Knowledge-aware Attentive Neural Network for Ranking Question Answer Pairs

Ying Shen[1], Yang Deng[1], Min Yang[2], Yaliang Li[3], Nan Du[3], Wei Fan[3], Kai Lei[1,*]
[1]School of Electronics and Computer Engineering, Peking University Shenzhen Graduate School
[2]Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences [3]Tencent Medical AI Lab
shenying@pkusz.edu.cn,ydeng@pku.edu.cn,min.yang@siat.ac.cn
{yaliangli,ndu,davidwfan}@tencent.com,leik@pkusz.edu.cn

ABSTRACT

Ranking question answer pairs has attracted increasing attention recently due to its broad applications such as information retrieval and question answering (QA). Significant progresses have been made by deep neural networks. However, background information and hidden relations beyond the context, which play crucial roles in human text comprehension, have received little attention in recent deep neural networks that achieve the state of the art in ranking QA pairs. In the paper, we propose KABLSTM, a Knowledge-aware Attentive Bidirectional Long Short-Term Memory, which leverages external knowledge from knowledge graphs (KG) to enrich the representational learning of QA sentences. Specifically, we develop a context-knowledge interactive learning architecture, in which a context-guided attentive convolutional neural network (CNN) is designed to integrate knowledge embeddings into sentence representations. Besides, a knowledge-aware attention mechanism is presented to attend interrelations between each segments of QA pairs. KABLSTM is evaluated on two widely-used benchmark QA datasets: WikiQA and TREC QA. Experiment results demonstrate that KABLSTM has robust superiority over competitors and sets state-of-the-art.

CCS CONCEPTS

• **Information systems → Question answering**;

ACM Reference Format:
Ying Shen[1], Yang Deng[1], Min Yang[2], Yaliang Li[3], Nan Du[3], Wei Fan[3], Kai Lei[1,*]. 2018. Knowledge-aware Attentive Neural Network for Ranking Question Answer Pairs. In *SIGIR '18: The 41st International ACM SIGIR Conference on Research and Development in Information Retrieval, July 8–12, 2018, Ann Arbor, MI, USA.* 4 pages. https://doi.org/10.1145/3209978.3210081

1 INTRODUCTION

Ranking question answer pairs, also known as answer selection, has become increasingly important in a variety of QA such as community-based question answering (CQA) and factoid question answering. Given a question, answer selection aims to pick out the most relevant answer from a set of candidates. Inspired by the recent successes of deep learning in natural language processing, the majority of literature employed deep neural networks, e.g., convolutional neural network (CNN) [6] or recurrent neural network (RNN) [11], to automatically select answers. 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 two texts. 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 [2, 4, 7].

Table 1: Example of QA candidate pairs.

Question	When was ***Pokemon*** first started ?
Positive Answer	Is a media franchise published and owned by Japanese video game company ***Nintendo*** and created by ***Satoshi Tajiri*** in ***1996*** .
Negative Answer	The official logo of ***Pokemon*** for its international release ; "***Pokemon***" is short for the original Japanese title of "***Pocket Monsters***" .

Despite the effectiveness of previous studies, ranking question answer pairs in real-world remains a challenge. (i) First, the background knowledge from open-domain knowledge graphs (KGs) plays a crucial role in question answering. Considering the example in Table 1, existing context-based models 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 word level. However, with the background knowledge, we can correctly identify the positive answer based on the relative facts contained in the KG such as (*Pokemon*, *owned_by*, *Nintendo*), (*Pokemon*, *created_by*, *Satoshi Tajiri*) and even (*Pokemon*, *created_in*, *1996*). Despite its usefulness, to our best knowledge, the background knowledge from KGs receives little attention in recent neural network models to rank question answer pairs [5, 9, 10]. (ii) In addition, the issues of redundancy and noise prevalent in real-world applications (e.g., CQA) are remained to be settled. However, previous researches [3, 13] leverage external knowledge from KG to exclusively conduct the knowledge-aware learning of individual sentence rather than capture the interrelations between different sentences, which is important for ranking question answer pairs.

