to go home before Eid; no matter what does it cost; even cancellation of my visa. Thank you for your advice.
Positive answer	You have 7 days... You left it a bit late to resign; no? If you are due to go home they have to give you a ticket. If not; then go to the Labour Department and report him.
Negative answer	My sponsor is not willing to let me go in any way until all of my projects are finished. Can my embassy or the labor department help?

2.1. Conventional community question answering approaches

The majority of previous approaches focused on syntactic matching between questions and answers. Cui et al. [12] was one of the earliest to use the general tree matching methods based on tree-edit distance. Wang et al. [13] proposed quasi-synchronous grammar to match each question/answer pair by the dependency trees. Later, Heilman and Smith [14] employed a logistic together with a tree kernel function and extracted features to learn the associations between the question/answer pair. Surdeanu et al. [15] studied a wide range of feature types such as translation features, frequency features and similarity features for answer ranking of non-factoid questions. Best results are obtained by aggregating the models trained with features that model question-to-answer transformations, frequency of content, and similarity features of QA pairs.

Some translation based methods [16–19] were proposed to combine the translation model and the language model for question retrieval in CQA by finding the similar questions for the user queried questions. For example, Xue et al. [17] presented a word-based translation language model for question retrieval. Lee et al. [18] further improved the translation probabilities based on question–answer pairs by choosing the most important terms to build compact translation models. Zhou et al. [19] presented the phrase-based translation method for question retrieval in CQA, which extended the word-based translation method with the contextual information. While useful, the effectiveness of these models highly relied on the availability of quality parallel question/answer pairs in the absence of which they are troubled by noise issue.

Topic modeling based approaches, such as probabilistic latent semantic analysis (PLSA) [20] and latent Dirichlet allocation (LDA) [21], were also widely used for question retrieval in CQA. Cai et al. [22] combined the semantic similarity based latent topics with the translation-based language model to improve the question retrieval performance. Ji et al. [23] proposed a question–answer topic model to capture the question/answer relationships, with a assumption that the question and the corresponding answer share the same topic distribution. Chen et al. [24] used the topic information of queries to improve the topic inference based translation language model. However, questions and answers do not share the same latent topic distribution in many cases since they are different in many aspects, which may result in performance of question/answering matching.

Subsequently, the community question answering has been operationalized in SemEval-2015 Task 3 [25], SemEval-2016 Task 3 [26], and SemEval-2017 Task 3 [27]. The top performers at SemEval-2015/2016/2017 Task 3 included the JAIST model [28], HITSZ-ICRC model [29], KeLP model [30,31], and Beihang-MSRA [32]. For example, Tran et al. [28] studied topic model based features (16 features belong to 5 groups) to predict answer quality and Filice et al. [30] treated the CQA task as binary classification problem and explored similarity features, thread-based features, and heuristic features. Despite the effectiveness of these methods, they relied heavily on feature engineering which was time-consuming and labor intensive.

2.2. Deep neural networks for community question answering

To exploit the semantic regularities in language and avoid feature engineering, many deep neural networks have been proposed for CQA. The main idea is to encode the input sequences as vector representations, and the learned questions/answers representations are used to calculate the ranking scores of answers. For example, Iyyer et al. [33] developed a dependency tree recursive neural network and extended it to combine predictions across sentences to rank answers. Wang and Nyberg [34] employed a stacked bidirectional LSTM network to sequentially process words from the question and answer, and then computed their relevance scores. Bonadiman et al. [35] proposed a deep neural network (DNN) to solve simultaneously the three CQA tasks, i.e., question–comment similarity, question–question similarity and new question–comment similarity. The DNN was trained jointly on all the three CQA tasks and encoded each question/answer into a single vector representation shared across the multiple tasks. Guo et al. [36] further introduce a skip convolutional neural network to capture the lexical semantic features. Zhou et al. [37] applied convolution neural networks to learn the joint representation of question–answer pair, and then used the learned representation as input of the LSTM to label the matching score of each answer. Zhou et al. [38] combined CNN with LSTM to capture both the semantic matching between questions and answers and the semantic correlations among the sequence of answers.

2.3. Attention-based deep neural networks for community question answering

Some recent studies explored attention mechanisms to learn the interaction information between questions and answers, which can better focus on relevant parts of the input [41–43]. Tan et al. [41] introduced a hybrid neural network, learning question/answer representations by integrating the results from both CNN and RNN. They also empirically demonstrated the effectiveness of attention mechanism in answer selection. dos Santos et al. [42] proposed a two-way attention mechanism to project the paired input sequences into a common representation space for better answer ranking. Chen et al. [43] presented a positional attention model to incorporate the positional context of the question words into the answers' attentive representations. Xiang et al. [45] introduced an attention-based DNN architecture that was constituted by CNN, LSTM and conditional random fields

Table 2
Summarization of previous community question answering approaches.

Types	Approaches	Representative	Limitations
Conventional CQA methods	Translation-based methods	[16–19]	The effectiveness of these models highly relies on the availability of quality parallel QA pairs.
	Topic modeling based methods	[22–24]	Questions and answers do not share the same latent topic distribution in many cases since they are different in many aspects, which may result in poor performance of question/answering matching.
	Feature-based methods	[12–15]	These methods rely heavily on feature engineering which was time-consuming and labor intensive.
Deep Learning based CQA methods	CNN based methods	[35,36,39]	These methods do not consider the word orders and syntactic structures of the long text in CQA.
	RNN based methods	[33,34]	These methods do not capture the dependency relations from a global perspective and cannot learn the interaction between questions and answers.
	CNN and RNN hybrid methods	[37,38,40]	Although these method consider both n-gram features and word orders, they cannot assign different weights to the context words and provide explainable results.
	Attention-based DNN methods	[41–44]	These methods do not consider the question category and commonsense knowledge, which may be disturbed by the noisy information and complicated syntactic structures of long questions or answers.

(CRF) to capture the dependency relations from a global perspective. Zhang et al. [44] proposed an attentive interactive neural network to learn interactions of each paired question/answer representations learned by the CNNs. Wu et al. [46] presented a question condensing network (QCN) to exploit the subject–body relationship of community questions by employing the multi-dimensional attention mechanism.

The previous CQA methods are summarized and discussed in Table 2. Different from the aforementioned CQA methods, we attempt to explore the question category knowledge and the commonsense from knowledge base to filter noisy information in the long text in CQA.

2.4. Knowledge base as external knowledge in NLP

Commonsense knowledge or factual background knowledge about entities and relations has been extensively explored in a wide range of downstream natural language processing tasks. Wu et al. [47] proposed a knowledge enhanced hybrid neural network to fuse prior knowledge into text representations by knowledge gates. Yang and Mitchell [48] leveraged continuous representations of KBs to enhance the learning of LSTM for machine reading. Xin et al. [49] took information from KBs into consideration to improve the performance of fine-grained entity typing, which bridges entity mentions and their context together. Han et al. [50] shared the representation learning framework between knowledge graph completion and relation extraction tasks. The representations of knowledge graphs and text enjoyed the shared semantic space.

On the other hand, knowledge base embedding is another task which has attracted much attention. It aims to embed entities and relations in the knowledge base into continuously distributed representations which capture the semantics and structure information of KB. TransE [51] was the most representative graph embedding method. It represented both the entities and their relations as distributed vectors in a shared space. TransH [52] extended TransE by introducing relation-specific hyperplanes. It also modeled entities again as vectors, but each relation was represented as a vector on a hyperplane. DeepWalk [11] adopted the SkipGram algorithm for graph embedding. It aimed to maximize the co-occurrence probability among the tokens that appeared within a window. The success of DeepWalk motivated many subsequent studies applying deep learning algorithms such as LSTM for graph embedding.

2.5. Multi-task learning

Multi-task learning algorithms optimize multiple learning tasks simultaneously, and exploit the commonalities and differences across tasks, improving the generalization performance of the tasks [53,54]. For example, Liu et al. [55] trained several text classification tasks jointly by sharing information of the recurrent neural networks and investigated the empirical performances of the proposed models on the related text classification tasks (subjective/sentiment classification tasks). Liu et al. [56] combined tasks of multiple-domain classification (for query classification) and information retrieval (ranking for web search), not only leveraging large amounts of cross-task data, but also benefiting from a regularization effect that leads to more general distributed representations to enhance tasks in new domains. Luong et al. [57] combined multi-task learning with the encoder–decoder model, which shared the parameters of the encoder and decoder across the tasks. They demonstrated improvements in machine translation. Luan et al. [58] built a persona-based conversation agent by employing the multi-task learning algorithm to jointly train the neural conversation models that leverage both conversation data across speakers.

