# Text-Enhanced Question Answering over Knowledge Graph

Jiaying Tian
tjy123jsls@163.com
Nanjing University of Aeronautics and Astronautics
Nanjing, Jiangsu, China

Bohan Li*
bhli@nuaa.edu.cn
Nanjing University of Aeronautics and Astronautics
Nanjing, Jiangsu, China

Ye Ji
goodboyjy@hotmail.com
Nanjing University of Aeronautics and Astronautics
Nanjing, Jiangsu, China

Jiajun Wu
w15250465300@163.com
Nanjing University of Aeronautics and Astronautics
Nanjing, Jiangsu, China

## ABSTRACT

Question answering over knowledge graph is an important area of research within question answering. Existing methods mainly focus on the utilization of information in knowledge graphs and ignore the abundant external information of entities. However, knowledge graphs are usually incomplete and entities in knowledge graphs are not completely described. In this paper, we propose a novel text-enhanced question answering model over knowledge graph by taking advantage of the rich context information in a text corpus. We believe the rich textual context information can effectively alleviate the information loss in knowledge graphs and enhance the knowledge representation capability in the answer end. To this end, we apply an attention model to realize dynamic fusion of internal and external information. Besides, Transformer Encoder network is used to obtain the representation of input question and descriptive text. The experiments on the WebQuestions dataset prove that compared with other state-of-the-art QA methods, our method can effectively improve the accuracy.

## CCS CONCEPTS

• **Information systems → Question answering**.

## KEYWORDS

Question answering; Knowledge graph; Embedding model

**ACM Reference Format:**
Jiaying Tian, Bohan Li, Ye Ji, and Jiajun Wu. 2021. Text-Enhanced Question Answering over Knowledge Graph. In *The 10th International Joint Conference on Knowledge Graphs (IJCKG'21), December 6–8, 2021, Virtual Event, Thailand*. ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3502223.3502241

*Corresponding author


*IJCKG'21, December 6–8, 2021, Virtual Event, Thailand*

ACM ISBN 978-1-4503-9565-6/21/12…$15.00
https://doi.org/10.1145/3502223.3502241

## 1 INTRODUCTION

Question answering over knowledge graph(KGQA) is an important application of knowledge graph in downstream tasks. Recent years, researchers have proposed lots of methods to solve the KGQA task. These methods can be roughly categorized into two main groups: semantic parsing based approaches(SP-based approaches) and information retrieval based approaches(IR-based approaches). The goal of SP-based approaches is to construct a semantic parser which converts natural language questions into intermediate logic forms. Traditional semantic parsers [16] need annotated logical forms as supervision for model training. These approaches usually struggle with the problem of coverage since the logical predicate are not comprehensive, and they are limited to specific domains. Recent efforts to overcome these limitations include Abujabal et al. [1] and Hu et al. [10] trying to construct hand-crafted rules or features, and Liang et al. [12] and Krishnamurthy et al.[11] trying to apply weak supervision by using question-answer pairs or distant supervision instead of full semantic annotations.

IR-based approaches firstly construct a set of candidate answers from the knowledge graph, and then map input question and candidate answers into vector space. Finally the similarity scores between question and candidate answers are calculated and used to get the final answer. The most important part of IR-based approaches is the method to obtain low-dimensional vector of question and answers. Most approaches utilize deep neural networks to learn the representations of the question and candidate answers while the construction of neural networks varies. Inspired by the process of deep learning in natural language processing(NLP) tasks, Bordes et al. [4] adopt a simplified way of bag-of-words model to obtain the question representation and use the subgraph of candidate answer to help represent candidate answer. Dong et al. [8] introduce the multi-column convolutional neural networks(MCCNNs) to enhance the information acquisition ability in the question end. Hao et al. [9] apply Bi-LSTM networks to represent question and the information of candidate answer in knowledge graph is used as an aid. Other approaches try to utilize information outside the knowledge graph, for example, Xu et al. [18] [19] use the Wikipedia free text as the external knowledge in the question end. However, knowledge graphs are usually incomplete and context information in knowledge graph is often cluttered. We believe that the context information of candidate answer can be enhanced by the rational utilization of external text information.