To alleviate these limitations, we propose a knowledge-aware attentive neural network to interactively learn knowledge-based sentence representations and context-based sentence representations for ranking QA pairs. In specific, we first employ knowledge embedding methods to pre-train the knowledge embeddings from

SIGIR '18, July 8–12, 2018, Ann Arbor, MI, USA

ACM ISBN 978-1-4503-5657-2/18/07…$15.00
https://doi.org/10.1145/3209978.3210081

* Corresponding Author

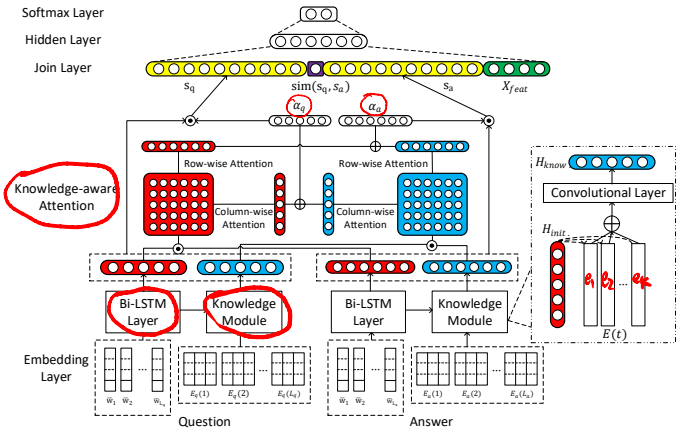

Figure 1: Knowledge-aware Attentive Bi-LSTM. Blue, red and yellow matrices denote knowledge-based representations, initial contextual representations and final knowledge-aware sentence representations, respectively.

KG. Then, we design a context-guided attentive CNN to learn the knowledge-based sentence representation from discrete candidate entity embeddings in KG. Finally, we present a knowledge-aware attention mechanism to learn knowledge-aware sentence representations of questions and answers, which adaptively decides the important information of questions and answers based on both the context and the background knowledge.

The main contributions of this paper can be summarized as follows: (1) We propose a novel deep learning model, Knowledge-aware Attentive Bi-LSTM (KABLSTM), which leverages external knowledge from KG to capture background information of the questions and answers for ranking QA pairs; (2) We develop a context-knowledge interactive learning architecture, which exploits the interactive information from input texts and KG to supervise the representational learning of both sentences and external knowledge; (3) The experimental results show that KABLSTM consistently outperforms the state-of-the-art methods.

2 METHOD

Given a question q, our model aims to rank a set of candidate answers $A = \{a_1, \ldots, a_n\}$. Concretely, we first employ a pair of Bi-LSTMs to learn the initial representations of questions and answers, separately (Section 2.1). Then a context-guided attentive CNN is designed to learn the knowledge-based sentence representation from entities in the sentence (Section 2.2). Afterwards, we introduce our proposed knowledge-aware attention mechanism to learn the final knowledge-aware attentive sentence representation (Section 2.3). Finally, there is a fully connected hidden layer before the final binary classification to join all the features (Section 2.4). Figure 1 illustrates the overall architecture of KABLSTM.

2.1 Bidirectional Long Short-Term Memory

In the Bi-LSTM, the model not only captures information from past contexts but also have access to future contexts. The Bi-LSTM layer contains two sub-networks for the head-to-tail and the tail-to-head context respectively. The output at time step t is represented by $h_t = [\overrightarrow{h_t} : \overleftarrow{h_t}]$, in which $\overrightarrow{h_t}$ is the output of the forward network and $\overleftarrow{h_t}$ is that of the backward network. Given the question q and the answer a, we generate the initial contextual sentence representation $H_{init} \in \mathbb{R}^{L \times d_h}$ for both the question and the answer, where L and d_h are the length of sentences and the dimension of h_t.