In this paper, we learn the external knowledge and the contexts of questions/answers interactively with attention mechanism for CQA task, which are helpful to capture important information from both context and knowledge representations of questions/answers.

3. Problem definition

This section defines the key notations and briefly formulates the problem of community question answering (CQA) we study

Table 3
Notation list.

Notation	Description
Q	Question containing the question subject and the question body
S	Question subject
B	Question body
A	Answer
e_k	Input word embedding at index k
h_k	Hidden state at time step k
d	Embedding dimension
d_h	Number of hidden states for each GRU
$\mathbf{r}_k$	The reset gate of GRU
$\mathbf{z}_k$	The update gate of GRU
H^s, H^b, H^a	Context-based hidden states of the question subject, question body and answer
n, m, l	Numbers of words in S, B and A, respectively
CS^{init}, CB^{init}, CA^{init}	Initial context representation of question subject, question body and answer
Z^s, Z^b, Z^a	Knowledge-based hidden states of the question subject, question body and answer
KS^{init}, KB^{init}, KA^{init}	Initial knowledge representation of question subject, question body and answer
α^b, α^a	Context attention weights of question body and answer, respectively
β^b, β^a	Knowledge attention weights of question body and answer, respectively
ρ	Attention function for the initial representation learning
C^b, C^a	Attention matrices of the ILB context representations of question and answer
K^b, K^a	Attention matrices of the ILB knowledge representations of question and answer
CB^{ILB}, CA^{ILB}	ILB context representation of question body and answer
KB^{ILB}, KA^{ILB}	ILB knowledge representation of question body and answer
OB^{ILB}, OA^{ILB}	ILB representations of question body and answer
ϱ	Attention function for the interactive learning based representation learning
M^{ILB}	Attention matrix based on ILB representations of question and answer
ω^b, ω^a	Attention vectors for the final representations of question body and answer
OB^{ILB}, OA^{ILB}	Final document representations of question body and answer

in this paper. Suppose that each question Q is composed of two parts: the subject of the question S and the body of the question B. Moreover, we also have a set of answers. Then, given a question $Q = (S, B)$ and a candidate answer A, the goal of CQA is to infer the label $Y \in \{\text{Good}, \text{Bad}\}$ where $Y = \text{Good}$ ($Y = \text{Bad}$) indicates the given answer A can (not) fit the given question Q well. Typically, S, B and A are text data. Thus, we denote them as $S = [s_1, s_2, \ldots, s_n]$, $B = [b_1, b_2, \ldots, b_m]$ and $A = [a_1, a_2, \ldots, a_l]$, where n, m and l are numbers of words in S, B and A. Recall that our work will discuss the use of the question-categorization task for improving the CQA task. Thus, we assume that each question Q has a category label F. To prevent conceptual confusion, we use the superscripts "s", "b", "a" to indicate the variables that are related to the question subject, question body, and answer, respectively. The main notations of this work are summarized in Table 3 for clarity.

4. Our methodology

Our model MKMIA-CQA, depicted in Fig. 1, consists of three modules. The first module ***initial representation learning*** learns the initial representations (including the initial context representations and initial knowledge representations) of the given question and answer. The second module ***interactive-learning based representation learning module*** aims to combine the strengths of the context and the knowledge representations by extracting interactive information between them. Finally, the model is trained via a ***multi-task learning module*** by regarding community question answering as the primary task and question categorization as the auxiliary task. In this rest of this section, we elaborate each component of the proposed MKMIA-CQA model in detail.

4.1. Initial representation learning

We employ GRU networks to learn the *initial context representations* of the question subject, the question description and the answer. Meanwhile, we also use DeepWalk [11], which is one of the most popular graph embedding methods, to learn the representations of entities and relations in KB. The embeddings of the extracted entity mentions from KB are concatenated to form the *initial knowledge representations* of the question and answer.

Initial context representations of questions/answers. First, MKMIA-CQA utilizes GRU layers and the attention mechanism to learn the initial representations of the given question $Q = (S, B)$ (where S and B denote the question subject and body, respectively) and the answer A. Recall that inputs S, B and A are text data. As the lower left corner of Fig. 1 shows, each word w in the input text is mapped to a low-dimensional embedding $e^w \in \mathbb{R}^d$ by embedding layers, where d denotes the embedding dimension. Then, hidden states of words in the question subject, the question body, and the answer are learned by GRU layers. Formally, given the input word embedding e_k at index k in the input text, the hidden state $h_k \in \mathbb{R}^{d_h}$ (d_h is the number of hidden states for each GRU) can be updated from the previous hidden state h_{k-1}, which is computed by

$$\mathbf{r}_k = \sigma(W_r \cdot e_k + U_r \cdot h_{k-1} + b_r), \tag{1}$$

$$\mathbf{z}_k = \sigma(W_z \cdot e_k + U_z \cdot h_{k-1} + b_z), \tag{2}$$

$$\tilde{h}_k = tanh(W_h \cdot e_k + U_h \cdot (\mathbf{r}_k \odot h_{k-1})), \tag{3}$$

$$h_k = \mathbf{z}_k \odot h_{k-1} + (1 - \mathbf{z}_k) \odot \tilde{h}_k. \tag{4}$$

where $\mathbf{r}_k$ and $\mathbf{z}_k$ are reset gate and update gate, respectively; W and U denote weight matrices to be learned; b represents biases; σ is a sigmoid function; $\cdot$ stands for matrix multiplication; and $\odot$ stands for element-wise multiplication. After processed by embedding and GRU layers, question subject S, question body B and answer A are represented by their hidden states $H^s = [h_1^s, h_2^s, \ldots, h_n^s]$, $H^b = [h_1^b, h_2^b, \ldots, h_m^b]$, and $H^a = [h_1^a, h_2^a, \ldots, h_l^a]$, where n, m and l are numbers of words in S, B and A.

In order to reduce the impact of redundancy and noise in both long questions and answers, the concise question subject is used as the attention source to capture the important information of the question body and the answer for CQA. Formally, the hidden states H^s are the input to a mean-pooling layer to obtain the representation of the question subject, denoted as CS_{init}:

$$CS^{init} = \sum_{i=1}^{n} h_i^s / n \tag{5}$$

Then, we use the representation of the question subject as attention source to capture the important information of the

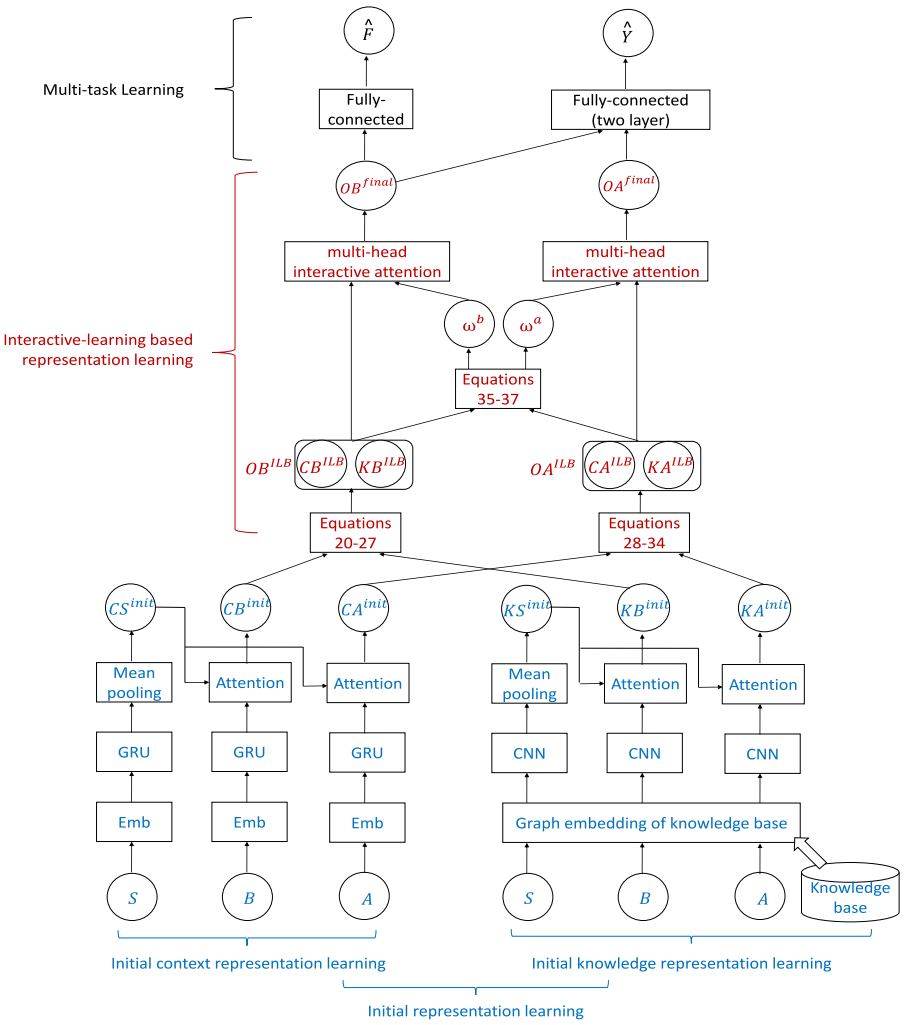

Fig. 1. The overview of MKMIA-CQA.

