Knowledge-enhanced Hierarchical Attention for Community Question Answering with Multi-task and Adaptive Learning

Min Yang[1] , **Lei Chen**[1,2] , **Xiaojun Chen**[3] , **Qingyao Wu**[4] , **Wei Zhou**[5] , **Ying Shen**[6*]
[1] Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences
[2] University of Chinese Academy of Sciences
[3]Shenzhen University
[4]South China University of Technology
[5]Chongqing University
[6] Peking University Shenzhen Graduate School
{min.yang, lei.chen}@siat.ac.cn, xjchen@zju.edu.cn, qyw@scut.edu.cn, zhouwei@cqu.edu.cn,
shenying@pkusz.edu.cn

Abstract

In this paper, we propose a Knowledge-enhanced Hierarchical Attention for community question answering with Multi-task learning and Adaptive learning (KHAMA). First, we propose a hierarchical attention network to fully fuse knowledge from input documents and knowledge base (KB) by exploiting the semantic compositionality of the input sequences. The external factual knowledge helps recognize background knowledge (entity mentions and their relationships) and eliminate noise information from long documents that have sophisticated syntactic and semantic structures. In addition, we build multiple CQA models with adaptive boosting and then combine these models to learn a more effective and robust CQA system. Furthermore, KHAMA is a multi-task learning model. It regards CQA as the primary task and question categorization as the auxiliary task, aiming at learning a category-aware document encoder and enhance the quality of identifying essential information from long questions. Extensive experiments on two benchmarks demonstrate that KHAMA achieves substantial improvements over the compared methods.

1 Introduction

Community question answering (CQA) websites, e.g., Stack Overflow and Quora have attracted increasing attention for people to get free advice directly from experienced users. With the influx of new questions and the varied quality of provided answers, it is time-consuming for users to inspect them all. Therefore, developing automated methods to identify valuable answers for a given question is of practical importance.

Earlier efforts for community question answering pay particular attention to designing various features, such as frequency features and translation features, to rank the answers. Nevertheless, these approaches are built on plenty of hand-crafted features and their performance easily stagnates. Recently, many deep learning techniques, such as long short-term memory (LSTM) and convolutional neural network (CNN) have been employed for answer selection in CQA. These methods can capture the semantics of documents and avoid labor-intensive feature engineering. Instead of encoding the question and answer representations separately, attention mechanisms have been applied to capture the correlations between the question and answer, and give more weights on relevant parts of the input for better answer selection.

Despite the successes achieved by prior work, we argue that CQA in real-world remains challenging for several reasons. **First**, the external factual knowledge in knowledge bases (KBs), such as Freebase [Bollacker *et al.*, 2008], provides rich information of entities and relations between them, and highlights the features that are essential for CQA. Despite its usefulness, the external knowledge from KBs is underutilized in recent deep learning based CQA systems. **Second**, the principle of semantic compositionality of a document is vital for text comprehension. To that end, a CQA model should focus on different levels of semantics of the input sequences and acquire comprehensive information to identify suitable answers. However, most existing methods do not exploit the semantic compositionality when learning the document representation. **Third**, a CQA system that does not identify the categories of input questions may learn a document encoder losing the crucial and discriminative features in questions. **Forth**, one of the most challenging issues of existing deep neural networks is that their performance may not be stable and can not effectively handle the quite imbalanced and noisy data in CQA.

In this paper, the mentioned challenges are considered and alleviated to some extent. We propose a Knowledge-enhanced Hierarchical Attention for community question answering with multi-task learning and adaptive learning. **First**, we propose a knowledge-enhanced hierarchical attention mechanism to fully explore the knowledge from input text documents and KB at different levels of granular-

[*] Ying Shen is corresponding author.

ity. In particular, we design a three-stage attention mechanism, including word-level attention, phrase-level attention, document-level attention, to integrate factual knowledge from KB into the representation learning of questions and answers, exploiting the semantic compositionality of the context and knowledge sequences. In addition, a multi-head co-attention network is proposed to capture correlations between the question and answer. **Second**, we propose a multi-task learning framework, in which the knowledge-enhanced representation learning is simultaneously optimized by two coupled tasks: CQA (primary task) and question categorization (auxiliary task). The main purpose of multi-task learning is to improve the quality of locating the salient information. **Third**, for CQA task, we construct multiple classifiers and ensemble their results as the final prediction result that is expected to more effectively and robustly solve the CQA task.

We summarize the main contributions of this paper as follows. (1) We incorporate the external knowledge from KB into deep neural networks to extract important information from the questions/answers that may be noisy and redundant by proposing a knowledge-enhanced hierarchical attention mechanism. (2) We leverage the question categorization to enhance the question representation learning for CQA, which improves the quality of identifying discriminative features in questions. (3) To the best of our knowledge, we are the first to combine adaptive boosting and deep neural networks for CQA task. The ensemble results from multiple models can be more accurate and robust. (4) Experiments on two benchmark CQA datasets show that KHAMA significantly outperforms the state-of-the-art baseline methods.