$$Q_{init} = \text{Bi-LSTM}(q); \quad A_{init} = \text{Bi-LSTM}(a), \tag{1}$$

2.2 Knowledge Module: Knowledge-based Sentence Representation Learning

Knowledge module is designed to learn knowledge-based sentence representations from discrete candidate entity embeddings with the guidance of contextual information. 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 sentence, due to the ambiguity of the entity, e.g., "Boston" can refer to a city or a person. We design a context-guided attention mechanism to learn the knowledge representation of each entity mention in the sentence by congregating the embeddings of the corresponding candidate entities in the KG. The contextual sentence representations of questions and answers are learned by Bi-LSTM layers, while the embeddings of entities in KG are pretrained by TransE [1]. Formally, we present candidate entities for the entity mention at time step t as $E(t) = \{e_1, e_2, \ldots, e_K\} \in \mathbb{R}^{K \times d_e}$, where d_e is the dimension of the entity embedding in KG. Then, the final context-guided embedding for the word at time step t (in accord with t in Sect. 2.1) is given by

$$m(t) = W_{em} E(t) + W_{hm} H_{init}, \tag{2}$$

$$s(t) \propto exp\left(w_{ms}^T tanh\left(m(t)\right)\right) \tag{3}$$

$$\widetilde{E(t)} = E(t)\, s(t)^T, \tag{4}$$

where W_{em}, W_{hm} and w_{ms} are attention parameters to be learned. $m(t)$ is a context-guided knowledge vectors, and $s(t)$ denotes the context-guided attention weight that is applied over each candidate entity embedding e_i.

This procedure produces a context-guided representation for each entity mention in the sentence. An CNN layer is then employed to capture the local n-gram information and learn a higher level knowledge-based sentence representation $H_{know} \in \mathbb{R}^{L \times d_f}$ from attentive knowledge embeddings $\widetilde{E} \in \mathbb{R}^{L \times d_e}$, where d_f is the total filter sizes of CNN and L is the length of the sentence.

$$Q_{know} = CNN(\widetilde{E}_q); \quad A_{know} = CNN(\widetilde{E}_a). \tag{5}$$

2.3 Knowledge-aware Attention: Context-based Sentence Representation Learning

Knowledge-aware attention mechanism is an approach that enables QA pairs be aware of some background information and hidden relations beyond the text. For both question and answer sentences, there are two different sentence-level representation vectors. Q_{init} and A_{init} are learned from word embeddings, while Q_{know} and A_{know} are derived from knowledge module. These two kinds of sentence vectors are input into knowledge-aware attention layer.

As is illustrated in Figure 1, we first compute two attention matrices M_{init} and M_{know}:

$$M_{init} = tanh\left(Q_{init}^T U_{init} A_{init}\right) \tag{6}$$

$$M_{know} = tanh\left(Q_{know}^T U_{know} A_{know}\right), \quad (7)$$

where $U_{init} \in \mathbb{R}^{d_h \times d_h}$ and $U_{know} \in \mathbb{R}^{d_f \times d_f}$ are parameter matrices to be learned.

Then column-wise and row-wise max-pooling are applied on M_{init} to generate context-based attention vectors for question and answer separately, while we conduct the same operation over M_{know} for knowledge-based attention vectors. In order to incorporate the knowledge-aware influence of the question words into answers' attentive representations and vice versa, we merge these two attention vectors to obtain the final knowledge-aware attention vectors, α_q and α_a:

$$\alpha_q \propto \left(\text{softmax}\left(\max_{1<l<L_q} M_{init}\right) + \text{softmax}\left(\max_{1<l<L_q} M_{know}\right)\right), \quad (8)$$

$$\alpha_a \propto \left(\text{softmax}\left(\max_{1<l<L_a} M_{init}^T\right) + \text{softmax}\left(\max_{1<l<L_a} M_{know}^T\right)\right). \quad (9)$$

We conduct dot product between the attention vectors and the overall sentence vectors to form the final knowledge-aware attentive sentence representations of question q (i.e., s_q) and answer a (i.e., s_a):

$$s_q = [Q_{init} : Q_{know}]^T \alpha_q, \quad s_a = [A_{init} : A_{know}]^T \alpha_a, \quad (10)$$

where [:] is the concatenation operation.

