SYNTAX-BASED GRAPH MATCHING FOR KNOWLEDGE BASE QUESTION ANSWERING

Lu Ma[1,2]*, Peng Zhang*[1]*, Dan Luo*[1,2]*, Xi Zhu*[1,2]*, Meilin Zhou*[1]*, Qi Liang*[1]*, Bin Wang*[3]

[1]Institute of Information Engineering, Chinese Academy of Sciences
[2]School of Cyber Security, University of Chinese Academy of Sciences
[3]Xiaomi AI Lab
{malu,pengzhang,luodan,zhuxi,zhoumeilin,liangqi}@iie.ac.cn, wangbin11@xiaomi.com

ABSTRACT

Semantic parsing is a mainstream method of knowledge base question answering task that first generates a set of logical forms according to question and knowledge base (KB), and then selects the most matching one to get answers. However, existing selection methods are usually based on word-level matching, which cannot capture the structural information or solve the long-term dependency problem of entities. To solve this problem, we propose a syntax-based graph matching method, which explicitly models both question and logical form as graphs, and performs matching at both word-level and structure-level. The multi-level matching strategy not only captures the structural and semantic relations between entities but also explores their intrinsic relations. Such a design can greatly minimize the gap between question and most relevant knowledge from KB, and hence can reason more accurately. We conduct extensive experiments on several benchmarks and demonstrate the effectiveness of our proposed method.

Index Terms— Question answering, knowledge base

1. INTRODUCTION

Recently, Question Answering over Knowledge Base (KB-QA) [1] has been well developed. KB-QA takes a natural language question as input and automatically returns a set of entities from the knowledge base (KB) as the answers. For example, a question like this: "*Who did Cristiano Ronaldo play for in 2010?*" Its answer *real madrid c.f.* is an entity from Freebase [2], which is used as the supporting KB.

Semantic parsing and information retrieval are two mainstream methods for KB-QA. This paper focuses on semantic parsing, which converts questions into logical forms. A logical form (e.g., query graph) can be directly transformed into a query (e.g., SPARQL [3]), which can be executed over KB to obtain answers. Early methods [4, 5] construct several candidate query graphs for a question by three steps (entity linking, relation extraction, and constraint attachment) independently. During relation extraction, they select the optimal relation by calculating the similarity of the word sequences of question and relations. Following [4], recent works [6–9] apply deep learning to construct query graphs, and represent query graphs better based on neural networks to select the optimal query graph. These methods generally perform matching by modeling the question and logical form as individual sequences, while cannot capture the structural information or solve the long-term dependency problem of entities. As shown in Fig. 1 (a), the query graph is a set of entities and relations from a path starts with *Barack Obama* in KB. We split the question and the query graph into word sequences respectively. The structural and semantic relations between entities will be lost when only computing the word-level matching score, especially for complex questions. Thus, additionally matching question and query graph in the modality of KB is necessary, see Fig. 1 (b). (cvt is a Compound Value Types node, which does not indicate real-world entities in Freebase.)

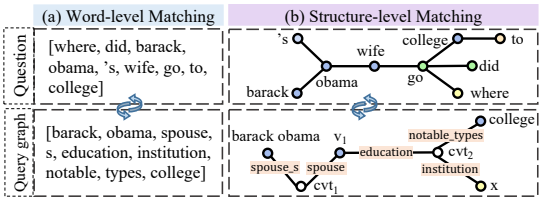


Fig. 1: Example of question "*Where did Barack Obama's wife go to college?*". (1) Word-level: represent question and query graph as individual sequences. (2) Structure-level: model the graph structure of question and query graph.