In the field of knowledge graph representation, entity description text effectively improves the performance of knowledge graph embedding models. Entity description text contains rich entity information. These information can be used as auxiliary information of structured information with high confidence interval in the knowledge graph to help the model represent the knowledge more accurate. Hence, some researchers considered that knowledge representation learning needs to incorporate more semantic information. Xie et al. [17] proposed a new representation learning model based on entity hierarchical type (TKRL). TKRL believes that an entity has multiple hierarchical types. In different semantic environments, entities have multiple representations according to their hierarchical types. Based on the TKRL model, RAHMAN et al.[13] proposed a knowledge graph embedding model TPRC using entity type attributes in a relational context. Wang et al. [15] proposed a text enhanced representation learning method (TEKE) for knowledge graph which enhances the effect of knowledge embedding. TEKE mainly refers to the text description information of entities to improve the effect of knowledge embedding. We assume that entity description text is helpful as a supplement to information of entities in knowledge graph. Build on top of that, we propose a novel model to improve the representation of candidate answers by associating entities with external text in the answer end. Besides, specially designed attention model is applied to dynamically fuse internal and external information.

In summary, our contributions are as follows: 1) we use co-occurrence network to associate entities in knowledge graph with external description text; 2) we design a novel approach to describe candidate answer in knowledge graph and apply an attention model to fuse internal and external information; 3) we conduct experiments on the WebQuestions dataset [2] and the results prove the effectiveness of our method.

## 2 RELATED WORK

In general, we follow the way of IR-based approaches. To be more specific, input question and candidate answers are embedded into vector space, then similarity scores are calculated to get ranking. In question end, Bordes et al. [4] adopt a simplified way of bag-of-words model, Dong et al. [8] introduce the multi-column convolutional neural networks(MCCNNs), and Hao et al. [9] apply Bi-LSTM networks. We construct Transformer Encoder network [14] to obtain question representation. In answer end, Bordes et al. [4] firstly make use of the subgraph of the candidate answer. Dong et al. [8] put more attention on the answer path, answer context and answer type. Different from the methods above, Hao et al. [9] use different aspects of the candidate answer to help represent the input question and use the input question to adjust the weight of different answer aspects. We draw on the work in the field of knowledge graph representation and use the text description information of entities to enhance the information of candidate answers.

## 3 METHODOLOGY

The whole architecture of our approach is shown as figure 1. Given an input question, we first retrieve its topic entity through Freebase API [3] and collect all the 2-hop nodes of the topic entity in FB2M as candidate answers. Then the input question is fed into a stacked Transformer neural network to get its vector representation. For each candidate answer, we use co-occurrence network to get its unique entity description text. The entity description text constructs external information through word embedding matrix. The word embedding matrix is pre-trained with BERT [7]. Internal information contains entity itself, entity type, entity relation and entity context. The we apply an attention model to dynamically fuse internal and external information. Finally, we calculate the similarity score for each candidate answer.

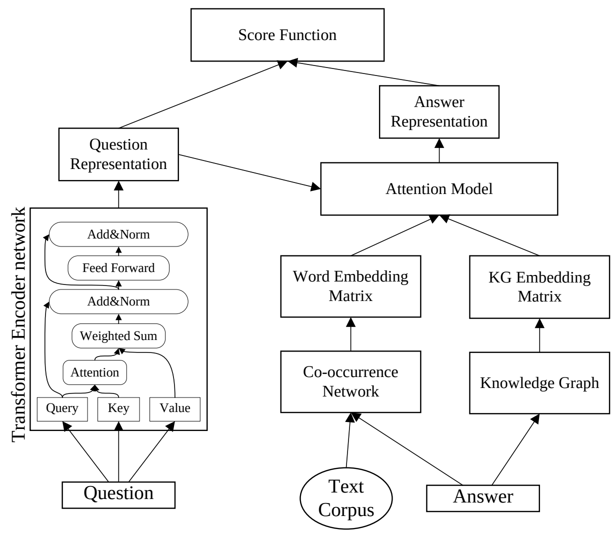


**Figure 1: The whole architecture of our method.**

### 3.1 Question Representation

The input question Q is expressed as a sequence of words $q = (w_1, w_2, \ldots, w_n)$, where $w_i$ denotes the $i$th word. We first construct a word embedding matrix $E_w \in R^{d\times|w|}$ to get the word embeddings. The word embedding matrix $E_w$ is randomly initialized and the parameters are updated in the training process. Here, $d$ means the dimension of word embeddings, and $|w|$ is the vocabulary size. Then these word embeddings are fed into a six-layer Transformer network to obtain the question representation q. The dimension of q is also set as $d$.