2.4 Hidden Layer and Softmax Layer

Following the strategies in [6, 8], additional features are exploited in our overall architecture. First, we compute the bilinear similarity score between final attentive QA vectors:

$$sim\left(s_q, s_a\right) = s_q^T W s_a, \quad (11)$$

where $W \in \mathbb{R}^{L \times L}$ is a similarity matrix to be learned. Besides, the same word overlap features $X_{feat} \in \mathbb{R}^4$ are incorporated into our model, which can be referred to [6, 8]. Thus, the inputs of the hidden layer is a vector $\left[s_q, sim(s_q, s_a), s_a, X_{feat}\right]$, and its output then go through a softmax layer for binary classification. The overall model is trained to minimize the cross-entropy loss function:

$$L = -\sum_{i=1}^{N} [y_i \log p_i + (1 - y_i) \log (1 - p_i)] + \lambda \|\theta\|_2^2, \quad (12)$$

where p is the output of the softmax layer. θ contains all the parameters of the network and $\lambda\|\theta\|_2^2$ is the L2 regularization.

3 EXPERIMENT

3.1 Experimental Setup

Datasets and Metrics. We evaluate our method on two widely-used QA benchmark datasets: TREC QA and WikiQA. Original TREC QA dataset, collected from TREC QA track 8-13 data [12], is a benchmark for factoid question answering. Besides, there is a cleaned version of TREC QA dataset, which removes questions that have only positive or negative answers or no answer. WikiQA dataset is an open-domain factoid answer selection benchmark, in which the standard pre-processing steps as [14] is employed to extract questions with correct answers. The statistics of these two datasets are described in Table 2. Following previous work [12, 14], the mean average precision (MAP) and mean reciprocal rank (MRR) are adopted as our evaluation metrics.

Table 2: Summary statistics of datasets.

Dataset (train/dev/test)	#Question	#QA Pairs	%Correct
TREC QA(original)	1229/82/100	53417/1148/1517	12.0/19.3/18.7
TREC QA(cleaned)	1160/65/68	53313/1117/1442	11.8/18.4/17.2
WikiQA	873/126/243	20360/1130/2352	12.0/12.4/12.5

Table 3: Result on TREC QA(original) (with ablation study)

Model	MAP	MRR
Wang & Nyberg (2015) [11]	0.7134	0.7913
Severyn & Moschitti (2015) [6]	0.7459	0.8078
Tay et al. (2017) [8]	0.7499	0.8153
Rao et al. (2016) [5]	0.7800	0.8340
Tay et al. (2018) [9]	0.7712	0.8384
KABLSTM	**0.7921**	**0.8444**
w/o attention	0.7805	0.8309
w/o KG	0.7596	0.8099

Table 4: Result on TREC QA(cleaned) and WikiQA (with ablation study)

Models	TREC QA(cleaned)		WikiQA	
	MAP	MRR	MAP	MRR
Yang et al. (2015) [14]	0.6951	0.7633	0.6520	0.6652
Santos et al. (2016) [4]	0.7530	0.8511	0.6886	0.6957
Rao et al. (2016) [5]	0.8010	0.8770	0.7010	0.7180
Chen et al. (2017) [2]	0.7814	0.8513	0.7212	0.7312
Wang et al. (2016) [10]	0.7369	0.8208	**0.7341**	0.7418
KABLSTM	**0.8038**	**0.8846**	0.7323	**0.7494**
w/o attention	0.7821	0.8654	0.7214	0.7363
w/o KG	0.7633	0.8320	0.7086	0.7255

Implementation Details. Pre-trained GloVE embeddings[1] of 300 dimensions are adopted as word embeddings. We use a subset of Freebase (FB5M[2]) as our KG, which includes 4,904,397 entities, 7,523 relations, and 22,441,880 facts.