To solve the above issue, we propose a syntax-based graph matching method by capturing structural information and deep semantic relationships between question and query graph. Different from [10] that only learns the structure of query graph and [7, 11] that model question as syntactic structure only for relation extraction stage, we explicitly model question as syntactic graph and logical form as query graph simultaneously. A *syntactic graph* can represent syntactic dependencies in question, while a *query graph* can model the critical topological information in KB. We also propose a multi-level matching method for exactly knowledge rea-

Peng Zhang is the corresponding author

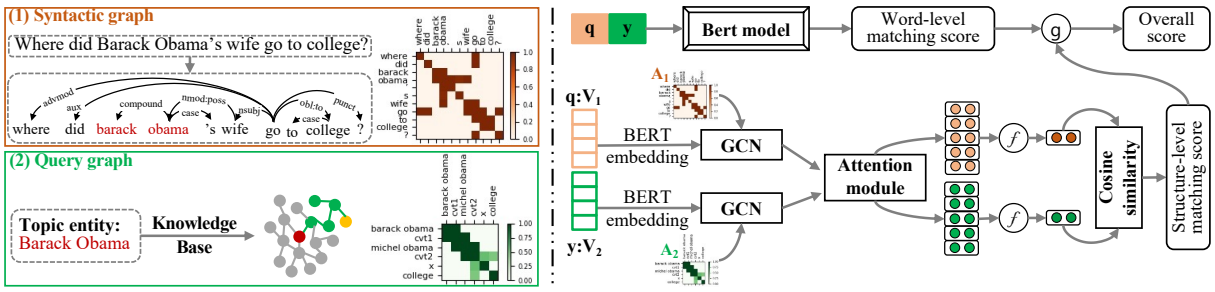


Fig. 2: Framework of our approach. Orange box: The syntactic graph construction. Green box: The query graph construction. The right part is our syntax-based graph matching method for KB-QA task. g is average operation. f is *max*-pooling method.

soning, which comprises a word-level matching module that associates each word in question with its relevant knowledge in KB, and a structure-level matching module that associates structure information in question and KB. Such a design encourages the complementation of semantic and structural relations and further boosts the performance of KB-QA task.

In our structure-level matching module, we first construct a syntactic graph by the probability matrix of all dependency arcs from LAL-Parser [12]. Compared with a discrete matrix, the probability matrix representing dependencies between words contains rich syntactic information. Reddy et al. [13, 14] have demonstrated that the dependency information helps understand the structural information of the question. Likewise, we score each edge in the query graph by computing the similarity between question and the corresponding relation, which can weigh the information in KB. We then apply Graph Convolutional Networks (GCN) [15] to model these two graphs. Previous works [16, 17] demonstrated the effectiveness of GCN for graph representation of question answering tasks. Moreover, motivated by the BiAffine of [18], we design an attention module to bridge the relevant information between syntactic graph and query graph.

In summary, the contributions of this paper are as follows.

(1) We explicitly model question and logical form as graphs to capture deep structural information for KB-QA. To the best of our knowledge, this is the first work to model both question and logical form at graph level.

(2) We propose a novel Syntax-based Graph Matching framework (SGM) for complex KB-QA task, which models both word-level information of question and candidate path and structure-level information of syntactic graph and query graph, and hence obtains enriched representations to measure the similarity of question and query graph.

(3) Experiments on benchmarks show that SGM achieves the new state-of-the-art performance for the KB-QA task.

2. METHODS

The architecture of our proposed approach is illustrated in Fig.2, it mainly consists of a word-level matching module (2.1) and a structure-level matching module (2.2) that jointly capture the semantic information in question and query graph.

2.1. Word-level Matching

We define the KB as a graph $\mathcal{K} = (\mathcal{E}, \mathcal{R})$, where $\mathcal{E}$ denotes the set of entities and $\mathcal{R}$ denotes the set of relations that connect entities in $\mathcal{E}$. Given a question q, following prior work [6], we first link the entity mentioned in q to KB and choose the topic entity V_q, and then apply the existing mainstream method, such as beam search, to generate candidate paths $\mathcal{Y}$ iteratively. A set of candidate paths $\mathcal{Y}_n = \{y | y = (E, R), E \subseteq \mathcal{E}, R \subseteq \mathcal{R}\}$ is obtained, where y comprises all entities and relations on one of the n-hop paths of V_q at the n-th iteration.

