Building Knowledge Subgraphs in Question Answering over Knowledge Graphs

Web Engineering - 22nd International Conference, ICWE 2022, Proceedings

Aghaei, Sareh; Angele, Kevin; Fensel, Anna

https://doi.org/10.1007/978-3-031-09917-5_16

Building Knowledge Subgraphs in Question Answering over Knowledge Graphs

Sareh Aghaei[1(✉)], Kevin Angele[1], and Anna Fensel[1,2]

[1] Department of Computer Science, Semantic Technology Institute (STI), University of Innsbruck, Innsbruck, Austria
{sareh.aghaei,kevin.angele,anna.fensel}@sti2.at

[2] Wageningen Data Competence Center and Chair Group Consumption and Healthy Lifestyles, Wageningen University and Research, Wageningen, The Netherlands
anna.fensel@wur.nl

Abstract. Question answering over knowledge graphs targets to leverage facts in knowledge graphs to answer natural language questions. The presence of large number of facts, particularly in huge and well-known knowledge graphs such as DBpedia, makes it difficult to access the knowledge graph for each given question. This paper describes a generic solution based on Personal Page Rank for extracting a small subset from the knowledge graph as a knowledge subgraph which is likely to capture the answer of the question. Given a natural language question, relevant facts are determined by a bi-directed propagation process based on Personal Page Rank. Experiments are conducted over FreeBase, DBPedia and WikiMovie to demonstrate the effectiveness of the approach in terms of recall and size of the extracted knowledge subgraphs.

Keywords: Knowledge graphs · Question answering systems · Knowledge subgraph · Personal Page Rank

1 Introduction

With the growth of the data web, a massive amount of structured data has become available on the web in the form of knowledge graphs (KGs). To assist end users to access KGs, knowledge graph-based question answering systems (KGQASs) have emerged to answer natural language questions [2,5,10]. Although large KGs such as DBPedia with millions or billions of facts are ideal sources for answering questions, accessing these KGs for each given question has become an intricate challenge. To overcome this challenge, the recent KGQASs extract a subset from the KG namely a knowledge subgraph for the question posed over the KG as illustrated in Fig. 1.

A knowledge subgraph targets to prune irrelevant parts of the KG's search space and contains only a set of facts that is likely to capture the answer of a given question. Reducing the search space plays a key role in the efficiency of

T. Di Noia et al. (Eds.): ICWE 2022, LNCS 13362, pp. 237–251, 2022.
https://doi.org/10.1007/978-3-031-09917-5_16

different types of KGQASs including (1) rule-based, (2) information retrieval-based, and (3) semantic parsing-based systems (discussed in Sect. 2). Knowledge subgraphs lead to reducing manual works required for setting up the rule-based systems [1,24,27], pruning candidate entities and reducing training cost in the retrieval-based systems [20,22,23] and making improvements in the mapping stages of semantic-parsing systems due to preventing unnecessary computations [4].

Thus, the task of building knowledge subgraphs over huge KGs avoids exploring the whole KG for each question in KGQASs and narrows down the search space. Basically, a trade-off between answer presence and search space size [9] is required to build knowledge subgraphs. For example, the mean shortest path between entities in DBpedia is around 5-hops, so extracting relevant subgraphs only by navigating a predefined number of hops from a set of entities that represent the question's focus leads to a big part of the DBpedia however covers the answers, as an instance, given a simple question such as "Where is the capital of the US?", there is approximately 600K facts around 1-hop of the US's entity in DBPedia. In contrast, to further reduce the retrieved facts, commonly used techniques [12,14,23] even fail to capture answers of some simple questions that can be addressed through one fact (discussed later).