2 Related Work

Conventional CQA methods focused on designing different kinds of features to enhance the performance of CQA. Surdeanu *et al.* [2008] investigated a variety of features, e.g., frequency features, similarity features, and translation features to rank answers for the given question. Heilman and Smith [2010] applied the logistic regression algorithm and tree kernel function to recognize the correlations between the question and answer. Tran *et al.* [2015] utilized topic model based features to forecast the quality of answers. Although these methods have achieved remarkable success in answer selection, the performance of above methods depends on the hand-crafted features, whose definition is time-consuming.

Recently, deep learning methods make a breakthrough and become the mainstream to tackle the CQA task by encoding questions and answers into continuous vector representations without heavy feature engineering. Qiu and Huang [2015] introduced a convolutional neural tensor network to model the interaction between sentences with tensor layers. Wang and Nyberg [2015] applied the bidirectional LSTM to learn the representations of the question and answer, and the semantic matching scores of QA pairs were then computed. Tay *et al.* [2017] presented Holographic Dual LSTM (HDLSTM) to incorporate holographic representational learning into question answering (QA) semantic matching. Guo *et al.* [2017] introduced a skip CNN to learn the lexical semantic features.

Attention mechanism has been proved to be able to significantly improve experimental results on answer selection task by enhancing the interaction between QA pairs [Dos Santos *et al.*, 2016; Chen *et al.*, 2017; Shen *et al.*, 2018; Yang *et al.*, 2019]. Yin *et al.* [2015] presented an attention-based CNN to incorporate mutual influence between sentences into CNNs. Dos Santos *et al.* [2016] presented attention-pooling (AP), a two-way attention mechanism, to map the question and answer into a common feature representation space so as to capture the comprehensive correlations of QA pairs for semantic matching. Tan *et al.* [2016a] integrated CNN and LSTM to capture the complex semantic relations of questions and answers, which combined the advantages of capturing linguistic information from both networks. Chen *et al.* [2017] proposes a positional attention based RNN model to integrate positional information into attentive representation. Despite the effectiveness of these studies, they exclusively consider context information rather than real-world background knowledge and hidden information beyond the context. Tay *et al.* [2018] proposed a cross temporal recurrent network to control the information retained or discarded over time, in which temporal gates are first learned and then applied to amend the question and answer representations temporally.

3 Our Methodology

3.1 Problem Definition

Given a question q and an answer a, CQA task aims at inferring the label $Y \in \{\text{Good}, \text{Bad}\}$, where $y = \text{Good}$ means that answer a is qualified for question q. We denote the question q and the answer a as $q = [w_1^q, w_2^q, \dots, w_n^q]$ and $a = [w_1^a, w_2^a, \dots, w_m^a]$, where n and m represent the lengths of q and a respectively. Since we leverage question categorization task to improve the performance of CQA, we also assume that there is a category label x for each question q. To prevent conceptual confusion, we use superscripts "q" and "a" to represent the variables that are related to questions and answers, respectively.

3.2 The Overall Architecture

KHAMA consists of five modules. *Knowledge-enhanced representation learning module* contains three key layers: word-level mutual attention layer, phrase-level attention layer, document-level attention layer, motivated by [Lei *et al.*, 2018]. *Interactive question/answer representation module* uses a multi-head co-attention network to model the interaction information between questions and answers. *Text categorization* module predicts a category label to the input question, and the category information is then fed into the question representation to learn a category-specific document representation. In *community question answering* module, we construct multiple CQA classifiers and ensemble their results as the final prediction result to more effectively and robustly solve the CQA problem. KHAMA is trained with the *multi-task learning module*, which regards CQA as primary task and question categorization as auxiliary task. Next, we will introduce each part of our KHAMA model in detail.

3.3 Knowledge-enhanced Representation Learning with Hierarchical Attention

Word-level Mutual Attention

Word Embedding The words which share common components (e.g., prefix, root, suffix) may be potentially related. Therefore, we design a joint word embedding layer to combine the merits of word-level and character-level representations. For the word-level embedding model, we adopt the popular word2vec [Mikolov *et al.*, 2013] embeddings which are widely used in NLP domain. For the character-level embedding model, the ELMo language model [Peters *et al.*, 2018] is used due to its superior performance in a wide range of NLP tasks. Then, each word is represented as a concatenation of the character-level embedding and word-level embedding, resulting in a hybrid word embedding vector $\mathbf{e}_t \in \mathbb{R}^{d_e}$ for each word w_t. Here, d_e denotes the size of the hybrid word embedding. The context representations of question q and answer a thus can be represented as $E^q = [\mathbf{e}^q_1, \dots, \mathbf{e}^q_n]$ and $E^a = [\mathbf{e}^a_1, \dots, \mathbf{e}^a_m]$, respectively.