question and answer. The initial context representations of the question and answer can be computed by:

$$CB_i^{init} = \alpha_i^b \cdot h_i^b \tag{6}$$

$$\alpha_i^b = \frac{exp(\rho([h_i^b; CS^{init}]))}{\sum_{j=1}^{m} exp(\rho([h_j^b; CS^{init}]))} \tag{7}$$

$$CA_i^{init} = \alpha_i^a \cdot h_i^a \tag{8}$$

$$\alpha_i^a = \frac{exp(\rho([h_i^a; CS^{init}]))}{\sum_{j=1}^{l} exp(\rho([h_j^a; CS^{init}]))} \tag{9}$$

where α_i^b and α_i^a indicate the importance of the ith word in the question body and the answer. ρ is the attention function that calculates the importance of h_i^b in the question body, which is defined as:

$$\rho([h_i^b; CS^{init}]) = U_d^T tanh(W_d[h_i^b; CS^{init}]), \tag{10}$$

where U_d and W_d are projection parameters to be learned.

Initial knowledge representations of questions/answers. As showed by the lower right corner of Fig. 1, we use DeepWalk [11] to learn the graph embeddings of KB. Motivated by SkipGram, DeepWalk starts the random walks at each vertex of the KB, and generate a random ordering to traverse the vertices. Each vertex v_i is firstly mapped to its current distributed representation $\phi(v_i)$, and the probability of its neighbors in the random walk is maximized to optimize the model. We can learn such posterior distribution using the *softmax* function, which would result in a huge number of labels that is equal to the number of vertices. To speed the training time, Hierarchical Softmax is used to approximate the probability distribution. Formally, if the path to the vertex v_k is identified by a sequence of binary tree nodes $[b_0, b_1, \ldots, b_{\log|V|}]$ ($|V|$ is the number of vertices, b_0 is the root, $b_{\log|V|}$ is the vertex v_k), then the model can be defined as:

$$P(v_k|\phi v_i) = \sum_{j=1}^{\log|V|} P(b_j|\phi(v_i)), \tag{11}$$

where $P(b_j|\phi(v_i))$ can be a binary classifier that is assigned to the parent of the node b_j. Finally, we use stochastic gradient descent (SGD) to learn the parameters of the graph embedding model, and get the vertex embedding $\phi(v)$ for each vertex v.

After learning the graph embedding of KB, we perform entity mention detection by n-gram matching and provide a set of top-K entity candidates from KG for each entity mention in the input text. Formally, we present candidate entities for the entity mention at time step t as $\{e_1, e_2, \ldots, e_K\} \in \mathbb{R}^{K\times d}$. The word embedding in text and the entity embedding in KB have the same dimensionality. The candidate entities are averaged to form the final entity representation $\tilde{e}_t$ at time step t:

$$\tilde{e}_t = \sum_{i=1}^{K} e_i/K \tag{12}$$

A convolution layer is then employed to capture the local n-gram information and learn a higher level knowledge representation $E \in L \times d_e$ from the entity mention embeddings, where L is the number of entity mentions in the input text and d_e is the filter numbers of CNN. Here we use CNN layers (rather than GRU layers) since they can better capture structures of n-gram entities. The tth input of the convolutional layer x_t is the concatenation of the tth entity mention and its surrounding entity mentions in a window:

$$x_t = [\tilde{e}_{t-d_w/2}, \ldots, \tilde{e}_t, \ldots, \tilde{e}_{t+d_w/2}] \tag{13}$$

where d_w is the size of the window. The hidden states of convolutional layer are obtained by convolution of the input window, adding a bias term and then apply a non-linear function:

$$Z_t = ReLu(W_{conv} x_t + b_{conv}) \tag{14}$$

where W_{conv} is transformation matrix, b_{conv} is bias vector. Hence, we can obtain the hidden states of the question subject Z^s, question body Z^b, and answer Z^a. The hidden states Z^s are inputted into a mean-pooling layer to obtain the mean representation of the question subject, denoted as KS^{init}:

$$KS^{init} = \sum_{i=1}^{L_s} Z_i^s / L_s \tag{15}$$

where L_s is the number of entity mentions in the question subject.

Finally, we also use the question subject as attention source to capture the important knowledge-based information of the question and answer. The initial knowledge representations of the question body and answer (i.e., KB^{init} and KA^{init}) are computed by:

$$KB_i^{init} = \beta_i^b \cdot Z_i^b \tag{16}$$

$$\beta_i^b = \frac{exp(\rho([Z_i^b; KS^{init}]))}{\sum_{j=1}^{L_b} exp(\rho([Z_j^b; KS^{init}]))} \tag{17}$$

$$KA_i^{init} = \beta_i^a \cdot Z_i^a \tag{18}$$

$$\beta_i^a = \frac{exp(\rho([Z_i^a; KS^{init}]))}{\sum_{j=1}^{L_a} exp(\rho([Z_j^a; KS^{init}]))} \tag{19}$$

where β_i^b and β_i^a indicate the importance of the ith entity mention in the question body and the answer; L_b and L_a are the numbers of entity mentions in the question subject, question body and the answer, respectively.

4.2. Interactive-learning based (ILB) representation learning

We propose a multi-head interactive attention network (MIA) to combine the strengths of the initial context and knowledge representations. MIA makes use of the interactive guidance between the external knowledge and the contexts of questions/answers by adopting an interactive attention mechanism, which is helpful to capture important information from both context and knowledge representations of questions/answers. In addition, MIA adopts the multi-head attention capture the important information from different representation subspaces at different positions.

MIA produces 2-dimensional attention weight matrix. Formally, the implementation process of interactive multi-head attention for question representations is shown as follow. MIA takes as input the initial context representation B^{init} and knowledge representation E^b of the question, the attention matrix $C^b \in \mathbb{R}^{b \times m}$ for the context representation of the question is computed as:

$$C^b = [C_1^b, C_2^b, \ldots, C_m^b] \tag{20}$$

$$C_i^b = \frac{exp(\varrho([CB_i^{init}; \mu(KB^{init})]))}{\sum_{j=1}^{m} \exp(\varrho([CB_j^{init}; \mu(KB^{init})]))}, \tag{21}$$

where $C_i^b \in \mathbb{R}^b$ denotes the ith row of attention matrix which indicates the importance of the ith word in multiple hops of attention; b is the number of hops of attention; each row of attention matrix C^b denotes one hop of attention on the whole document, namely a single-head attention; μ is the average operation; and ϱ is the attention function that calculates the importance of CB_i^{init} in multiple hops of attention:

$$\varrho([CB_i^{init}; \mu(KB^{init})]) = (U_2^b)^T tanh(U_1^b [CB_i^{init}; \mu(KB^{init})]), \tag{22}$$

where U_2^b and U_1^b are attention parameters to be learned.

Only using the attention matrix C^b cannot capture the interactive information of the context and the knowledge representations, and lacks the ability of discriminating the importance of the knowledge representations. To make use of the interactive information between the context and knowledge representations of the question, we also use the context representations as attention source to attend to the knowledge representations. Similarly, we can calculate the attention matrix K^b for the knowledge representations as:

$$K^b = [K_1, K_2, \ldots, K_{L_b}] \tag{23}$$

$$K_i^b = \frac{exp(\varrho([KB_i^{init}; \mu(CB^{init})]))}{\sum_{j=1}^{L_b} exp(\varrho([KB_j^{init}; \mu(CB^{init})]))}, \tag{24}$$

where ϱ is the same as Eq. (22).