### 3.2 Candidate answer representation

**Internal information representation** We employ TransE [6] as the knowledge graph embedding model. For each candidate answer, We use three answer aspects to describe the information of candidate answer in the knowledge graph. Answer entity $a_e$ denotes the embedding vector of candidate answer, answer relation $a_r$ denotes the average of embeddings of relations that appear on the answer path, answer context $a_c$ denotes the average of embeddings of entities and predicates that directly connect to the answer entity.
**External information representation** Given a knowledge graph $\boldsymbol{G}$ and textual corpus $\boldsymbol{T} = \{\boldsymbol{w}_1, \boldsymbol{w}_2, \cdots, \boldsymbol{w}_n\}$, we first conduct entity annotation with entity linking tools to label the entities in knowledge graph and get an entity-annotated text corpus $\boldsymbol{D}$ =

$\{\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_m\}$. Here $m \le n$ because multiple adjacent words could be labeled as one entity. Then we construct a co-occurrence network to bridge the candidate answer entity $a$ and the entity-annotated text corpus $\boldsymbol{D}$.

For each candidate $a$, we use co-occurrence frequency to find its external neighboring nodes and these nodes constitute the external textual context of candidate answer entity. The weighted average of the vectors of these external neighboring nodes is used as external information representation of candidate answer entity $a$. To be more specific, $y_i$ denotes the co-occurrence frequency between candidate answer entity $a$ and $x_i \in \boldsymbol{D} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_m\}$. The external textual context of candidate answer entity $a$ is defined as:

$$n(a) = \{x_i \mid y_i > \theta\} \tag{1}$$

Here, the co-occurrence window is set as 5 and $\theta$ is the threshold used to filter out low frequency nodes. We use pre-trained word embedding matrix to obtain the embedding vectors of these external neighboring nodes. Then the external information representation of candidate answer entity $a_o$ can be calculated by:

$$a_o = \frac{1}{\sum_{x_i \in n(a)} y_i} \sum_{x_i \in n(a)} y_i x_i \tag{2}$$

## 3.3 Attention model

In order to comprehensively fuse internal information in the knowledge graph and external information in text corpus, we design an attention model to dynamically aggregate vectors. For each candidate answer $a$, the extent of attention should be measured by the relatedness between representation of input question q and different answer aspect embeddings $a_e, a_r, a_c, a_o$. We use the following formula to calculate the weights:

$$\alpha_i = \frac{\exp(\eta_i)}{\sum_{j \in \{e,r,c,o\}} \exp(\eta_j)} \tag{3}$$

$$\eta_i = f\left(W^T [\mathrm{q}; a_i] + b\right) \tag{4}$$

Here, $[m; n]$ denotes the connection of vector $m$ and vector $n$, $W^T \in \mathbb{R}^{2d \times d}$ is an intermediate matrix which is randomly initialized. $b$ is the offset and is also randomly initialized The intermediate matrix and offset are updated in the training process. We use q to define the vector representation of input question. $f(\cdot)$ is a non-linear activation function and $j \in \{e, r, c, o\}$ denotes different answer aspects. The similarity score of the question $q$ and candidate answer $a$ can be calculated as follows:

$$S(q, a) = \mathrm{h}(\mathrm{q}, \mathrm{a}) \tag{5}$$

$$\mathrm{a} = \sum_{i \in \{e,r,c,o\}} \alpha_i a_i \tag{6}$$

Here, $\mathrm{h}(\cdot)$ is the inner product between the question vector q and candidate answer vector a.

## 3.4 Model Training

For every question $q$, we first construct its correct answer set $R_q$ and wrong answer set $W_q$ according to its candidate answer set $W_q$. Then, for every correct answer in $R_q$, we construct negative examples by randomly selecting $k$ wrong answers in $W_q$. Here, $k$ is a hyper-parameter used to control the number of negative examples. We apply pairwise training in the training process and the loss function is defined as follows:

$$L\left(q, a, a'\right) = \left[m - S(q, a) + S\left(q, a'\right)\right]_+ \tag{7}$$

Here, $\mathrm{S}(\cdot)$ denotes the function used to calculate similarity score, $m$ denotes the margin parameter. $[z]_+$ denotes the maximum value between 0 and $z$. The whole objective function is defined as follows:

$$\min \sum_q \frac{1}{|R_q|} \sum_{a \in R_q} \sum_{a' \in W_q} L\left(q, a, a'\right) \tag{8}$$

In the training process, we apply stochastic gradient descent (SGD) based on minibatch as optimizer.

