Attention-based Multi-hop Reasoning for Knowledge Graph

Zikang Wang*†, Linjing Li*, Daniel Dajun Zeng*†, Yue Chen‡
*The State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing, China
†School of Computer and Control Engineering, University of Chinese Academy of Sciences, Beijing, China
‡CNCERT/CC, Beijing, China
{wangzikang2016, linjing.li, dajun.zeng}@ia.ac.cn, chenyue@cert.org.cn

***Abstract*—Knowledge graph plays an important role in detection, prediction, early warning, and other security related applications. A fundamental task in applying knowledge graph is the so-called multi-hop reasoning, which focuses on inferring new relations between entities. In this paper, we introduce attention mechanism to the classic compositional method. After finding reasoning paths between entities, we aggregate these paths' embeddings into one according to their attentions, and infer the relation of entities based on the combined embedding. Two experiments on NELL-995 dataset, fact prediction and link prediction, validated that our method outperforms all baselines.**

***Index Terms*—knowledge graph, multi-hop reasoning, attention mechanism, random walking**

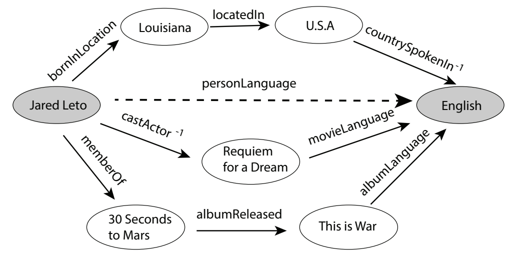

Fig. 1: Multi-hop Reasoning in Knowledge Graph

I. Introduction

Knowledge graph is a large-scale knowledge base containing entities and their relations. The semantic associations stored in knowledge graph are very essential and helpful in varies applications. As reasoning on knowledge graph can help discover unknown relations between entities, it can be used to reveal connections between seemingly disparated people, places, and events [1], which are key roles in security related tasks, such as detection, prediction, and early warning.

Multi-hop reasoning is one of the basic problems in knowledge graph, it aims to make inference on entities and relations not directly stored in the given knowledge graph based on stored knowledge [2]. For example, as shown in Fig 1, there are three paths between entity "Jared Leto" and "English", each consists of multiple entities and relations, The goal of multi-hop reasoning is to infer new relation "personLanguage" between these two entities based on the known paths. Existing methods for multi-hop reasoning include symbolic methods, such as Path Ranking Algorithm (PRA) [3], compositional methods like Path-RNN [4], and reinforcement learning methods like DeepPath [5] and MINERVA [6].

In this paper, we present an approach for multi-hop reasoning based on compositional methods. Existing compositional methods calculate each path's similarity score with the target relation separately, then simply combine these scores by averaging or summing. However, as showing in Fig 1, different paths may related with the target relation in a different level. Thus, simply applying the average or summation operation may cause loss of knowledge. To utilize the disparate knowledge better, we introduce attention mechanism into the compositional model. To be more specific, we first find multiple paths between entities by random walking, and compute the attention for each path based on its semantic similarity with the query. Then we aggregate the representations of these paths according to their attention, final result is formed based on the similarity of this aggregated representation and the query. Two experiments, fact prediction and link prediction, on dataset NELL-995 validate the effectiveness of the proposed method.

II. Related Work

Several methods have been proposed to accomplish the reasoning task on knowledge graph. One of the most popular approach determines whether a triple is true based on the similarity score of embeddings. This approach includes TransE [7], TransH [8], etc. However, these methods cannot be employed to address problems involving multi-hop relation paths as they can only deal with triples. Path-Ranking Algorithm (PRA) [3] is the first algorithm which uses random walking to find multiple paths and reasons relations. Path-RNN [4] is the first compositional method, it reasons relations of entities based on paths between them, However, path-RNN only considers relations on paths. Das *et al.* [2] improved Path-RNN by adding entities into consideration and introducing much better score pooling function. Besides, reinforcement learning methods have also been explored, such as DeepPath [5] and MINERVA [6]. They model multi-hop reasoning as a Markov Decision Process and perform reasoning using reinforcement learning.