After computing the multi-head interactive attention matrices for the context and knowledge representations, we can get Interactive Learning Based (ILB) representations B_{ILB} and E_{ILB}^b based on the attention matrices C^b and M^b by:

$$CB^{ILB} = flat(C^b \cdot CB^{init}) \tag{25}$$

$$KB^{ILB} = flat(K^b \cdot KB^{init}), \tag{26}$$

where *flat* is an operation that flattens matrix into vector form. Finally, we concatenate CB^{ILB} and KB^{ILB} as the interactive learning based question representation OB^{ILB}:

$$OB^{ILB} = concat(CB^{ILB}, KB^{ILB}). \tag{27}$$

In the same way, we can also get the interactive learning based answer representation OA_{ILB}:

$$C^a = [C_1^a, C_2^a, \ldots, C_m^a] \tag{28}$$

$$C_i^a = \frac{exp(\varrho([CA_i^{init}; \mu(KA^{init})]))}{\sum_{j=1}^{m} \exp(\varrho([CA_j^{init}; \mu(KA^{init})]))}, \tag{29}$$

$$K^a = [K_1, K_2, \ldots, K_{L_a}] \tag{30}$$

$$K_i^a = \frac{exp(\varrho([KA_i^{init}; \mu(CA^{init})]))}{\sum_{j=1}^{L_a} exp(\varrho([KA_j^{init}; \mu(CA^{init})]))}, \tag{31}$$

$$CA^{ILB} = flat(C^a \cdot CA^{init}) \tag{32}$$

$$KA^{ILB} = flat(K^a \cdot KA^{init}) \tag{33}$$

$$OA^{ILB} = concat(CA^{ILB}, KA^{ILB}). \tag{34}$$

After obtaining the final representations of questions and answers, we further compute the interaction of the question and answer to capture their relations and focus on useful segments of text. Formally, the attention matrix for the interactive learning based representations is computed as:

$$M^{ILB} = softmax((OA^{ILB})^T \cdot U_\mu \cdot OB^{ILB}) \tag{35}$$

where $U_\mu \in \mathbb{R}^{d \times d}$ is a bilinear term, which represents the similarity between OA^{ILB} and OB^{ILB}, each element $M_{i,j}^{ILB}$ denotes the

relevant degree between the ith term in the answer representation and the jth term in the question representation. Then, we average the values of each column and each row of M^{ILB}, and take them as the input of softmax function to produce the attention vectors for the question and answer representations, respectively:

$$\omega^b = softmax(\sum_{i=1}^{|OB^{ILB}|} M^{ILB}[i,:]) \tag{36}$$

$$\omega^a = softmax(\sum_{j=1}^{|OA^{ILB}|} M^{ILB}[:,j]) \tag{37}$$

Finally, we conduct dot product between the attention vector ω^b (ω^a) and the question (answer) representation OB^{ILB} (OA^{ILB}) to obtain the final representations of question Q and answer A:

$$OB^{final} = (OB^{ILB})^T \cdot \omega^b \tag{38}$$

$$OA^{final} = (OA^{ILB})^T \cdot \omega^a \tag{39}$$

4.3. Multi-task learning

4.3.1. Auxiliary task: text categorization

Question categorization task assigns a category to the given question, which helps learn a category-specific text encoder and improves the quality of locating salient information of the question. This task can be seen as a multi-class classification problem.

We feed the final question representation OB^{final} into a task-specific fully-connected layer followed by a softmax layer (for probabilistic classification) to obtain the predicted category label $\hat{F}$ of question Q:

$$V^{cat} = \tanh(U_1^{cat} \cdot OB^{final} + b^{cat}) \tag{40}$$

$$\hat{F} = softmax(U_2^{cat} \cdot V^{cat}) \tag{41}$$

where U_1^{cat} and U_2^{cat} are projection parameters, b^{cat} is the bias term.

The parameters of question categorization model is learned in a supervised manner. In particular, given a labeled training data set $\{(S_{1:N}, B_{1:N}, A_{1:N}, F_{1:N})\}$, we minimize directly the cross-entropy between the predicted label distribution $\hat{F}$ and the ground truth distribution F as the objective function:

$$L_1 = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{J} \mathbf{I}\{F_i = j\} \log(\hat{F}_i), \tag{42}$$

where $\mathbf{I}\{\cdot\}$ is an indicator such that $\mathbf{I}\{\text{true}\} = 1$ and $\mathbf{I}\{\text{false}\} = 0$, J is the number of category classes, N is the number of training samples.

4.3.2. Primary task: Community question answering

In this task, the final question and answer representations are concatenated and fed into a task-specific two-layer feed-forward neural network. The softmax function is then applied to the end of the last layer to predict the answer probability distribution:

$$V^{qa} = U_2^{qa}\tanh(U_1^{qa}[OB^{final}, OA^{final}] + b_1^{qa}) + b_2^{qa} \tag{43}$$

$$\hat{Y} = softmax(U_3^{qa} \cdot V^{qa}), \tag{44}$$

where U_1^{qa}, U_2^{qa}, and U_3^{qa} are projection parameters, b_1^{qa} and b_2^{qa} are the bias terms.

Similar to question categorization, we also use cross-entropy as the objective function.

$$L_2 = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{D} \mathbf{I}\{Y_i = j\} \log(\hat{Y}_i), \tag{45}$$

where D is the number of answer classes belonging to *Good* or *Bad*.

4.3.3. Parameters optimization for multi-task learning

Overall, our model consists of two subtasks, each has a training objective. For the purpose of strengthening the learning of the share document-query representations, we train these two related tasks simultaneously. The joint multi-task objective function is minimized by:

$$L = \lambda_1 L_1 + \lambda_2 L_2, \tag{46}$$

where λ_1 and λ_2 are hyper-parameters that determines the weights of L_1 and L_2. We empirically show that setting $\lambda_1 = 0.2$ and $\lambda_2 = 0.8$ achieves the best performance.

5. Experimental setup

5.1. Datasets description

We conduct experiments on three widely used CQA datasets, i.e. SemEval-2015 Task 3 [25], SemEval-2016 Task 3 [26], and SemEval-2017 Task 3 [27], which contain real data from the QatarLiving forum.[1] Table 4 shows the statistics of these three datasets. Each question in the datasets consists of a short question title (question subject) and a detailed question description (question body). For each question, several community answers are provided, each of which is classified as "*Definitely Relevant*" (*Good*), "Potentially Useful" (*Potential*), or "*Bad*" (*bad*, *dialog*, *non—English*, *other*). Following the strategy as used in previous CQA studies [30,44], both the "*Potentially Useful*" and "*Bad*" *are considered as* "*Bad*" *in the our experiments since the* "Potentially Useful" is both the smallest and the noisiest class, which makes it the most difficult samples to predict. Some metadata is also provided for each question, such as the internal identifier for the user who posted the question, the date of posting, the question category according to the Qatar Living taxonomy, and the type of question.

In all experiments, data preprocessing is performed. The texts are tokenized with a natural language toolkit NLTK.[2] Then, we remove non-alphabet characters, numbers, pronoun, and punctuation from the text. To reduce data sparsity, all tokens were transformed to lowercase letters, and all but the 20,000 most frequent tokens were replaced with a generic "*UNK*" token.

5.2. Baseline methods

In the experiments, we evaluate and compare our model with several strong baseline methods, which we describe below:

- **JAIST** [28]: This method investigated various features (16 features belong to 5 groups) such as topic model based features, and the SVM classifier was then used to predict the answer quality. It was the top performer of the SemEval-2015 Task 3.
- **KeLP** [30]: It used three kinds of features, including linguistic similarities between texts, syntactic trees, and task-specific information. This model was the winner of the SemEval-2016 and SemEval-2017 Task 3.
- **CNN** [59]: It used the traditional features and a convolutional neural network to represent the question and answer representations.
- **LSTM** [60]: This model was implemented with a simple LSTM network. The question subject and the question descriptions were concatenated to represent the question representation.
- **Bi-LSTM-attention** [60]: A Bi-LSTM network followed by an attention mechanism was used to learn the question and answer representations.

[1] http://www.qatarliving.com/forum.

[2] http://www.nltk.org.

Table 4
Statistics of the three CQA datasets.