## 3.5 Inference

In the testing process, we calculate scores for all the answers $a \in C_q$ with the scoring function $S(q, a)$, since there may be more than one correct answer for many questions, it is inappropriate to choose the answer with the highest score as the final answer. The margin parameter $m$ is used to help solve this problem. We first find out the highest score of all the answers, the highest score $S_{\max}$ is defined as:

$$S_{\max} = \max_{a \in C_q} (S(q, a)) \tag{9}$$

Then the candidate answers whose scores are close to the highest score $S_{\max}$ are select as the final answers. To be more specific, we use the following formulas to determine the final outputs and the final outputs consist of the final answer set.

$$A_q = \left\{a \mid a \in C_q \text{ and } S_{\max} - S(q, a) < m\right\} \tag{10}$$

# 4 EXPERIMENTS

We conduct our experiments on the WebQuestions dataset, the WebQuestions dataset consists of 5810 question-answer pairs, in which 3778 question-answer pairs are for training and 2032 question-answer pairs are for testing. As for knowledge graph, we employ FB2M which is a subset of the huge knowledge graph Freebase that contains 647,657 entities, 4,641 relations and 1,604,874 triples. For WebQuestions, we further split the training part into a training set and a validate set with the percentage of 80% and 20%. The text corpus is generated from Wikipedia. Macro F1 score is used as the evaluation metrics and is calculated through the official evaluation script provided by Berant et al [2]. To evaluate our method, we use IR-based approaches proposed in recent years as baseline.

## 4.1 Settings

In the training process, we perform word embedding model and knowledge graph embedding model at first and the training process of question answering will directly use the embedding vectors output by them. We employ BERT [7] to obtain the word embedding matrix of text corpus and TransE is used to get the embedding vectors of entities and relations in the knowledge graph. The embedding dimension $d$ is set to 250. The embedding dimension of input question is also set to 250. Besides, we set the minibatch size to 50 and the learning rate is set to 0.01. The hyper-parameter $k$ and the margin parameter $m$ are set to 1000 and 0.7. Finally, we employ hyperbolic tangent as the activation function $f(\cdot)$.

**Table 1: Results on the WebQuestions dataset.**

| Methods | Macro F1 |
|---|---|
| Bordes et al., 2014 | 39.2 |
| Dong et al., 2015 | 40.8 |
| Bordes et al., 2015 | 42.2 |
| Hao et al., 2017 | 42.9 |
| Our approach | **43.7** |

**Table 2: Results of the model under different combinations of components.**

| Methods | Macro F1 |
|---|---|
| Transformer | 39.8 |
| Transformer + external information | 41.8 |
| Transformer + internal information | 42.5 |
| Transformer + internal and external information | 43.7 |

## 4.2 Performance Comparison

As is shown in Table 1, our method achieves higher F1 score on the WebQuestions dataset. On one hand, in terms of question representation learning, Bordes et al., 2014 [4] apply a bag-of-words model. Dong et al., 2015 [8] use multi-column convolutional neural network. Bordes et al., 2015 [5] apply memory network and Hao et al., 2017 [9] introduce Bi-LSTM network. The Transformer Encoder network we use can further capture the word-to-word dependencies within the input question and extract information of the input question more efficiently. On the other hand, in the answer end, Bordes et al., 2014 [4] use the subgraph of candidate answer to help represent candidate answer. Dong et al., [8] use answer path, answer type and answer context to represent candidate answer. Hao et al., 2017 [9] consider the impact of the question on the candidate answer aspects and use attention model to aggregate information in knowledge graph. We further utilize text information outside the knowledge graph and design an attention model to dynamically aggregate internal and external information. The answer aspects in the knowledge graph can effectively provide key information of candidate answer like the relation between question and candidate answer while the description textual context outside the knowledge graph can effectively supplement the context information of the candidate answer. At the same time, specially designed attention model can reasonably fuse these information.

## 4.3 Ablation Study

We also conduct comprehensive experiments to evaluate the components of our model. Table 2 shows the performance of the model under different combinations of components. Transformer means we only apply Transformer Encoder network to get the representation of question and the multiple aspect information of answer is not used. External information means the text corpus is used to get the context information outside the knowledge graph. Internal information means the text corpus is abandoned while answer aspects in the knowledge graph are utilized.