The word-level matching module captures the semantic information between sequences of question and candidate path. Similar to [6], we construct a sequence pair $[q, y]$ as input, and then use BERT to process it and derive a score S^T that measure the semantic similarity between q and y:

$$S^T(y, q) = f\left(\text{concat}\left(\mathbf{h}_{fe}, f_{head}\left(f_{enc}\left([q, y]\right)\right)\right)\right) \quad (1)$$

where $f_{head} \circ f_{enc}$ denotes the probability of the candidate path computed by BERT, $\mathbf{h}_{fe}$ denote several feature vectors. f is a full-connected layer. $\text{concat}(\cdot)$ denotes concatenation.

2.2. Structure-level Matching

To associate structure information from question and KB, we explicitly model question as syntactic graph and candidate path as query graph by GCN, and design an attention module to bridge the relevant information between these two graphs.

2.2.1. Graph construction

Syntactic graph. It is observed that two words in a question usually hold certain relations that can be identified by a dependency parser. Thus, we propose to model the syntactic structure of question to capture its deep semantic information.

Formally, we construct a syntactic graph $G_{syn} = (V_1, E_1)$ for a question by treating each word as a node and each dependency relation between two words as an edge. We use a

Table 1: Average F_1 scores of models on WQSP and CQ test sets. * refers to our re-implementation. The bests are in bold.

Method	WQSP	CQ
STAGG [4]	69.00	36.89
MulCG [20]	-	40.94
UHop [21]	68.50	35.30
TextRay [11]	60.30	-
Topic Units [22]	67.90	-
SeqM [6]*	71.83	40.55
SGM (ours)	**72.36**	**41.38**

dependency probability matrix to alleviate dependency errors via LAL-Parser. Finally, we obtain a set of nodes V_1 and a weighted adjacent matrix A_1 as shown in Fig. 2 (1).

Query graph. Candidate path completely consists of entities and relations in KB, which should contain abundant structure information among entities, but is omitted in word level. Therefore, we construct a query graph for each candidate path to model the structure information in KB.

Formally, we construct a query graph $G_{qry} = (V_2, E_2)$, $G_{qry} \subseteq \mathcal{K}$ for a candidate path by treating each entity as a node and each relation between entities as an edge. As shown in Fig. 2 (2), the candidate path is a 2-hop path of *Barack Obama*, and the corresponding query graph comprises topic entity (red node), variable entities (green nodes), relations (green edges) and aggregation function (e.g. *argmin*) if exists. For purpose of capturing the importance of each relation to the question, we embed relations and question by GloVe[19], and calculate their cosine similarity as weight. We obtain a set of nodes V_2 and a weighted adjacent matrix A_2.

2.2.2. Graph matching module

Graph Convolutional Network (GCN). Through the message passing of multilayer GCNs [15], each node in a graph can integrate rich information about its neighborhoods. We separately capture the semantic structural information of syntactic graph and query graph by GCNs.

Taking G_{qry} for example, we first tokenize the label of the i-th node into m tokens and map each token to a d_w-dimensional word embedding using the BERT model, and then get the embeddings of the i-th node: $\mathbf{e}_i \in \mathbb{R}^{m \times d_w}$. We initialize the hidden states for the i-th node via: $\mathbf{h}_i^{(0)} = \mathbf{e}_i$. At the l-th layer, the i-th node representation is updated by:

$$\mathbf{h}_i^{(l)} = \text{RELU}\left(\sum\nolimits_{j=1}^{|V_2|} A_{ij} W_l \mathbf{h}_j^{(l-1)} + b^l\right) \tag{2}$$

where d is the dimension of hidden state, W_l is a weight matrix, b^l is a bias term. We stack such networks for L_2 layers and obtain representations of all nodes $H_{qry} \in \mathbb{R}^{|V_2| \times d}$.

Likewise, we utilize L_1-layer GCNs to represent syntactic graph G_{syn}, and obtain the representations $H_{syn} \in \mathbb{R}^{|V_1| \times d}$.

Table 2: Test results of ablation study. The bests are in bold.