Method	SemEval-2015			SemEval-2016			SemEval-2017		
	Train	Dev	Test	Train	Dev	Test	Train	Dev	Test
# of ques.	2376	266	300	4879	244	327	5124	327	293
# of ans.	15,013	1447	1793	36,198	2440	3270	38,638	3270	2930
Avg. len. of sub.	6.36	6.08	6.24	6.40	6.04	6.17	6.38	6.16	5.76
Avg. len. of body	39.26	39.47	39.53	43.29	46.88	49.77	43.01	47.98	54.06
Avg. len. of ans.	35.82	33.90	37.33	37.76	36.18	37.27	37.67	37.30	39.50

- **BGMN** [61]: It applied the memory network to iteratively process the questions and answers, which helped identify the relationship between questions and answers.
- **CNN–LSTM–CRF** [40]: The question and the corresponding answers were encoded by the CNN, and then the representations of the question and answers were connected sequentially. We applied the LSTM with attention mechanism and a Conditional Random Fields (CRF) layer on the top of LSTM to encode the sequence.
- **MTL** [35]: A deep neural network (i.e., CNN) was trained jointly on three CQA tasks and learned to encode questions and answers into a single vector representation shared across the three tasks.
- **A-ARC** [45]: An attentive deep neural network was proposed to learn the deterministic information for answer selection from a global perspective, which combined the strengths of CNN, attention-based LSTM and Conditional Random Fields (CRF).
- **RCNN** [38]: A recurrent convolutional neural network was proposed for answer selection in CQA. The representations of question/answer were learned with a CNN, and then the sequence of QA pair representations were fed into the RNNs to capture the semantic correlations among answers.
- **AP-LSTM** [42]: The attentive pooling LSTM network designed a two-way attention mechanism to project the paired input sequences into a common representation space for better answer ranking.
- **AI-CNN** [44]: It proposed an attentive interactive based convolutional neural network to capture the important information of both questions and answers for answer selection task.

5.3. Evaluation metrics

We adopt the official evaluation metrics that are widely used in previous work [42,44]. For the SemEval-2015 dataset, the official evaluation metrics are accuracy and macro-averaged F1 score over three categories (*Good*, *Potential*, and *Bad*). However, following the setting in many recent studies dos Santos et al. [42]; Zhang et al. [44], we switch the three-class classification setting to a binary classification setting that identifies *Good* and *Bad* answers. Because binary classification is much closer to a real-world CQA application. Therefore, we adopt the accuracy and macro-averaged F1 score on two categories for evaluation. In addition, to comprehensively evaluate the proposed model, we also report the precision and recall scores. To be specific, accuracy measures the percentage of correct predicted samples in all samples:

$$Accuracy = -\frac{T}{N} \tag{47}$$

where T is the number of correctly predicted samples, N is the total number of samples. Generally, a well performed system has a higher accuracy.

Precision is the ratio of correctly predicted positive samples to the total predicted positive samples; Recall is the ratio of correctly predicted positive samples to the all samples in positive class; Macro F1 score is the weighted average of Precision and Recall, which takes both false positives and false negatives into account.

$$F_1 = \frac{2 \cdot \text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \tag{48}$$

For the SemEval-2016 and SemEval-2017 datasets, three evaluation metrics are used in the evaluation, including a ranking metric, Mean Average Precision (MAP), and the classification metrics: accuracy, precision, recall and Macro F1 score. MAP is a very popular performance measure in information retrieval (ranking task), considering the order in which the returned documents are presented. Precision and recall are single-value metrics based on the whole list of documents returned by the system. For the model that returns a ranked list of documents, it is desirable to also consider the order in which the returned documents are presented. Average precision (AP) computes the average value of the area under the precision–recall curve. This integral is in practice replaced with a finite sum over every position in the ranked sequence of documents. Mean average precision for a set of queries is the mean of the average precision scores for each query.

$$MAP = \frac{\sum_{q=1}^{\mathcal{Q}} AP(q)}{\mathcal{Q}} \tag{49}$$

where $\mathcal{Q}$ is the number of queries.

5.4. Implementation details

We use a subset of Freebase (FB5M3) as our KB, which includes 4,904,397 entities, 7523 relations, and 22,441,880 facts. The number of entity candidates K is set to 5. We use 100-dimensional word2vec [62] trained on English Wikipedia corpus to initialize the word embeddings in the three datasets, and the out-of-vocabulary words are initialized to zero vector. The graph embeddings are initialized by randomly sampling from the normal distribution $\mathcal{N}(0, 1)$, and the size of the embeddings is set to 100. These embeddings are fine-tuned during training procedure. For the convenience of calculation, we set both the number of hidden states of LSTM and the number of feature maps of CNN to 200. The width of the convolution filters is set to be 2. The weight parameters are randomly sampled from the uniform distribution $U(-0.01,0.01)$, and the bias parameters are set to zero. We conduct mini-batch training (batch size $= 64$) using Adadelta optimization method to train the model which follows the suggested parameter setup in [63]. The learning rate is set to 1×10^{-4}. L_2 regularization (with a weight decay value of 0.001) and dropout (with a dropout rate of 0.2) are used on the output layer to avoid overfitting.

6. Experimental results

In this section, we compare our model with baseline methods on the three datasets from both quantitative and qualitative perspectives.

Table 5
Quantitative evaluation results on SemEval-2015.

Method	Accuracy	Precision	Recall	F1 score
JAIST	79.1	79.48	78.45	78.96
KeLP	81.96	81.43	80.04	80.73
CNN	77.33	76.84	77.00	76.92
LSTM	76.21	76.32	74.02	75.15
Bi-LSTM-attention	81.12	79.48	78.70	79.09
BGMN	81.24	79.45	81.01	80.22
CNN–LSTM–CRF	82.15	82.25	80.43	81.33
MTL	79.12	77.95	78.57	78.26
A-ARC	82.95	82.46	81.82	82.14
RCNN	81.68	80.87	82.18	81.52
AP-LSTM	79.45	80.45	77.71	79.06
AI-CNN	83.06	81.25	82.60	81.92
MKMIA-CQA	**86.78***	**85.35***	**84.36***	**84.85***

* Numbers mean that improvement from our model is statistically significant over the baseline methods (t-test, p-value the < 0.05).

Table 6
Quantitative evaluation results on SemEval-2016.

Method	Accuracy	Precision	Recall	F1 score	MAP
JAIST	74.50	61.54	62.79	62.16	77.56
Kelp	75.11	64.93	63.80	64.36	79.19
CNN	73.23	65.43	64.41	64.92	76.21
LSTM	72.84	64.09	64.21	64.15	76.08
Bi-LSTM-attention	74.55	70.37	69.28	69.82	77.31
BGMN	74.06	69.32	69.46	69.39	77.10
CNN–LSTM–CRF	75.18	71.32	68.81	70.04	77.45
MTL	73.94	67.45	63.64	65.49	76.43
A-ARC	75.86	71.98	72.50	72.24	78.55
RCNN	75.65	71.32	72.36	71.84	78.44
AP-LSTM	75.47	70.45	73.04	71.72	77.12
AI-CNN	76.30	72.78	72.72	72.75	79.17
MKMIA-CQA	**78.46***	**75.44***	**73.30**	**74.35***	**81.27***

* Numbers mean that improvement from our model is statistically significant over the baseline methods (t-test, p-value < 0.05).

Table 7
Quantitative evaluation results on SemEval-2017.

Method	Accuracy	Precision	Recall	F1 score	MAP
JAIST	73.78	66.73	69.40	68.04	87.24
Kelp	73.89	70.42	69.33	69.87	88.43
CNN	73.22	72.58	71.71	72.14	86.21
LSTM	74.05	74.24	72.68	73.45	86.28
Bi-LSTM-attention	76.60	75.67	73.99	74.82	88.05
BGMN	74.75	76.25	74.55	75.39	87.68
CNN–LSTM–CRF	77.18	76.49	77.60	77.04	87.66
MTL	74.18	73.45	72.22	72.83	87.32
A-ARC	77.86	77.38	76.52	76.95	88.06
RCNN	77.14	77.08	75.59	7676.33	87.80
AP-LSTM	77.64	76.65	76.99	76.82	87.82
AI-CNN	78.24	78.23	77.28	77.75	88.33
MKMIA-CQA	**81.56***	**80.30**	**79.27***	**79.78***	**88.93**

* Numbers mean that improvement from our model is statistically significant over the baseline methods (t-test, p-value < 0.05).