We conduct entity mention detection by n-gram matching and obtain top-K entity candidates from KB for each entity mention in input documents. The entity embeddings in KB are learned via DeepWalk [Perozzi *et al.*, 2014]. Formally, the candidate entities for the t-th entity mention as $\{ent_{t1}, ent_{t2}, ..., ent_{tK}\} \in \mathbb{R}^{K \times d_{\text{kb}}}$, where d_{kb} is the dimension of the entity embedding in KB. A context-guided attention model is designed to compute the knowledge representation of each entity mention in the document, which is computed as:

$$\tilde{\mathbf{e}}_t = \sum_{i=1}^{K} \tau_{ti} ent_{ti}, \quad \tau_{ti} = \text{softmax}(\nu(ent_{ti}, \mu(E))) \tag{1}$$

where ν is the attention function (i.e., a multilayer perceptron), E represents the context representation of the question or answer. Thus, we get the knowledge representations for question q and answer a, denoted as $\tilde{E}^q = [\tilde{\mathbf{e}}^q_1, \dots, \tilde{\mathbf{e}}^q_n]$ and $\tilde{E}^a = [\tilde{\mathbf{e}}^a_1, \dots, \tilde{\mathbf{e}}^a_m]$, respectively.

After obtaining the entity and word embedding, we design a word-level mutual attention mechanism to identify the relations between context and knowledge representations. Formally, we adopt the dot-product between the context and knowledge representations to calculate the correlation matrix $\mathbf{M}^q$ for question q as:

$$\mathbf{M}^q = (E^q)^T \cdot \tilde{E}^q \in \mathbb{R}^{n \times n} \tag{2}$$

where each element in $\mathbf{M}^q_{i,j}$ refers to the correlation between the i-th element in the context representation and the j-th element in the knowledge representation.

Next, we average the values of each row and each column of $\mathbf{M}^q$ as attention sources to calculate attention vectors $\boldsymbol{\lambda}^q$ and $\tilde{\boldsymbol{\lambda}}^q$ for context and knowledge representations respectively.

$$\boldsymbol{\lambda}^q = \text{softmax}(\frac{\sum_{i=1}^{n} \mathbf{M}[:, i]}{n}); \quad \tilde{\boldsymbol{\lambda}}^q = \text{softmax}(\frac{\sum_{j=1}^{n} \mathbf{M}[j, :]}{n}) \tag{3}$$

Finally, the knowledge-enhanced context representation matrix W^q and context-enhanced knowledge representation matrix $\tilde{W}^q$ can be computed as:

$$W^q = \tanh(U^w(E^q + (\mathbf{I}^q \otimes \boldsymbol{\lambda}^q) \odot E^q)) \tag{4}$$

$$\tilde{W}^q = \tanh(\tilde{U}^w(\tilde{E}^q + (\mathbf{I}^q \otimes \tilde{\boldsymbol{\lambda}}^q) \odot \tilde{E}^q)) \tag{5}$$

where U^w and $\tilde{U}^w$ are word-level projection parameters, $\mathbf{I}^q = [1, \dots, 1]^T$ denotes a d-dimensional all-ones vector, $\mathbf{I}^q \otimes \boldsymbol{\lambda}^q$ denotes the kronecker product operation between $\mathbf{I}^q$ and $\boldsymbol{\lambda}^q$, $\odot$ refers to the element-wise multiplication.

Phrase-level Attention

After obtaining W^q and $\tilde{W}^q$, we adopt n-gram convolution operation to extract local semantic features. The convolution operation involves a filter $\mathbf{K}^q$, which is applied to l continuous words. We assume that the feature maps for context and knowledge representations are P^q and $\tilde{P}^q$.

$$F^q = \tanh(W^q * \mathbf{K}^q + \mathbf{b}) \in \mathbb{R}^{(n-l+1) \times k} \tag{6}$$

$$\tilde{F}^q = \tanh(\tilde{W}^q * \mathbf{K}^q + \mathbf{b}) \in \mathbb{R}^{(n-l+1) \times k} \tag{7}$$

where $\mathbf{b}$ is a bias matrix, represents the convolution operator, k indicates the number of filters

A phrase-level attention mechanism is designed to learn important local n-gram chunks. Mathematically, we formulate the chunk-based attention mechanism as follows:

$$\mathbf{C}^q = \mathit{softmax}((F^q)^T \tilde{F}^q) \tag{8}$$

$$P^q = F^q \odot \{\tilde{F}^q U^p (\mathbf{C}^q)^T\}, \quad \tilde{P}^q = \tilde{F}^q \odot \{F^q \tilde{U}^p (\mathbf{C}^q)^T\} \tag{9}$$

where $\mathbf{C}^q$ is the correlation matrix between the context word chunks and the knowledge entities chunks. $\odot$ denotes element-wise multiplication. U^p and $\tilde{U}^p$ are parameters to be learned for phrase-level attention. P^q and $\tilde{P}^q$ refer to the phrase-level knowledge-enhanced context representation and context-enhanced knowledge representation.