Method	WQSP test			CQ test		
	P@1	Acc	F_1	P@1	Acc	F_1
SGM	**72.12**	**66.38**	**72.36**	**40.63**	**31.63**	**41.38**
w/o SyP	71.45	65.47	71.49	40.00	29.75	40.64
w/o ReP	70.53	65.28	70.95	39.00	30.25	40.42
w/o [SyP,ReP]	71.14	65.77	71.57	38.38	30.38	39.58
w/o Att	71.69	65.47	71.06	39.00	30.13	39.99
w/o [SyP,ReP,Att]	71.87	66.26	71.61	37.75	29.63	39.18

Attention module. We adopt a BiAffine transformation to bridge the relevant information between syntactic graph and query graph. The process is as follows:

$$\hat{H}_{syn} = \text{softmax}\left(H_{syn} W_1 H_{qry}^T\right) H_{qry} \tag{3}$$

$$\hat{H}_{qry} = \text{softmax}\left(H_{qry} W_1 H_{syn}^T\right) H_{syn} \tag{4}$$

where W_1 and W_2 are trainable parameters.

Finally, we apply the max-pooling operation on nodes of G_{syn} and G_{qry} to obtain the final representations of the two graphs: $\mathbf{H}_{syn} = max\left(\hat{H}_{syn}\right)$, $\mathbf{H}_{qry} = max\left(\hat{H}_{qry}\right)$, and use cosine similarity to compute the semantic similarity of each (syntactic graph, query graph) pair $S^G(y, q)$.

2.3. Learning Objectives

Finally, the overall matching score S^F is the average of the obtained matching scores (S^T and S^G). Hence, the candidate path with the maximum S^F is transformed into a SPARQL query, which can be executed over KB to retrieve the answers. In the training data, each question has a set of correct answers. Therefore, we use weak supervision as suggested in [23] to train our model. Given a question and a set of candidate paths $\mathcal{Y}_n$ at the n-th iteration, $n \in [1, N]$, we use REINFORCE algorithm to learn a policy function $P_\theta(y|q)$ that maximizes the expected reward following [6]:

$$P_\theta(y|q) = \mathbb{E}_{C^{(1)},\ldots,C^{(N-1)} \sim \pi_\theta}[R(N)] \tag{5}$$

$$C^{(n)} \sim Categorical(d^{(n)}) \tag{6}$$

$$d^{(n)} = (1 - \lambda) * \text{softmax}(S^F) + \lambda * F_1^{(n)} \tag{7}$$

where θ is the set of parameters we want to learn. Reward function $R(N)$ is the F_1 score of the final predicted answers and the ground truth. C is the selected candidate path. d is the probability of each candidate path. λ is a hyper-parameter.

3. EXPERIMENT

3.1. Experimental setup

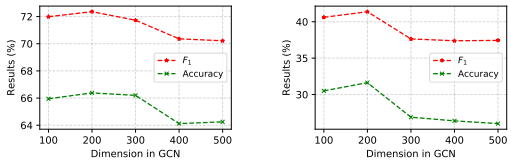
To evaluate the performance of our proposed approach on KB-QA task, we carried out experiments on ComplexQuestions (CQ) [20] and WebQuestionSP (WQSP) [24]. Freebase

Table 3: Qualitative results for KB-QA on WQSP test. GT denotes ground truth.

q	What part of the country is ohio in?		What year was first world series?	
	Candidate path	Answer	Candidate path	Answer
SeqM*	m.05kkh country ?e1	United States of America	m.0fjp3 current_frequency ?e1	Yearly
SGM	m.05kkh containedby ?e1 ?e1 notable_types m.01nt	Midwestern United States East North Central States	m.0fjp3 instances ?e1 first ?e1 start_date 1903-10-01	1903 World Series
GT	m.05kkh containedby ?e1 ?e1 notable_types m.01nt	Midwestern United States East North Central States	m.0fjp3 instances ?e1 first ?e1 start_date 1903-10-01	1903 World Series

Table 4: Average F_1 scores of different GCN layer numbers.

Layers	1	2	3	4
WQSP test	71.28	**72.36**	72.22	70.96
CQ test	40.14	**41.38**	38.31	38.78



(a) WQSP test (b) CQ test

Fig. 3: Results by varying embedding dimensions in GCN.