978-1-5386-7848-0/18/$31.00 ©2018 IEEE

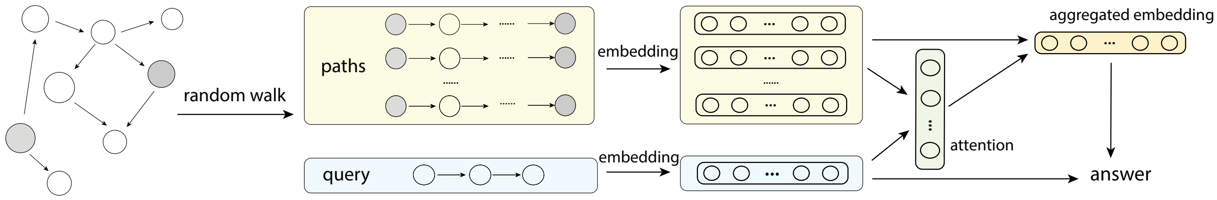

Fig. 2: Overall Architecture of Our Approach

III. Methodology

The whole architecture of our proposed approach is shown in Fig 2. In the first phase, for each triple (h, r, t), multiple reasoning paths can be found between head entity h and tail entity t through random walking. These paths usually include more than one relation or entity, so are referred to as "multi-hop" paths. In general, every path is semantically related to the target relation "r" in a different level, while complementary to each other as well. To bring more flexibility and diversity to the path finding process, we also take the inverse r^{-1}, from tail entity t to head entity h, of relation r into consideration. Thus, for every triple (h, r, t), we add a triple (t, r^{-1}, h) into the knowledge graph, which allows each entity can be visited several times if required.

For a triple (h, r, t), suppose we found n paths, each consists of several entities and relations, denote the i-th path as p_i, p_i can be represented as a list of ordered relations and entities:

$$p_i = h, r_1, e_1, r_2, e_2, \ldots, r_m, t. \tag{1}$$

Where path p_i starts from the head h, passes through entity $e_1, e_2, \ldots$ via relations $r_1, r_2, \ldots, r_m$ respectively, finally, arrives at the tail t. The set of all found paths for this triple is denoted as $S = p_1, p_2, \ldots, p_n$.

For a query, say q, we also represent it by entities and relations, as shown in Fig 2. For fact prediction task, the query can be denoted in form $q_i = (h, r, t)$, which is a path with two entities and a single relation (The output for fact prediction task will be $1/0$, which indicates the triple is true or false respectively). For other tasks, the resulting forms of queries can still be denoted as paths consisting of entities and relations, but may be different from each other. This structure not only provides a unified architecture to solve different tasks which increases flexibility of the model, but also offers the opportunity to guide the learning process by the query to achieve better results.

To better represent the semantic knowledge of different paths, we initialize the entity and relation representations with pretrained embeddings. Pretraining usually uses fundamental representation learning methods, such as TransE [7], TransH [8], etc. Next, we compute the attention for each path based on embeddings. Then we combine these paths based on their attentions to get an aggregated embedding through average or summation of the paths' embeddings. The final answer to a specific task can be obtained based on this aggregated embedding and the query's embedding.

Let A denote the pretrained embedding matrix, then the representation for path p_i can be computed as

$$r_i = \sum_j A p_{ij}, \tag{2}$$

where j runs through all entities and relations included in path p_i. To encode the order information into representations, we add position encoding to the representation r_i:

$$i = \sum_j l_j \cdot A p_{ij}, \tag{3}$$

where l_j is a vector [9] with the following definition

$$l_{jk} = (1 - j/J) - (k/d)(1 - 2j/J), \quad k = 1, 2, \cdots, J \tag{4}$$

where J is the total number of entities and relations in the path. The query will be represented in the same way.

As different paths carry different knowledge, each related to the query in a different level. Thus, it is nature to give each path a different attention while combing them. For query q, attention a_i for path p_i can be calculated as

$$a_i = \text{softmax}(q^T r_i). \tag{5}$$

Combining different paths according to their attentions, then we can get the aggregated representation p as:

$$p = \sum_i a_i p_i. \tag{6}$$

After getting the aggregated representation of different paths, we can construct the final prediction with respect to the query q by

$$o = \text{softmax}(W(p + q)), \tag{7}$$

where W is the weight matrix, output o is the final prediction respecting to query q. Besides, the similarity score s of the aggregated path and target relation r is

$$s = \text{normalize}(W(p + q)), \tag{8}$$

again W is the weight matrix.

For each query, the loss is defined as the crossing entropy between the ground truth and our model's prediction, we then minimize the loss function with stochastic gradient descent (SGD) algorithm.

TABLE I: Dataset Statistics

Dataset	#Entities	#Relations	#Triples	#Tasks
NELL-995	75492	200	154213	12

IV. Results

In this section, we evaluate the proposed approach through two kinds of experiments, fact prediction and link prediction, on the NELL-995 dataset, the 995-th iteration of the NELL system [5]. The statistics of the dataset is shown in Table I, it includes 12 reasoning tasks, each focuses on a different relation and corresponds to multiple reasoning paths.

A. Baselines and Implementation Details

For fact prediction, we use translation based representation methods [7], [8], [10] and DeepPath [5] as baselines. In this experiment, we make direct comparison with results reported by Xiong et al. [5]. As for link prediction, we compare our methods with PRA [3] and translation based methods [7], [8].