Next, we employ two independent LSTM networks to encode hidden states of the context and knowledge representations as $H^q = \text{LSTM}(P^q)$ and $\tilde{H}^q = \text{LSTM}(\tilde{P}^q)$, respectively.

Document-level Attention

We use knowledge representation as attention source to attend to the context so as to select those crucial knowledge-enhanced context word chunks to compose the knowledge-aware document representation. Formally, document-level attention is defined as follows:

$$O^q_i = \alpha_i H^q_i, \quad \alpha_i = \frac{\exp(\rho([H^q_i; \mu(\tilde{H}^q)]))}{\sum_{j=1}^{k} \exp(\rho([H^q_j; \mu(\tilde{H}^q)]))} \tag{10}$$

$$\rho([H^q_i; \mu(\tilde{H}^q)]) = U^d_2 \tanh(U^d_1 [H^q_i; \mu(\tilde{H}^q)]) \tag{11}$$

where μ is mean operation, ρ is the attention function, U^d_1 and U^d_2 are document-level parameters, O^q is the document-level knowledge-enhanced representation for question q.

In the same way, we can get the knowledge-enhanced answer representation O^a.

3.4 Interactive Question/Answer Representation Learning

We propose a multi-head co-attention (MC) network to learn the interaction information between questions and answers. MC learns a 2-dimensional attention matrix. Given the knowledge-enhanced question and answer representations (i.e., O^q and O^a), the attention matrix Σ^q for the answer-aware question representation is computed as:

$$\Sigma^q = [\Sigma^q_1, \cdots, \Sigma^q_k], \quad \Sigma^q_i = \frac{\exp(\delta([O^q_i; \mu(O^a)]))}{\sum_{j=1}^{k} \exp(\delta([O^q_j; \mu(O^a)]))} \quad (12)$$

$$\delta([O^q_i; \mu(O^a)]) = U^d_3 \tanh(U^d_4 [O^q_i; \mu(O^a)]) \quad (13)$$

where $\Sigma^q_i \in \mathbb{R}^b$ denotes the i-th row of the attention matrix, b is the number of hops of attention. In the same way, we can compute the attention matrix Σ^a for the question-aware answer representation.

After computing the multi-perspective co-attention matrices for the question and answer representations, we can obtain the interactive representations for question q and answer a, which are computed as:

$$emb^q = flatten(\Sigma^q \cdot O^q), \;\; emb^a = flatten(\Sigma^a \cdot O^a) \quad (14)$$

where *flatten* is the operation that flattens matrix into vector form.

3.5 Question Categorization

Given a question q, the goal of question categorization is to assign a category label to the question. Question categorization facilitates category-specific question representation learning, which enhances the quality of identifying discriminative features of the given question. It can be treated as a multi-class text classification task.

The question representation emb^q is fed into a task-specific fully-connected layer followed by a softmax layer to predict the category distribution $\hat{\mathbf{x}}$ of question q:

$$\hat{\mathbf{x}} = \text{softmax}(f_{FC}(emb^q)) \quad (15)$$

where f_{FC} is a multilayer perceptron (MLP).

We optimize this question categorization model by minimizing the cross-entropy between the predicted category distribution $\hat{\mathbf{x}}$ and the true category distribution $\mathbf{x}$ (one-hot vector):

$$L_1 = -\frac{1}{N} \sum_{i=1}^{N} \mathbf{x}_i \log(\hat{\mathbf{x}}_i), \quad (16)$$

where N is the number of training samples, $\mathbf{x}_i$ is the ground truth category label (one-hot vector) of the i-th question.

3.6 Community Question Answering

Category-aware Representation Learning

Inspired by [Cao *et al.*, 2017], we develop a category-aware transformation process to make the transformed question embeddings hold the category information. Formally, our model transforms the question representation emb^q into a category-specific question representation $\tilde{emb}^q$ by

$$\tilde{emb}^q = \tanh(\mathbf{W}_\mu \times emb^q) \quad (17)$$

where $\mathbf{W}_\mu \in \mathbb{R}^{d_c}$ is a transformation matrix, d_c is the dimensionality of the category-aware question representation. It is noteworthy that the knowledge-aware question representation and the category-specific question representation have the same dimensionality.