6.1. Quantitative evaluation

Tables 5–7 show the performance comparison of MKMIA-CQA with other baseline methods on the official SemEval-2015, SemEval-2016 and SemEval-2017 datasets, respectively. Our model achieves substantially better or competitive performance compared to the state-of-the-art competitors on the three datasets. Concretely, standard CNN and LSTM perform poorly since they pay identical attention to each word. The performance of CQA would be hindered by the noises in the long text. The JAIST model performs better than standard LSTM and CNN. The advantage of JAIST is that it uses various extra features. Bi-LSTM-attention, CNN–LSTM–CRF, and AP-LSTM stably outperform LSTM by a noticeable margin since they add attention mechanism to capture important information and filter out the noisy information in the long text. Further, AI-CNN exceeds other methods because it identifies the relationship and interactions between questions and answers. **Our model** takes a further step towards incorporating the external commonsense knowledge from KB by proposing a multi-head interactive attention mechanism. We can observe that for the accuracy the proposed MKMIA-CQA method achieves statistically significantly better performance than the compared methods (improve 4.47% on SemEval-2015, 2.83% on SemEval-2016, 4.24% on SemEval-2017) and gets state-of-the-art on all the three datasets. As we know, it is difficult to boost 1 percent of accuracy on CQA.

Table 8
Ablation test of MKMIA-CQA on SemEval-2015.

Method	Accuracy	Precision	Recall	F1 score
MKMIA-CQA	**86.78**	**85.35**	**84.36**	**84.85**
w/o knowledge	84.03	83.15	81.33	82.23
w/o text	84.64	83.84	82.67	83.25
w/o IA	85.46	84.33	83.63	83.98
w/o MA	86.27	84.07	84.57	84.32

Table 9
Ablation test of MKMIA-CQA on SemEval-2016.

Method	Accuracy	Precision	Recall	F1 score	MAP
MKMIA-CQA	**78.46**	**75.44**	**73.30**	**74.35**	**81.27**
w/o knowledge	76.42	73.58	72.55	73.06	79.83
w/o text	77.15	74.01	73.14	73.57	80.35
w/o IA	77.22	73.06	73.44	73.25	80.07
w/o MA	77.96	73.89	74.19	74.04	80.93

6.2. Ablation study

In order to analyze the effectiveness of each component of MKMIA-CQA, we also report the ablation test of MKMIA-CQA in terms of discarding external knowledge from KB (w/o knowledge), text classification task (w/o text), interactive attention (w/o IA), and multi-head attention (w/o MA), respectively. For the model without interactive attention (i.e., w/o IA), we simply concatenate the initial context and knowledge representations of the texts. For the model without multi-head attention (i.e., w/o MA), we use single-head attention to take the place of multi-head attention.

The ablation results are summarized in Tables 8–10 for the three experimental datasets. Generally, all three factors contribute great improvement to MKMIA-CQA. From the results, we can observe that the accuracy and F1 score decrease sharply when discarding the external knowledge in KB. This is within our expectation since KB introduces commonsense background knowledge beyond the context to enrich overall text representations and focus on useful information. In addition, text classification task also contributes to the effectiveness of MKMIA-CQA. This verifies that the auxiliary task helps to comprehend the document semantics and locate the salient information of the input with respect to specific text category. Not surprisingly, combining all components achieves the best performance for all evaluation metrics

6.3. Qualitative evaluation

MKMIA-CQA provides an intuitive way to inspect the soft-alignment between the question and the answers by visualizing the question attention score and the answer attention score from Eqs. (36)–(37), respectively. Due to the space limitation, we take one question and its corresponding answer from SemEval-2015 dataset as an example, and visualize the attention scores

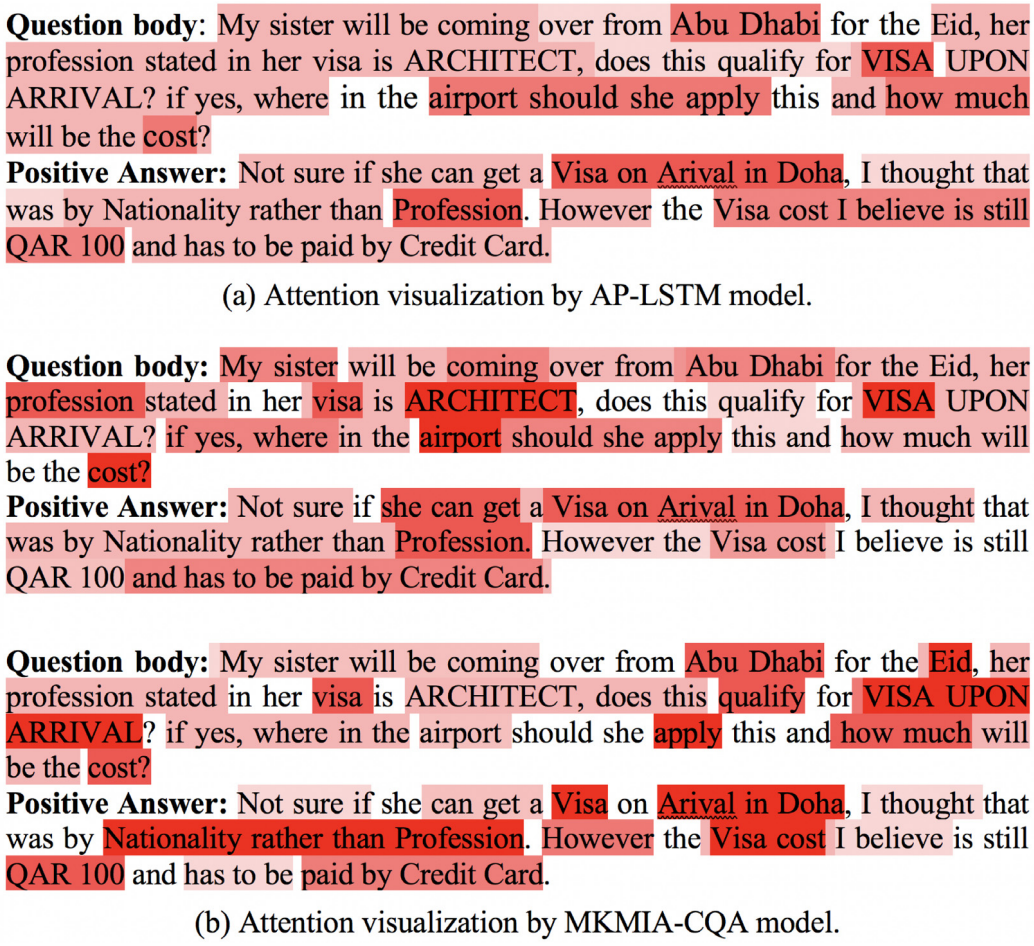

Fig. 2. The attention weights for a question and a positive answer by (a) AP-LSTM and (b) MKMIA-CQA (with the number of attention hops is 2, i.e., $b = 2$).

Table 10
Ablation test of MKMIA-CQA on SemEval-2017.

Method	Accuracy	Precision	Recall	F1 score	MAP
MKMIA-CQA	**81.56**	**80.30**	**79.27**	**79.78**	**88.93**
w/o knowledge	79.22	78.05	77.91	77.98	87.61
w/o text	80.15	78.68	77.98	78.33	88.35
w/o IA	80.76	78.73	79.51	79.12	88.13
w/o MA	81.24	79.15	78.73	78.94	88.47

predicted by AP-LSTM and MKMIA-CQA in Fig. 2. The color depth indicates the importance degree of the words. The darker the color, the more important the word. From Fig. 2, we observe that AP-LSTM pays much attention to those words that are contextually related to the question, such as "*visa*", but neglecting the commonsense knowledge beyond the context of the question such as "*Nationality*". This limitation can be alleviated decently by knowledge-aware attention mechanism. MKMIA-CQA detect that the words "*VISA UPON ARRIVAL*" plays a dominant role in the question attention vector for CQA. Similarly, the words "*Nationality rather than Profession*" achieves a higher value than other words in the answer attention vector. This verifies that our model has the capability of capturing the distinguishable features.

6.4. Error analysis

We randomly select 100 samples that are incorrectly predicted by our model from the SemEval-2017 test set. We observe that these samples are imbalanced across categorical domains. As shown in Table 11, some question types such as "Family Life in Qatar" and "Moving to Qatar" are more difficult than other types such as "Doha Shopping" and "Cars and driving".

Table 11
Incorrectly predicted samples statistics.