Therefore, the primary research question of this paper is *how to extract a knowledge subgraph for a posed natural language question that reduces the size of the KG significantly and covers the answer.* For example, given the question sentence "Give me all the companies with more than 1000 employees that were founded in the US from 1986 to 2000" over DBPedia, the extracted knowledge subgraph has to contain relevant facts around the entity of "the US" from millions facts stored in DBPedia which cover the foundation date and employee number of the companies located in the US. Note that the state-of-the-art KGQASs require to learn models for mapping the question to DBPedia facts to find the answer, where the extracted knowledge subgraph helps these systems to tackle with the huge search space size of DBPedia.

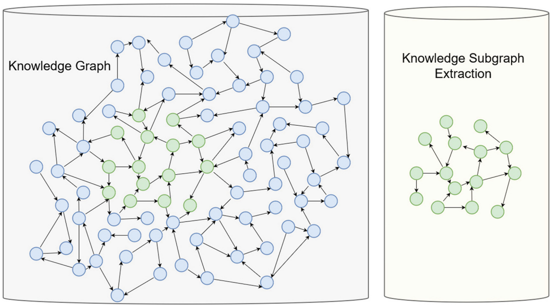

Fig. 1. Extraction a subset from the knowledge graph.

A general architecture to construct a knowledge subgraph for each question to avoid exploring the whole KG is shown in Fig. 2. The architecture consists of three main steps namely topic entity identification, neighborhood retrieval, and knowledge subgraph retrieval. The topic entity identification step employs entity linking (EL) to recognize named entities of questions which reflect the major focus of the questions and next map each entity mentioned in the questions to its corresponding entity in the KG (known as topic entity). Then, the neighbors around topic entities need to be retrieved through n-hop reasoning over the underlying KG. Finally, a knowledge subgraph which includes the topic entity as its first entity, is expanded based on various techniques such as heuristics, neural networks across the retrieved neighbors.

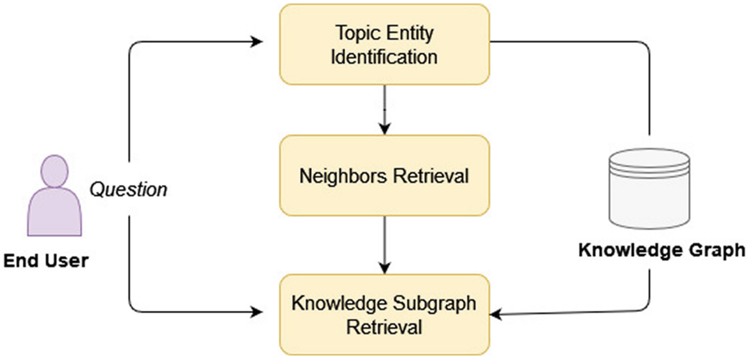

Fig. 2. General architecture of knowledge subgraph construction.

Personal Page Rank (PPR) [13] as a heuristic query-dependent technique is widely used in KGQASs to build a knowledge subgraph around the topic entity with respect to the natural language question posed by the end user [12,14,19,23]. This paper follows the research of [23] in using PPR and proposes a bi-directed propagation technique, called BiDPPR to compute relevance scores for nodes. The BiDPPR employs a bi-directed iterative process in which the scores are propagated through incoming and outgoing edges of nodes in each iteration. The major novelty of the proposed approach lies in detecting when there is no directed path from topic entities to answer entities, the PPR technique fails to build subgraphs covering the answer entities and then proposing a solution to deal with it. For example, given posed questions "Where does Piccadilly start?" and "Where was the author of the theory of relativity educated?" over WikiData and DBPedia, respectively, PPR technique fails to retrieve the knowledge subgraphs which cover the answers because there are no direct paths from topic entities ("Piccadilly" and "theory of relativity") to answer entities ("Dover street" and "ETH Zurich") over the underlying KGs as shown in Fig. 3. Note that although the question "Where does Piccadilly start?" only needs one fact to be answered, the PPR-generated knowledge subgraph does not include the answer.