We develop the category-aware transformation matrix $\mathbf{W}_\mu$ conforming to the predicted question category so that the transformed question representation could capture category information. To that end, we introduce X sub-matrices $(\mathbf{W}^1_\mu, \cdots, \mathbf{W}^X_\mu)$, where each sub-matrix is respective to one question category. The category-aware transformation matrix $\mathbf{W}_\mu$ can be calculated as the weighted sum of the sub-matrices: $\mathbf{W}_\mu = \sum_{i=1}^{X} \hat{\mathbf{x}} \mathbf{W}^i_\mu$. Here, X is the number of question categories. In this manner, $\mathbf{W}_\mu$ can be biased to the sub-matrix of the predicted question category.

For CQA task, we concatenate the final question and answer representations and feed them into a task-specific fully-connected layer. A softmax function is then employed to predict the answer probability distribution:

$$\hat{\mathbf{y}} = \text{softmax}(f_{FC}([\tilde{emb}^q; emb^a])) \quad (18)$$

where f_{FC} is a multilayer perceptron (MLP).

Similar to Eq. (16), we also optimize the CQA model with supervised learning by minimizing the cross-entropy between the prediction label distribution $\hat{Y}$ and the true distribution Y:

$$L_2 = -\frac{1}{M} \sum_{i=1}^{M} \mathbf{y}_i \log(\hat{\mathbf{y}}_i), \quad (19)$$

where M is the number of question-answer pairs, $\mathbf{y}_i$ is the ground truth label (one-hot vector) of the i-th question-answer pair.

Ensemble Learning

We randomly choose 100 questions that are incorrectly predicted by KHAMA from SemEval-2017 test set to perform error analysis. We observe that the incorrectly predicted questions are imbalanced across question categories. The samples in some question categories such as “Family Life in Qatar” and “Moving to Qatar” are more difficult to predict than the samples in other categories such as “Doha Shopping” and “Cars and driving”. To develop a more stable and robust CQA model, we construct multiple CQA classifiers and ensemble their results as the final prediction result. Inspired by [Yang *et al.*, 2018], multiple CQA classifiers are combined together according to their weight vector $\boldsymbol{\alpha}$, learned from the adaptive boosting algorithm. The weight α_i of the i-th CQA classifier g_i is updated based on the training error ϵ_i of g_i on the training set, computed by:

$$\alpha_i = \frac{1}{2} ln \frac{1-\epsilon_i}{\epsilon_i}, \quad (20)$$

The final prediction model G is obtained by weighted voting:

$$Y = \text{softmax}(\sum_{i=1}^{T} \alpha_i g_i), \quad (21)$$

where α_i means the weight of each CQA classifier g_i for our final predictor Y. The *softmax* function is to predict the labels of CQA.

In the adaptive boosting algorithm, the individual classifiers are trained hierarchically to learn harder and harder samples of the classification problem. The i-th classifier is trained with more emphasis on different input samples, which is based on a probability distribution $D_i = \{d_2, \ldots, d_N\}$ to re-weighing the training samples. The process of the boosting algorithm is given in Algorithm 1.

Algorithm 1 The process of adaptive boosting.

```
1. **Input:** The training set $\{Q_{1:N}, A_{1:N}\}$; Initialize the sample weights $d_n = 1/N, n = 1, 2, \ldots, N$.
2. **For** $i = 1$ to $T$ **do**:
    (a) Train a CQA model $g_i$ with the training data.
    (b) Compute the training error $\epsilon_i$ of $g_i$.
    (c) Compute the classifier weight $\alpha_i$ based on Eq. 20.
    (d) Update the sample weight: $d_n = d_n * (1 - \alpha_i)$, if $Y_n = g_i(Q_n, A_n)$; otherwise, $d_n = d_n * (1 + \alpha_i)$.
    (e) Normalize the sample weight vector $D_i$.
    (f) Re-sample training data with weights $D_i$.
3. **end for**
4. Get the final prediction model $Y$ based on Eq. 21.
```

3.7 Multi-task Learning

Overall, the proposed KHAMA model has two subtasks, each has a training objective. To strengthen the shared question representation learning module, the two related tasks are optimized jointly. Formally, the objective function for the multi-task learning is minimized by:

$$L = \lambda_1 L_1 + \lambda_2 L_2, \tag{22}$$

where λ_1 and λ_2 control the importance of L_1 and L_2. We empirically show that setting $\lambda_1 = 0.2$ and $\lambda_2 = 0.8$ achieves the best performance of our model.

4 Experimental Setup

Experimental Datasets We evaluate our model on two benchmark datasets, i.e. SemEval-2015 Task 3 [Nakov *et al.*, 2015] and SemEval-2017 Task 3 [Nakov *et al.*, 2017], which consist of real-life data from the QatarLiving forum[1]. Table 1 shows the statistics of the datasets. Several candidate answers are given for each question, and each answer has a label “*Definitely Relevant*” (Good), “*Potentially Useful*” (Potential), or “*Bad*” (bad, dialog, non-English, other). Following the strategy as used in previous studies [Filice *et al.*, 2016], both the “*Potentially Useful” and “Bad” are considered as “Bad” in all experiments because the* “Potentially Useful” is the noisiest and smallest class. Some metadata is given for each question, e.g., the question category and the type of question.