Question category	# of incorrect samples.
Socializing	17
Advice and help	12
Opportunities	12
Welcome to Qatar	11
Doha Shopping	9
Family Life in Qatar	8
Funnies	8
Qatar living lounge	6
Salary and allowances	5
Working in Qatar	5
Investment and finance	5
Moving to Qatar	2

6.5. Case study for "How to" questions

In this section, we empirically demonstrate that our model can effectively deal with the "how-to" question that is one of the most challenging questions in CQA. In total, there are 13 (67 QA pairs), 10 (100 QA pairs), 6 (60 QA pairs) "how to" questions in test sets of SemEval-2015/2016/2017 datasets, respectively. MKMIA-CQA correctly predicted 53, 85, and 48 QA pairs in the three datasets. In Table 12, we list a "how to" question from SemEval-2017 that is incorrectly predicted by AI-CNN (the

Table 12
Example of a "how to" question and its two answers from SemEval-2017.

Question	"How to go home without exit permit. My sponsor is not willing to give me exit permit in the fear of my not return. I am about 3 years at Qatar; and I want to go home on a vacation before Eid. Is there any way to go home without the exit permit from sponsor? I want to go home before Eid; no matter what does it cost; even cancellation of my visa. Thank you for your advice".
Positive answer	"You have 7 days... You left it a bit late to resign; no? If you are due to go home they have to give you a ticket. If not; then go to the Labour Department and report him".
Negative answer	"My sponsor is not willing to let me go in any way until all of my projects are finished. Can my embassy or the labor department help?"

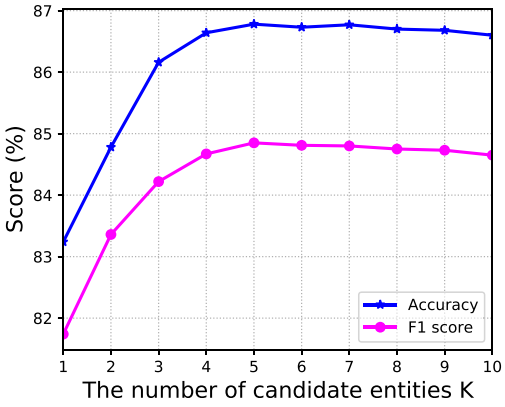

Fig. 3. Experimental results of MKMIA-CQA on SemEval-2015 by varying the value of the number of candidate entities K.

strongest baseline) but correctly predicted by MKMIA-CQA. We observe that AI-CNN tends to assign a higher score to the negative answer than the positive answer, since the negative answer is more similar to the given question at the word level. However, with the background knowledge from KB, MKMIA-CQA can correctly identify the positive answer based on the relative facts contained in the KB such as (*labor, related_to, work*) and (*permit, related_to, ticket_of_leave*).

6.6. The effect of parameter K

K is the number of entity candidates from KB for each entity mention in the input document. In this paper, we investigate the effect of K by varying its value from 1 to 10 with step size 1. We report the experimental results on SemEval-2015 dataset in Fig. 3. We can achieve best results when $K = 5$ on SemEval-2015. As K increases from 1 to 10, the accuracy and F1 scores increase sharply till an optimal value (when $K = 5$), after which the results decrease slightly.

7. Conclusion and future work

In this paper, we described a novel community question answering model, called MKMIA-CQA, which leveraged text categorization to improve the performance of CQA. The text categorization task helped improve the quality of locating salient information of the text. In addition, we exploited the external commonsense knowledge from KB to capture the important entities and their relations, and reduce the effects of redundant and noisy information by proposing a multi-head interactive attention mechanism. We conducted extensive experiments on three benchmark datasets to evaluate the effectiveness of MKMIA-CQA. Quantitatively, the experimental results demonstrated that our model outperformed all the compared methods. We also conducted ablation test to analyze the effectiveness of each component of MKMIA-CQA and the results showed that all the investigated factors contributed a great improvement to MKMIA-CQA. Qualitatively, by visualizing the attention scores of the question and the answer, we observed that MKMIA-CQA could assign higher attention scores to the informative words than the meaningless words.

Despite the remarkable progress of recent studies, human-like reading strategy, which plays crucial roles in reading comprehension, has received little attention in recent deep learning based methods that achieve the-state-of-the-art results in community question answering. When humans read and comprehend a piece of text, their exploration of the reading process organizes itself most naturally into an examination of three phases: pre-reading, active reading (i.e., task-specific reading comprehension), and post-reading. Motivated by the process of the human reading cognition that follows a hierarchical routine, in the future, we may devote our effort to design a hierarchical human-like attention network for community question answering, whose major components are consistent with the stages of the human reading cognitive process (i.e., pre-reading, active reading, and post-reading). In addition, we also plan to exploit other external knowledge like the synonyms in WordNet that helps provide more comprehensive information for text matching.

Acknowledgments

This work was also partially supported by the National Natural Science Foundation of China (Grant No. 61803249), the Shanghai Sailing Program, China (Grant No. 18YF1407700), the SIAT Innovation Program for Excellent Young Researchers, China (Grant No. Y8G027), and the CAS Pioneer Hundred Talents Program, China (Grant No. Y84402). Min Yang was sponsored by CCF-Tencent Open Research Fund, China.

References

[1] Y. Yang, W.T. Yih, C. Meek, Wikiqa: A challenge dataset for open-domain question answering, in: EMNLP, 2015, pp. 2013–2018.
[2] M. Feng, B. Xiang, M.R. Glass, L. Wang, B. Zhou, Applying Deep Learning to Answer Selection: A Study and an Open Task, 2015, pp. 813–820.
[3] B. Patra, A survey of Community Question Answering, arXiv preprint arXiv:1705.04009, 2017.
[4] A. Severyn, A. Moschitti, Learning to rank short text pairs with convolutional deep neural networks, in: SIGIR, 2015, pp. 373–382.
[5] D. Wang, E. Nyberg, A long short-term memory model for answer sentence selection in question answering, in: Meeting of the Association for Computational Linguistics, 2015, pp. 707–712.
[6] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, J. Taylor, Freebase: a collaboratively created graph database for structuring human knowledge, in: SIGMOD, ACM, 2008, pp. 1247–1250.
[7] R. Navigli, S.P. Ponzetto, Babelnet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network, Artificial Intelligence 193 (2012) 217–250.
[8] G. Weikum, G. Weikum, G. Weikum, Yago: a core of semantic knowledge, in: International Conference on World Wide Web, 2007, pp. 697–706.