For both experiments, we use Mean Average Precision (MAP) as criterion. Triples in the original knowledge base are treated as "golden" ones, while triples constructed by replacing their head or tail entities are "corrupted" ones, for each golden triple, there may exist several corrupted triples in the test dataset. MAP evaluates whether similarity scores for golden triples are higher than corrupted ones. A higher MAP value indicates better performance. In both experiments, we limit the path number found for each triple to no more than 6, and initialize entity and relation embeddings with embeddings pretrained by TransE [7].

B. Fact Prediction

For a given triple (h, r, t), fact prediction task aims to decide whether a triple is true or false. Experimental results are shown in table Table II. We report two kinds of results of our method in the table, the one without entities refers to the experiment where only relations are involved. During this experiment, entities are not included in paths and representations. Entities are also not included in the query, query is represented as target relation directly, e.g., $q = r$. For the experiment with entities, entities and relations are both considered in paths and query. From table II, we can conclude that our method is better than all baselines and the performance is much better when both entities and relations are taken into account.

TABLE II: Results of Fact Prediction

Methods	TransE	TransH	TransR	TransD	DeepPath
MAP(%)	49.3	38.3	38.9	40.6	41.3
Methods	Our model(without entities)			Our model(with entities)	
MAP(%)	49.6			**52.5**	

TABLE III: Results of Link Prediction

Methods	PRA	TransE	TransR	TransH	Our Model
MAP(%)	67.5	75	74	75.1	**76.5**

C. Link Prediction

Link prediction is a task first proposed in [11] to predict the missing entities in triples. It has been widely used to evaluate the representation learning and reasoning methods. Both entities and relations are considered in this task, results are shown in Table III, from which we can observe that our model get better results on MAP.

V. Conclusions and Future Work

This paper proposed a new multi-hop reasoning method for knowledge graph based on compositional methods. By introducing attention mechanism into the model, knowledge dispersed among all the paths between entities is effectively exploited. We applied our method in both fact and link prediction applications, the experiments validated the effectiveness of the devised method. In our ongoing works, to better capture the semantic knowledge, we will use RNN or LSTM to get the representation of paths. Another direction is using memory mechanism to accomplish larger storage of knowledge.

Acknowledgments

This work was supported in part by the National Key Research and Development Program of China under Grant 2017YFC0820105 and 2016QY02D0305, the National Natural Science Foundation of China under Grants 71702181, 71621002, and U1435221, as well as the Key Research Program of the Chinese Academy of Sciences under Grant ZDRW-XH-2017-3.

References

[1] A. Sheth, B. Aleman-Meza, I. B. Arpinar, C. Bertram, Y. Warke *et al.*, "Semantic association identification and knowledge discovery for national security applications," *Journal of Database Management (JDM)*, vol. 16, no. 1, pp. 33–53, 2005.

[2] R. Das, A. Neelakantan, D. Belanger, and A. McCallum, "Chains of reasoning over entities, relations, and text using recurrent neural networks," in *ACL*, 2017, pp. 132–141.

[3] N. Lao and W. W. Cohen, "Relational retrieval using a combination of path-constrained random walks." *Machine Learning*, vol. 81, pp. 53–67, 2010.

[4] A. Neelakantan, B. Roth, and A. McCallum, "Compositional vector space models for knowledge base completion," in *ACL*, 2015.

[5] W. Xiong, T. Hoang, and W. Y. Wang, "Deeppath: A reinforcement learning method for knowledge graph reasoning," *arXiv preprint arXiv:1707.06690*, 2017.

[6] R. Das, S. Dhuliawala, M. Zaheer, L. Vilnis *et al.*, "Go for a walk and arrive at the answer: Reasoning over paths in knowledge bases with reinforcement learning," *arXiv preprint arXiv: 1711.05851*, 2017.

[7] A. Bordes, N. Usunier, A. Garcia-Duran *et al.*, "Translating embeddings for modeling multi-relational data," in *NIPS*, 2013, pp. 2787–2795.

[8] Z. Wang, J. Zhang, J. Feng, and Z. Chen, "Knowledge graph embedding by translating on hyperplanes," in *AAAI*, 2014, pp. 1112–1119.

[9] S. Sukhbaatar, A. Szlam, J. Weston, and R. Fergus, "End-to-end memory networks," in *NIPS*, 2015, pp. 2440–2448.

[10] G. Ji, S. He, L. Xu, K. Liu, and J. Zhao, "Knowledge graph embedding via dynamic mapping matrix," in *ACL (1)*, 2015, pp. 687–696.

[11] A. Bordes, J. Weston, R. Collobert, and Y. Bengio, "Learning structured embeddings of knowledge bases," in *AAAI*, 2011, pp. 301–306.