[2] is used as the knowledge base. We adopt Precision@1 (P@1), accuracy (Acc) and averaged F_1 score (F_1) as evaluations following [4, 6, 7, 11, 23]. We implement our method in PyTorch [25]. The initial learning rate is set as 0.00002. The number of iterations N is set as 2. d_w and d are set as 768 and 200, respectively. L_1, L_2 are both set as 2.

3.2. Experimental results

Main results. We compare our proposed approach with the state-of-the-art on benchmarks. As shown in Table 1, our SGM model outperforms the state-of-the-art on all the benchmarks, which showcases that the structure-level matching module effectively associates structural information in question and KB, and improves the performance of KB-QA task. SeqM model achieves the state-of-the-art performance with F_1 scores of 74.0% on WQSP and 43.3% on CQ reported in the original paper [6]. But we failed to reproduce such performances. We re-implement it as SeqM* and obtain F_1 scores of 71.83% on WQSP and 40.55% on CQ. SGM framework is inspired by the reproduced SeqM* model so that we report these reproduced performances. Based on SeqM* model, our SGM method achieves the new state-of-the-art performances, and improves (**0.53%**) on WQSP and (**0.83%**) on CQ.

Ablation study. As shown in Table 2, we design ablation studies to evaluate the effectiveness of each component of our model. SGM w/o SyP, SGM w/o ReP indicate that we use a discrete adjacent matrix for syntactic graph and query graph, respectively. SGM w/o Att indicates that we remove the attention module. Results suggest that the probability matrix for G_{syn} contains more syntactic knowledge, and the probability matrix for G_{qry} captures the structure in KB more precisely. The performance degrades substantially when the attention module is removed, the reason is that the mutual relations between the graphs are missing. It is observed that SGM w/o [SyP,ReP,Att] outperforms SGM w/o SyP, SGM w/o ReP and SGM w/o Att on WQSP. It shows that the cooperation of SyP, ReP and Att is important for more complex questions.

Parameters. We investigate the effect of the layer number of GCNs as shown in Table 4. SGM model achieves the best performance when GCNs have 2 layers. Fig. 3 depicts the performances with different hidden sizes in GCNs. Results drop when the dimension is larger than 200.

Qualitative results. In Table 3, we offer two examples on KB-QA using our SGM method and SeqM* [6], which is the closest to our work. For a question and a candidate path, it is observed that when the similarity between word sequences of them is high, they are not semantically similar. SGM can address this issue by matching them in structure level.

4. CONCLUSIONS

In this work, we present a syntax-based graph matching method (SGM) to model both word-level and structure-level information of question and KB for KB-QA task. A syntactic graph is employed to model the syntactic structure of question, and a query graph is constructed to model the topological structure in KB. SGM architecture can successfully capture the structural and semantic relations between question and candidate path, and find most relevant knowledge to get answer. Experiments demonstrate that our approach outperforms the state-of-the-art on benchmarks.

5. ACKNOWLEDGEMENT

This work is supported by the National Natural Science Foundation of China (grant No. 61876223), Youth Innovation Promotion Association, Chinese Academy of Sciences (No. 2020163), National Natural Science Foundation of China (grant No. 61832004), Projects of International Cooperation and Exchanges NSFC (grant No. 62061136006).

References

[1] Christina Unger, André Freitas, and Philipp Cimiano, "An introduction to question answering over linked data," in *Reasoning Web*, 2014, pp. 100–140.

[2] Kurt D. Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor, "Freebase: A collaboratively created graph database for structuring human knowledge," in *SIGMOD*, 2008.

[3] Prudhommeaux Eric, *SPARQL query language for RDF*, 2008.

[4] Wen-tau Yih, Ming-Wei Chang, Xiaodong He, and Jianfeng Gao, "Semantic parsing via staged query graph generation: Question answering with knowledge base," in *ACL-IJCNLP*, 2015, pp. 1321–1331.

[5] Wen-tau Yih, Xiaodong He, and Christopher Meek, "Semantic parsing for single-relation question answering," in *ACL: Short Papers*, 2014, pp. 643–648.