[9] J. Lehmann, R. Isele, M. Jakob, A. Jentzsch, D. Kontokostas, P.N. Mendes, S. Hellmann, M. Morsey, P. Van Kleef, S. Auer, et al., DBpedia–a Large-scale, multilingual knowledge base extracted from wikipedia, Semant. Web 6 (2) (2015) 167–195.
[10] R. Speer, J. Chin, C. Havasi, ConceptNet 5.5: An open multilingual graph of general knowledge, in: AAAI, 2017, pp. 4444–4451.
[11] B. Perozzi, R. Al-Rfou, S. Skiena, DeepWalk: online learning of social representations, in: SIGKDD, 2014, pp. 701–710.
[12] H. Cui, R. Sun, K. Li, M.-Y. Kan, T.-S. Chua, Question answering passage retrieval using dependency relations, in: Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, 2005, pp. 400–407.
[13] M. Wang, N.A. Smith, T. Mitamura, What is the Jeopardy model? A quasi-synchronous grammar for qa, in: Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL), 2007.
[14] M. Heilman, N.A. Smith, Tree edit models for recognizing textual entailments, paraphrases, and answers to questions, in: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technologies, Association for Computational Linguistics, 2010, pp. 1011–1019.
[15] M. Surdeanu, M. Ciaramita, H. Zaragoza, Learning to rank answers on large online QA collections, in: Proceedings of ACL-08: HLT, 2008, pp. 719–727.
[16] J. Jeon, W.B. Croft, J.H. Lee, Finding similar questions in large question and answer archives, in: Proceedings of the 14th ACM International Conference on Information and Knowledge Management, ACM, 2005, pp. 84–90.
[17] X. Xue, J. Jeon, W.B. Croft, Retrieval models for question and answer archives, in: Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, 2008, pp. 475–482.
[18] J.-T. Lee, S.-B. Kim, Y.-I. Song, H.-C. Rim, Bridging lexical gaps between queries and questions on large online q&a collections with compact translation models, in: Proceedings of the Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, 2008, pp. 410–418.
[19] G. Zhou, L. Cai, J. Zhao, K. Liu, Phrase-based translation model for question retrieval in community question answer archives, in: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1, Association for Computational Linguistics, 2011, pp. 653–662.
[20] T. Hofmann, Unsupervised learning by probabilistic latent semantic analysis, Mach. Learn. 42 (1–2) (2001) 177–196.
[21] D.M. Blei, A.Y. Ng, M.I. Jordan, Latent dirichlet allocation, J. Mach. Learn. Res. 3 (Jan) (2003) 993–1022.
[22] L. Cai, G. Zhou, K. Liu, J. Zhao, Learning the latent topics for question retrieval in community qa, in: Proceedings of 5th international joint conference on Natural Language Processing, 2011, pp. 273–281.
[23] Z. Ji, F. Xu, B. Wang, B. He, Question-answer topic model for question retrieval in community question answering, in: Proceedings of the 21st ACM International Conference on Information and Knowledge Management, ACM, 2012, pp. 2471–2474.
[24] M. Chen, L. Li, Q. Xie, Translation language model enhancement for community question retrieval using user adoption answer, in: Asia-Pacific Web (APWeb) and Web-Age Information Management (WAIM) Joint Conference on Web and Big Data, Springer, 2017, pp. 251–265.
[25] P. Nakov, L. Màrquez, W. Magdy, A. Moschitti, J. Glass, B. Randeree, SemEval-2015 task 3: Answer selection in community question answering, in: Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015), 2015, pp. 269–281.
[26] P. Nakov, L. Màrquez, A. Moschitti, W. Magdy, H. Mubarak, A.A. Freihat, SemEval-2016 task 3: Community question answering, in: Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2016), 2016, pp. 525–545.
[27] P. Nakov, D. Hoogeveen, L. Màrquez, A. Moschitti, H. Mubarak, T. Baldwin, K. Verspoor, SemEval-2017 task 3: Community question answering, in: Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017), 2017, pp. 27–48.
[28] Q.H. Tran, V. Tran, T. Vu, M. Nguyen, S.B. Pham, Jaist: Combining multiple features for answer selection in community question answering, in: Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015), 2015, pp. 215–219.
[29] Y. Hou, C. Tan, X. Wang, Y. Zhang, J. Xu, Q. Chen, HITSZ-ICRC: Exploiting classification approach for answer selection in community question answering, in: Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015), 2015, pp. 196–202.
[30] S. Filice, D. Croce, A. Moschitti, R. Basili, Kelp at semeval-2016 task 3: Learning semantic relations between questions and answers, in: Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016), 2016, pp. 1116–1123.
[31] S. Filice, G. Da San Martino, A. Moschitti, KeLP at SemEval-2017 Task 3: Learning pairwise patterns in community question answering, in: Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017), 2017, pp. 326–333.
[32] W. Feng, Y. Wu, W. Wu, Z. Li, M. Zhou, Beihang-MSRA at SemEval-2017 Task 3: A ranking system with neural matching features for community question answering, in: Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017), 2017, pp. 280–286.
[33] M. Iyyer, J. Boyd-Graber, L. Claudino, R. Socher, H. Daumé III, A neural network for factoid question answering over paragraphs, in: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2014, pp. 633–644.
[34] D. Wang, E. Nyberg, A long short-term memory model for answer sentence selection in question answering, in: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers), Vol. 2, 2015, pp. 707–712.
[35] D. Bonadiman, A. Uva, A. Moschitti, Multitask learning with deep neural networks for community question answering, arXiv preprint arXiv:1702.03706, 2017.
[36] J. Guo, B. Yue, G. Xu, Z. Yang, J.-M. Wei, An enhanced convolutional neural network model for answer selection, in: Proceedings of the 26th International Conference on World Wide Web Companion, International World Wide Web Conferences Steering Committee, 2017, pp. 789–790.
[37] X. Zhou, B. Hu, Q. Chen, B. Tang, X. Wang, Answer sequence learning with neural networks for answer selection in community question answering, in: The 53rd Annual Meeting of the Association for Computational Linguistics, ACL, 2015, pp. 713–718.
[38] X. Zhou, B. Hu, Q. Chen, X. Wang, Recurrent convolutional neural network for answer selection in community question answering, Neurocomputing 274 (2018) 8–18.
[39] X. Qiu, X. Huang, Convolutional neural tensor network architecture for community-based question answering, in: IJCAI, 2015, pp. 1305–1311.
[40] Y. Xiang, X. Zhou, Q. Chen, Z. Zheng, B. Tang, X. Wang, Y. 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, 2016, pp. 1231–1241.
[41] M. Tan, C.D. Santos, B. Xiang, B. Zhou, Improved representation learning for question answer matching, in: Meeting of the Association for Computational Linguistics, 2016, pp. 464–473.
[42] C.N. dos Santos, M. Tan, B. Xiang, B. Zhou, Attentive pooling networks, CoRR, abs/1602.03609, 2016.
[43] Q. Chen, Q. Hu, J.X. Huang, L. He, W. An, Enhancing recurrent neural networks with positional attention for question answering, in: SIGIR, ACM, 2017, pp. 993–996.
[44] X. Zhang, S. Li, L. Sha, H. Wang, Attentive interactive neural networks for answer selection in community question answering, in: AAAI, 2017, pp. 3525–3531.
[45] Y. Xiang, Q. Chen, X. Wang, Y. Qin, Answer selection in community question answering via attentive neural networks, IEEE Signal Process. Lett. 24 (4) (2017) 505–509.
[46] W. Wu, S. Xu, W. Houfeng, 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), Vol. 1, 2018, pp. 1746–1755.
[47] Y. Wu, W. Wu, Z. Li, M. Zhou, Knowledge enhanced hybrid neural network for text matching, in: AAAI, 2016.
[48] B. Yang, T. Mitchell, Leveraging knowledge bases in lstms for improving machine reading, in: Meeting of the Association for Computational Linguistics, Vol. 1, 2017, pp. 1436–1446.
[49] J. Xin, Y. Lin, Z. Liu, M. Sun, Improving neural fine-grained entity typing with knowledge attention, in: AAAI, 2018.
[50] X. Han, Z. Liu, M. Sun, Neural knowledge acquisition via mutual attention between knowledge graph and text, in: AAAI, 2018.
[51] A. Bordes, N. Usunier, J. Weston, O. Yakhnenko, Translating embeddings for modeling multi-relational data, in: NIPS, 2013, pp. 2787–2795.
[52] Z. Wang, J. Zhang, J. Feng, Z. Chen, Knowledge graph embedding by translating on hyperplanes, in: AAAI, Vol. 14, 2014, pp. 1112–1119.
[53] R. Caruana, Multitask learning, in: Learning to Learn, Springer, 1998, pp. 95–133.
[54] R. Collobert, J. Weston, A unified architecture for natural language processing: Deep neural networks with multitask learning, in: Proceedings of the 25th International Conference on Machine Learning, ACM, 2008, pp. 160–167.
[55] P. Liu, X. Qiu, X. Huang, Recurrent neural network for text classification with multi-task learning, in: Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, 2016, pp. 2873–2879.

[56] X. Liu, J. Gao, X. He, L. Deng, K. Duh, Y.-Y. Wang, Representation learning using multi-task deep neural networks for semantic classification and information retrieval, in: The 2015 Annual Conference of the North American Chapter of the ACL, 2015, pp. 912–921.
[57] M.-T. Luong, Q.V. Le, I. Sutskever, O. Vinyals, L. Kaiser, Multi-task sequence to sequence learning, in: International Conference on Learning Representations, 2016.
[58] Y. Luan, C. Brockett, B. Dolan, J. Gao, M. Galley, Multi-Task learning for speaker-role adaptation in neural conversation Models, in: Proceedings of the The 8th International Joint Conference on Natural Language Processing, 2017, pp. 605–614.
[59] L. Yu, K.M. Hermann, P. Blunsom, S. Pulman, Deep learning for answer sentence selection, in: Proceedings of Deep Learning and Representation Learning Workshop, NIPS, 2014.
[60] M. Tan, C.d. Santos, B. Xiang, B. Zhou, Lstm-based deep learning models for non-factoid answer selection, in: International Conference on Learning Representations, 2016.
[61] G. Wu, Y. Sheng, M. Lan, Y. Wu, Ecnu at semeval-2017 task 3: Using traditional and deep learning methods to address community question answering task, in: Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017), 2017, pp. 365–369.
[62] T. Mikolov, K. Chen, G. Corrado, J. Dean, Efficient estimation of word representations in vector space, in: International Conference on Learning Representations, 2013, pp. 1–12.
[63] M.D. Zeiler, ADADELTA: an adaptive learning rate method, arXiv preprint arXiv:1212.5701, 2012.