Compared Methods We evaluate and compare KHAMA with several state-of-the-art baseline models, including *JAIST* [Tran *et al.*, 2015], *KeLP* [Filice *et al.*, 2016], *CNN* [Yu *et al.*, 2014], *LSTM* [Tan *et al.*, 2016b], *Bi-LSTM-attention* [Tan *et al.*, 2016b], *BGMN* [Wu *et al.*, 2017], *CNN-LSTM-CRF* [Xiang *et al.*, 2016], *AP-LSTM* [Dos Santos *et al.*, 2016], *AI-CNN* [Zhang *et al.*, 2017].

method	SemEval-2015			SemEval-2017		
	Train	Dev	Test	Train	Dev	Test
# of Ques.	2,376	266	300	5,124	327	293
# of Ans.	15,013	1,447	1,793	38,638	3,270	2,930
Avg. sub. len.	6.36	6.08	6.24	6.38	6.16	5.76
Avg. body len.	39.26	39.47	39.53	43.01	47.98	54.06
Avg. ans. len.	35.82	33.90	37.33	37.67	37.30	39.50

Table 1: Statistics of the two CQA datasets.

Evaluation Metrics We use the official evaluation measures [Dos Santos *et al.*, 2016; Zhang *et al.*, 2017] to verify KHAMA. Specifically, for SemEval-2015 dataset, two popular classification measures, accuracy and Macro F1 score, are used to evaluate the performance of CQA. For SemEval-2017 dataset, we use three measures to evaluate KHAMA, including mean average precision (MAP), accuracy and Macro F1 score. MAP is a widely used metric for evaluating ranking algorithms, which considers the ranks of the returned documents.

Implementation Details We use a subset of Freebase (FB5M3) as our KB, which includes 4,904,397 entities, 7,523 relations, and 22,441,880 facts. We initialize the graph embeddings through choosing the values from normal distribution $\mathcal{N}(0, 1)$ and set the size of the graph embedding as 100. We use the pre-trained word2vec [Mikolov *et al.*, 2013] with 100-dimensional embeddings to initialize the word embeddings, and initialize the word embeddings of out-of-vocabulary words as zero vector. The weight parameters are initialized using the Xavier uniform initializer, and the bias terms are initialized to zero. Both the hidden size of LSTM and the number of feature maps of CNN are set to 200. The size of each convolution filter is set to 2. We set the number of predictors in our ensemble learning to 5 (i.e., $T = 5$) and the number of entity candidates from KB for each entity mention to 6 (i.e., $K = 6$). We use Adadelta optimizer with a initial learning rate of 1×10^{-4}. Batch size is set as 64. L_2 regularization (weight decay = 0.001) and dropout strategy (dropout rate = 0.2) are used to avoid overfitting.

5 Experimental Results

5.1 Quantitative Evaluation

Tables 2-3 reports the results of KHAMA and compared models on SemEval-2015 and SemEval-2017 datasets, respectively. KHAMA achieves significantly better performance than the state-of-the-art competitors on the two datasets. For example, for the accuracy the proposed KHAMA method substantially and consistently outperforms other methods (improves 4.66% on SemEval-2015 and 4.78% on SemEval-2017). As we know, it is difficult to improve 1 percent of accuracy for CQA.

5.2 Ablation Study

To investigate the effectiveness of different factors of KHAMA, we also conduct ablation test in terms of removing

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

Method	Accuracy	F1 score
JAIST	79.10	78.96
KeLP	81.96	80.73
BGMN	81.24	80.22
CNN	77.33	76.92
LSTM	76.21	75.15
Bi-LSTM-attention	81.12	79.09
CNN-LSTM-CRF	82.15	81.33
AP-LSTM	79.45	79.06
AI-CNN	83.06	81.92
KHAMA (Ours)	**86.98**	**85.45**
w/o knowledge	84.36	82.56
w/o categorization	85.44	83.24
w/o boosting	84.25	82.33
w/o word-level	85.87	84.45
w/o phrase-level	86.23	84.92
w/o doc-level	85.74	84.76
w/o intra-doc	85.14	83.44

Table 2: Quantitative evaluation results on SemEval-2015.