For all the implemented models, we apply the same parameter settings. The LSTM hidden layer size and the final hidden layer size are both set to 200. The learning rate and the dropout rate are set to 0.0005 and 0.5 respectively. We train our models in batches with size of 64. All other parameters are randomly initialized from [-0.1, 0.1]. The model parameters are regularized with a L2 regularization strength of 0.0001. The maximum length of sentence is set to be 40. In the knowledge module, the width of the convolution filters is set to be 2 and 3, and the number of convolutional feature maps and the attention sizes are set to be 200.

3.2 Experimental Results

The experimental results on TREC QA and WikiQA are summarized in Table 3 and Table 4. We compare our results with the recent work reported in the literature. For original TREC QA dataset, five state-of-the-art baselines are adopted for comparison: (1) a combination of the Bi-LSTM model and BM25 model [11]; (2) a CNN-based architecture with overlap features [6]; (3) dual Bi-LSTM models

[1] http://nlp.stanford.edu/data/glove.6B.zip
[2] https://research.facebook.com/researchers/1543934539189348

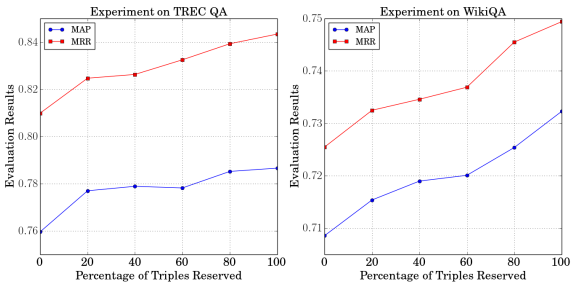

Figure 2: Effect of KG completeness

with holographic composition approach [8]; (4) the LSTM model with noise-contrastive estimation approach [5]; (5) a LSTM-based model using temporal gates to synchronously and jointly learn the interactions between text pairs [9]. For cleaned TREC QA and WikiQA, in addition to [5], we compare our method with four more strong baselines: (1) a bigram CNN model with average pooling [14]; (2) the attentive pooling LSTM network [4]; (3) a position-aware attention based RNN [2]; (4) inner attention based RNN [10]. We observe that KABLSTM substantially and consistently outperforms the existing methods by a noticeable margin on both TREC QA and WikiQA datasets. For instance, on the original TREC QA dataset, KABLSTM improves 2% on MAP over these baselines.

In order to analyze the effectiveness of different factors of KABLSTM, we also report the ablation test in terms of discarding knowledge-aware attention mechanism (w/o attention) and knowledge graph information (w/o KG), respectively. Generally, both factors contribute, and it makes larger performance boosting to integrate knowledge graph information. Even the basic LSTM model with external knowledge integrated by our knowledge module (w/o attention) achieves competitive results with these strong baselines, which demonstrates the effectiveness of incorporating background knowledge into ranking QA pairs. This is within our expectation since KG introduces background knowledge beyond the context to enrich overall sentence representations, while the knowledge-aware attention mechanism further enhances mutual representational learning of QA sentences.

3.3 Analysis

3.3.1 Completeness of Knowledge Graph. To analyze the performance of our model with respect to the the completeness of the knowledge graph, we report the MAP and MRR results with the incomplete subgraphs that have only 20%-80% triples reserved. Figure 2 shows that our model is robust and achieves excellent performance on the KG with different completeness. As one may expect, training with more complete KG actually improves the overall performance, which also indicates the importance of background knowledge in QA.

3.3.2 Case Study. KABLSTM provides an intuitive way to inspect the soft-alignment between the question and the answers by visualizing the attention weight from Equation (9). Due to the limited space, we randomly choose one question-answer pair from TREC QA dataset and visualize the attention scores predicted by AP-BLSTM and KABLSTM in Figure 3, respectively. The color depth indicates the importance degree of the words, the darker the more important. We observe that AP-BLSTM pays much attention to

Question	*who is the president or chief executive of amtrak ?*
AP-BLSTM	" long-term success here has to do with doing it right , getting it right and increasing market share , " said george warrington , amtrak 's president and chief executive .
KABLSTM	" long-term success here has to do with doing it right , getting it right and increasing market share , " said george warrington , amtrak 's president and chief executive .