From the results in table 2, we draw observations as follows. First, when internal or external information is used alone, using internal information can achieve higher F1 score. This means the information in the knowledge graph is richer and more conducive to the representation of candidate answer. Then, the external description information screened by co-occurrence network can effectively supplement the context information of candidate answers. The comprehensive utilization of internal and external information can further improve the performance of the model and achieve the highest score.

# 5 CONCLUSION

In this work, we introduce a novel method that utilizes text information outside the knowledge graph to enhance the representation ability of candidate answers for the task of KGQA. Firstly, we use answer entity, answer relation and answer context to help represent answer information in the knowledge graph. Then, we employ co-occurrence network to find external neighboring nodes and these nodes constitute the external textual context of candidate answer entity. Finally, we design an attention model to dynamically fuse internal and external information. Besides, we apply Transformer Encoder network to conduct question representation learning in order to further capture word-to-word dependencies. The experiments on the WebQuestions dataset proves the effectiveness of our method.

# REFERENCES

[1] Abdalghani Abujabal, Mohamed Yahya, Mirek Riedewald, and Gerhard Weikum. 2017. Automated Template Generation for Question Answering over Knowledge Graphs. In *the 26th International Conference*.

[2] Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013. Semantic parsing on freebase from question-answer pairs. In *Proceedings of the 2013 conference on empirical methods in natural language processing*. 1533–1544.

[3] K. Bollacker. 2008. Freebase : A collaboratively created graph database for structuring human knowledge. *Proc. SIGMOD' 08* (2008).

[4] Antoine Bordes, Sumit Chopra, and Jason Weston. 2014. Question answering with subgraph embeddings. *arXiv preprint arXiv:1406.3676* (2014).

[5] Antoine Bordes, Nicolas Usunier, Sumit Chopra, and Jason Weston. 2015. Large-scale simple question answering with memory networks. *arXiv preprint arXiv:1506.02075* (2015).

[6] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Neural Information Processing Systems (NIPS)*. 1–9.

[7] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. (2018).

[8] Li Dong, Furu Wei, Ming Zhou, and Ke Xu. 2015. Question Answering over Freebase with Multi-Column Convolutional Neural Networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*.

[9] Yanchao Hao, Yuanzhe Zhang, Kang Liu, Shizhu He, and Jun Zhao. 2017. An End-to-End Model for Question Answering over Knowledge Base with Cross-Attention Combining Global Knowledge. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.

[10] Sen Hu, Lei Zou, Jeffrey Xu Yu, Haixun Wang, and Dongyan Zhao. 2017. Answering Natural Language Questions by Subgraph Matching over Knowledge Graphs. *IEEE Transactions on Knowledge and Data Engineering* (2017).

[11] Jayant Krishnamurthy and Tom M. Mitchell. 2012. Weakly Supervised Training of Semantic Parsers. In *Joint Conference on Empirical Methods in Natural Language Processing & Computational Natural Language Learning*.

[12] Percy Liang, Michael I. Jordan, and Dan Klein. 2013. Learning dependency-based compositional semantics. *Computational Linguistics* 39, 2 (2013), 389–446.

[13] Md Mostafizur Rahman and Atsuhiro Takasu. 2020. Leveraging Entity-Type Properties in the Relational Context for Knowledge Graph Embedding. *IEICE Transactions on Information and Systems* E103.D, 5 (2020), 958–968.

[14] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention Is All

You Need. *arXiv* (2017).

[15] Zhigang Wang, Juanzi Li, Zhiyuan Liu, and Jie Tang. 2016. Text-enhanced representation learning for knowledge graph. In *Proceedings of International Joint Conference on Artificial Intelligent (IJCAI)*. 4–17.

[16] Yuk Wah Wong and Raymond J. Mooney. 2007. Learning Synchronous Grammars for Semantic Parsing with Lambda Calculus. In *Acl, Meeting of the Association for Computational Linguistics, June, Prague, Czech Republic*.

[17] Ruobing Xie, Zhiyuan Liu, Maosong Sun, et al. 2016. Representation Learning of Knowledge Graphs with Hierarchical Types.. In *IJCAI*. 2965–2971.

[18] Kun Xu, Yansong Feng, Songfang Huang, and Dongyan Zhao. 2016. Hybrid question answering over knowledge base and free text. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*. 2397–2407.

[19] Kun Xu, Siva Reddy, Yansong Feng, Songfang Huang, and Dongyan Zhao. 2016. Question answering on freebase via relation extraction and textual evidence. *arXiv preprint arXiv:1603.00957* (2016).