Method	Accuracy	F1 score	MAP
JAIST	73.78	68.04	87.24
Kelp	73.89	69.87	88.43
BGMN	74.75	75.39	87.68
CNN	73.22	72.14	86.21
LSTM	74.05	73.45	86.28
Bi-LSTM-attention	76.60	74.82	88.05
BGMN	74.75	75.39	87.68
CNN-LSTM-CRF	77.18	77.04	87.66
AP-LSTM	77.64	76.82	87.82
AI-CNN	78.24	77.75	88.33
KHAMA (Ours)	**82.32**	**81.15**	**90.76**
w/o knowledge	80.12	78.54	88.32
w/o categorization	80.75	79.35	88.93
w/o boosting	79.44	78.69	87.64
w/o word-level	81.65	80.43	90.14
w/o phrase-level	81.73	80.36	89.32
w/o doc-level	81.07	79.84	88.21
w/o intra-doc	80.53	79.35	88.47

Table 3: Quantitative evaluation results on SemEval-2017.

factual knowledge from KB (w/o knowledge), question categorization task (w/o categorization), boosting algorithm (w/o boosting), word-level attention (w/o word-level), phrase-level attention (w/o phrase-level), document-level attention (w/o doc-level), intra-document attention (w/o intra-doc.), respectively. In particular, for the model without knowledge (i.e., w/o IA), we eliminate the word-level, phrase-level and document-level attention layers.

The ablation results are illustrated in Tables 2-3 (last seven rows). In general, combining all the factors achieves the best performance on all evaluation metrics. From Tables 2-3, we can see that the accuracy and F1 score decrease sharply when discarding the factual knowledge in KB and adaptive boosting algorithm. This is within our expectation since KB introduces background knowledge beyond the context to enrich the text representations and helps the model focusing on useful information. In addition, ensembling multiple classifiers with adaptive boosting could build a more effective and robust CQA model. Question categorization task also contributes to the effectiveness of KHAMA. This suggests that the auxiliary task improves the document representation learning and helps to identify discriminative features of the question.

5.3 The Effect of Parameters T and K

T is the number of classifiers in our ensemble learning. In this experiment, we analyze the impact of T on the overall performance of KHAMA by varying its value from 1 to 10 with step size 1 on the two datasets. We illustrate the experimental results in Figure 1 (left). As T increases from 1 to 3, the accuracy and F1 scores increase sharply till $T = 3$, after which the results become stable and slightly increase.

K represents the number of entity candidates from KB for each entity mention in questions and answers. In this experiment, we investigate the impact of K on the overall performance of KHAMA through varying its value from 1 to 10 with step size 1. The experimental results on SemEval-2015 are reported in Figure 1 (right). KHAMA obtains the best results when $K = 5$. As K increases from 1 to 10, the accuracy and F1 scores grow sharply till an optimal value (when $K = 5$), after which the accuracy and F1 scores decrease slightly.

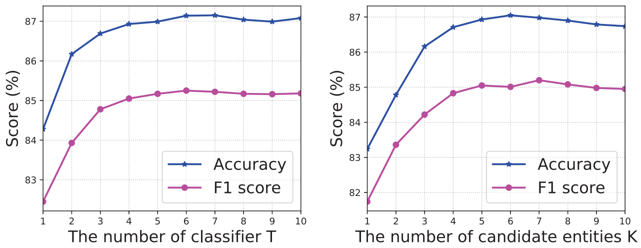

Figure 1: Experimental results of KHAMA on SemEval-2015 by varying the values of T (left) and K (right).

6 Conclusion

In this paper, we propose a knowledge-enhanced hierarchical attention for community question answering with multi-task learning and adaptive learning. We leverage external knowledge from the knowledge base (KB) to learn better representations of questions and answers by exploiting the semantic compositionality of the input sequences. In addition, we combine multiple CQA models with adaptive boosting to learn a more effective and robust CQA system. Extensive experiments on two benchmark datasets show that KHAMA obtains significantly better results than the compared methods.

Acknowledgments

This work was also partially supported by the SIAT Innovation Program for Excellent Young Researchers (No. Y8G027), Shenzhen Fundamental Research Project (No. JCYJ20180302145633177), NSFC (No. 61876208), Guangdong Provincial Scientific and Technological funds (2017B090901008, 2018B010108002). Min Yang was sponsored by CCF-Tencent Open Research Fund.

References

[Bollacker *et al.*, 2008] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In *SIGMOD*, pages 1247–1250. ACM, 2008.

[Cao *et al.*, 2017] Ziqiang Cao, Wenjie Li, Sujian Li, and Furu Wei. Improving multi-document summarization via text classification. In *AAAI*, pages 3053–3059, 2017.

[Chen *et al.*, 2017] Qin Chen, Qinmin Hu, Jimmy Xiangji Huang, Liang He, and Weijie An. Enhancing recurrent neural networks with positional attention for question answering. In *SIGIR*, pages 993–996. ACM, 2017.

[Dos Santos *et al.*, 2016] Cıcero Nogueira Dos Santos, Ming Tan, Bing Xiang, and Bowen Zhou. Attentive pooling networks. *CoRR, abs/1602.03609*, 2016.

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

[Guo *et al.*, 2017] Jiahui Guo, Bin Yue, Guandong Xu, Zhenglu Yang, and Jin-Mao Wei. An enhanced convolutional neural network model for answer selection. In *WWW*, pages 789–790, 2017.