Figure 3: An example of the visualization of attention

those words that are contextually related to the question, such as such as "amtrak", "president", "chief executive", while neglecting the knowledge beyond the context of the question like "george warrington". This limitation can be alleviated decently by knowledge-aware attention mechanism, since there is a strong correlation between "amtrak" and "george warrington" in the external knowledge graph.

4 CONCLUSIONS

In this paper, we propose a knowledge-aware attentive neural network for ranking QA pairs, which effectively incorporate external knowledge from KGs into sentence representational learning. The knowledge-aware attention mechanism is proved to be more effective to notice crucial information between questions and answers than current attention mechanism. Experimental results on two benchmark datasets demonstrate the superiority of our proposed method on answer selection task.

ACKNOWLEDGMENTS

This work was financially supported by the National Natural Science Foundation of China (No.61602013), Shenzhen Science and Technology Innovation Committee (Grant No. JCYJ20151030154330711) and the Shenzhen Key Fundamental Research Projects (Grant No. JCYJ20170818091546869).

REFERENCES

[1] Antoine Bordes, Nicolas Usunier, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *NIPS*. 2787–2795.
[2] Qin Chen, Qinmin Hu, Jimmy Xiangji Huang, Liang He, and Weijie An. 2017. Enhancing Recurrent Neural Networks with Positional Attention for Question Answering. In *SIGIR*. ACM, 993–996.
[3] Yun-Nung Chen, Dilek Hakkani-Tur, Gokhan Tur, Asli Celikyilmaz, Jianfeng Gao, and Li Deng. 2016. Knowledge as a teacher: Knowledge-guided structural attention networks. *arXiv preprint arXiv:1609.03286* (2016).
[4] Cıcero Nogueira dos Santos, Ming Tan, Bing Xiang, and Bowen Zhou. 2016. Attentive pooling networks. *CoRR, abs/1602.03609* (2016).
[5] Jinfeng Rao, Hua He, and Jimmy Lin. 2016. Noise-contrastive estimation for answer selection with deep neural networks. In *CIKM*. ACM, 1913–1916.
[6] Aliaksei Severyn and Alessandro Moschitti. 2015. Learning to Rank Short Text Pairs with Convolutional Deep Neural Networks. In *SIGIR*. 373–382.
[7] Ming Tan, Cicero Dos Santos, Bing Xiang, and Bowen Zhou. 2016. Improved Representation Learning for Question Answer Matching. In *ACL*. 464–473.
[8] Yi Tay, Minh C Phan, Luu Anh Tuan, and Siu Cheung Hui. 2017. Learning to Rank Question Answer Pairs with Holographic Dual LSTM Architecture. In *SIGIR*.
[9] Yi Tay, Luu Anh Tuan, and Siu Cheung Hui. 2018. Cross Temporal Recurrent Networks for Ranking Question Answer Pairs. In *AAAI*.
[10] Bingning Wang, Kang Liu, and Jun Zhao. 2016. Inner Attention based Recurrent Neural Networks for Answer Selection. In *ACL*. 1288–1297.
[11] Di Wang and Eric Nyberg. 2015. A Long Short-Term Memory Model for Answer Sentence Selection in Question Answering. In *ACL*. 707–712.
[12] Mengqiu Wang, Noah A. Smith, and Teruko Mitamura. 2007. What is the Jeopardy Model? A Quasi-Synchronous Grammar for QA. In *EMNLP*. 22–32.
[13] Bishan Yang and Tom Mitchell. 2017. Leveraging knowledge bases in lstms for improving machine reading. In *ACL*, Vol. 1. 1436–1446.
[14] Yi Yang, Wen Tau Yih, and Christopher Meek. 2015. WikiQA: A Challenge Dataset for Open-Domain Question Answering. In *EMNLP*. 2013–2018.