[6] Yunshi Lan and Jing Jiang, "Query graph generation for answering multi-hop complex questions from knowledge bases," in *ACL*, 2020, pp. 969–974.

[7] Kangqi Luo, Fengli Lin, Xusheng Luo, and Kenny Zhu, "Knowledge base question answering via encoding of complex query graphs," in *EMNLP*, 2018, pp. 2185–2194.

[8] Gaurav Maheshwari, Priyansh Trivedi, Denis Lukovnikov, Nilesh Chakraborty, Asja Fischer, and Jens Lehmann, "Learning to rank query graphs for complex question answering over knowledge graphs," in *ISWC*, 2019, pp. 487–504.

[9] Shuguang Zhu, Xiang Cheng, and Sen Su, "Knowledge-based question answering by tree-to-sequence learning," *Neurocomputing*, vol. 372, pp. 64 – 72, 2020.

[10] Daniil Sorokin and Iryna Gurevych, "Modeling semantics with gated graph neural networks for knowledge base question answering," in *COLING*, 2018, pp. 3306–3317.

[11] Nikita Bhutani, Xinyi Zheng, and H V Jagadish, "Learning to answer complex questions over knowledge bases with query composition," in *CIKM*, 2019, p. 739–748.

[12] Khalil Mrini, Franck Dernoncourt, Quan Hung Tran, Trung Bui, Walter Chang, and Ndapa Nakashole, "Rethinking self-attention: Towards interpretability in neural parsing," in *Findings: EMNLP*, 2020, pp. 731–742.

[13] Siva Reddy, Oscar Täckström, Michael Collins, Tom Kwiatkowski, Dipanjan Das, Mark Steedman, and Mirella Lapata, "Transforming dependency structures to logical forms for semantic parsing," *TACL*, vol. 4, pp. 127–140, 2016.

[14] Siva Reddy, Oscar Täckström, Slav Petrov, Mark Steedman, and Mirella Lapata, "Universal semantic parsing," in *EMNLP*, 2017, pp. 89–101.

[15] Thomas N. Kipf and Max Welling, "Semi-supervised classification with graph convolutional networks," in *ICLR*, 2017.

[16] Nicola De Cao, Wilker Aziz, and Ivan Titov, "Question answering by reasoning across documents with graph convolutional networks," in *NAACL-HLT*, 2019, pp. 2306–2317.

[17] Yu Cao, Meng Fang, and Dacheng Tao, "BAG: Bi-directional attention entity graph convolutional network for multi-hop reasoning question answering," in *NAACL-HLT*, 2019, pp. 357–362.

[18] Hao Tang, Donghong Ji, Chenliang Li, and Qiji Zhou, "Dependency graph enhanced dual-transformer structure for aspect-based sentiment classification," in *ACL*, 2020, pp. 6578–6588.

[19] Jeffrey Pennington, Richard Socher, and Christopher Manning, "GloVe: Global vectors for word representation," in *EMNLP*, 2014, pp. 1532–1543.

[20] Junwei Bao, Nan Duan, Zhao Yan, Ming Zhou, and Tiejun Zhao, "Constraint-based question answering with knowledge graph," in *COLING: Technical Papers*, 2016, pp. 2503–2514.

[21] Zi-Yuan Chen, Chih-Hung Chang, Yi-Pei Chen, Jijnasa Nayak, and Lun-Wei Ku, "UHop: An unrestricted-hop relation extraction framework for knowledge-based question answering," in *NAACL-HLT*, 2019, pp. 345–356.

[22] Yunshi Lan, Shuohang Wang, and Jing Jiang, "Knowledge base question answering with topic units," in *IJCAI*, 2019, pp. 5046–5052.

[23] Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang, "Semantic parsing on Freebase from question-answer pairs," in *EMNLP*, 2013, pp. 1533–1544.

[24] Wen-tau Yih, Matthew Richardson, Chris Meek, Ming-Wei Chang, and Jina Suh, "The value of semantic parse labeling for knowledge base question answering," in *ACL:Short papers*, 2016, pp. 201–206.

[25] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer, "Automatic differentiation in pytorch," 2017.