[Heilman and Smith, 2010] Michael Heilman and Noah A Smith. Tree edit models for recognizing textual entailments, paraphrases, and answers to questions. In *NAACL*, pages 1011–1019, 2010.

[Lei *et al.*, 2018] Zeyang Lei, Yujiu Yang, and Min Yang. Saan: A sentiment-aware attention network for sentiment analysis. In *SIGIR*, 2018.

[Mikolov *et al.*, 2013] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. In *ICLR*, pages 1–12, 2013.

[Nakov *et al.*, 2015] Preslav Nakov, Lluís Màrquez, Walid Magdy, Alessandro Moschitti, Jim Glass, and Bilal Randeree. SemEval-2015 task 3: Answer selection in community question answering. In *SemEval 2015*, pages 269–281, 2015.

[Nakov *et al.*, 2017] Preslav Nakov, Doris Hoogeveen, Lluís Màrquez, Alessandro Moschitti, Hamdy Mubarak, Timothy Baldwin, and Karin Verspoor. SemEval-2017 task 3: Community question answering. In *SemEval-2017*, pages 27–48, 2017.

[Perozzi *et al.*, 2014] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: online learning of social representations. In *SIGKDD*, pages 701–710, 2014.

[Peters *et al.*, 2018] Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. *NAACL*, 2018.

[Qiu and Huang, 2015] Xipeng Qiu and Xuanjing Huang. Convolutional neural tensor network architecture for community-based question answering. In *IJCAI*, 2015.

[Shen *et al.*, 2018] Ying Shen, Yang Deng, Min Yang, Yaliang Li, Nan Du, Wei Fan, and Kai Lei. Knowledge-aware attentive neural network for ranking question answer pairs. In *SIGIR*, pages 901–904, 2018.

[Surdeanu *et al.*, 2008] Mihai Surdeanu, Massimiliano Ciaramita, and Hugo Zaragoza. Learning to rank answers on large online qa collections. *ACL*, pages 719–727, 2008.

[Tan *et al.*, 2016a] Ming Tan, Cicero Dos Santos, Bing Xiang, and Bowen Zhou. Improved representation learning for question answer matching. In *ACL*, pages 464–473, 2016.

[Tan *et al.*, 2016b] Ming Tan, Cicero dos Santos, Bing Xiang, and Bowen Zhou. Lstm-based deep learning models for non-factoid answer selection. In *ICLR*, 2016.

[Tay *et al.*, 2017] Yi Tay, Minh C Phan, Luu Anh Tuan, and Siu Cheung Hui. Learning to rank question answer pairs with holographic dual lstm architecture. In *SIGIR*, 2017.

[Tay *et al.*, 2018] Yi Tay, Luu Anh Tuan, and Siu Cheung Hui. Cross temporal recurrent networks for ranking question answer pairs. In *AAAI*, 2018.

[Tran *et al.*, 2015] Quan Hung Tran, Vu Tran, Tu Vu, Minh Nguyen, and Son Bao Pham. Jaist: Combining multiple features for answer selection in community question answering. In *SemEval 2015*, pages 215–219, 2015.

[Wang and Nyberg, 2015] Di Wang and Eric Nyberg. A long short-term memory model for answer sentence selection in question answering. In *ACL*, 2015.

[Wu *et al.*, 2017] Guoshun Wu, Yixuan Sheng, Man Lan, and Yuanbin Wu. Ecnu at semeval-2017 task 3: Using traditional and deep learning methods to address community question answering task. In *SemEval-2017*, 2017.

[Xiang *et al.*, 2016] 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 *COLING*, pages 1231–1241, 2016.

[Yang *et al.*, 2018] Dongdong Yang, Senzhang Wang, and Zhoujun Li. Ensemble neural relation extraction with adaptive boosting. In *IJCAI*, 2018.

[Yang *et al.*, 2019] Min Yang, Wenting Tu, Qiang Qu, Wei Zhou, Qiao Liu, and Jia Zhu. Advanced community question answering by leveraging external knowledge and multi-task learning. *Knowledge-Based Systems*, 2019.

[Yin *et al.*, 2015] Wenpeng Yin, Hinrich Schütze, Bing Xiang, and Bowen Zhou. Abcnn: Attention-based convolutional neural network for modeling sentence pairs. *Computer Science*, 2015.

[Yu *et al.*, 2014] Lei Yu, Karl Moritz Hermann, Phil Blunsom, and Stephen Pulman. Deep learning for answer sentence selection. In *Proceedings of Deep Learning and Representation Learning Workshop*. NIPS, 2014.

[Zhang *et al.*, 2017] Xiaodong Zhang, Sujian Li, Lei Sha, and Houfeng Wang. Attentive interactive neural networks for answer selection in community question answering. In *AAAI*, pages 3525–3531, 2017.