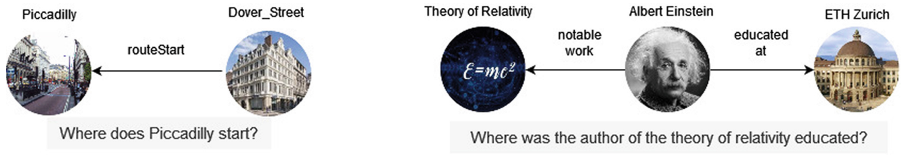

Fig. 3. The path between topic entity and answer entity

The main contributions of the paper can be summarized as follows:

1. An approach to build knowledge subgraphs over KGs for questions is proposed which follows the generic existing architecture shown in Fig. 2.
2. A new bi-directed propagation technique based on PPR is introduced to retrieve those entities from a KG which are more likely to answer questions.
3. Experimental results are demonstrated on QA datasets over FreeBase, DBPedia and WikiMovie and a comparison with available solutions to prove the effectiveness of the proposed approach in terms of recall. Furthermore, the results show how the proposed solution contributes to extracting smaller knowledge subgraphs.

The remind of the paper is organized as follows. Section 2 provides an overview on the related works. The proposed approach is discussed in Sect. 3, and Sect. 4 provides a detailed experimental evaluation including a comparison against state-of-the-art solutions. Finally, Sect. 5 concludes the paper and gives directions for future research.

2 Related Work

The research progress on building knowledge subgraphs in question answering (QA) over KGs can be divided into three categories including filtering-based techniques, heuristic-based techniques, and neural-based techniques.

1. **Filtering-based techniques** rely on predefined rules to filter the number of facts around topic entities. The definition of rules leads to limited scalability and researchers and developers require familiarity with the underlying scheme's KG. Moreover, these techniques are not able to significantly prune irrelevant entities. The introduced Graph Alignment Question Answering (GAQA) approach in [4], defines some query patterns and leverages users' interceptions through an interface to determine the number of required hops to retrieve the paths in the KG. Then, each given question is mapped into a query pattern according to the identified required hops. To prune unnecessary facts while avoid knowledge loss for answering the question, three filtering functions are inserted into the query patterns: (1) filtering out unnecessary predicates (e.g., predicates <http://dbpedia.org/ontology/wikiPageID>, <http://dbpedia.org/ontology/abstract> are assumed as unnecessary predicates in DBPedia KG), (2) filtering out unnecessary literal leaf nodes (e.g., the nodes with irrelevant

language tags have to be eliminated), and (3) filtering out unnecessary resource nodes (e.g., a set of unnecessary namespace URI is defined and resource nodes which belong to this set, are filtered). Finally, a SPARQL query is executed according to the mapped query pattern and the returned result is considered as the knowledge subgraph.

2. **Heuristic-based techniques** use heuristics to build a knowledge subgraph. The PPR [13] as a heuristic algorithm is widely applied in recent KGQASs to retrieve relevant facts around questions [12,14,19,23].
   The PageRank-Nibble (PRN) [3] is an approximate of PPR by applying a tolerance threshold (ϵ) which is used in [23]. Firstly, the topic entity is assumed as query node and all the paths with a maximum length starting at the topic entities are retrieved as a neighborhood graph. Then, the adjacency matrix of the neighborhood graph as a directed graph is generated based on the edge weights. The edge weight is calculated based on the similarity between the edge's surface form[1] and the question. To find the similarity between the question and the edge, GloVe[2] is applied to obtain vector representations and the cosine similarity between two vectors is calculated. Then, the initial PRN score of the topic entity is set to 1 and the other nodes are set to 0. Next, through an iterative process, the PRN score of nodes are computed. In each iteration t, the PRN score is propagated through the outgoing edges of the nodes. After T iterations, the k-top nodes with highest PRN scores (their scores are greater than ϵ) with edges among them are selected as the more relevant facts to the question. The main issue is that PRN fails to retrieve the answer entities once there is no directed path from topic entities to answer entities. The introduced approach in [14] follows the same idea in [23] and expands one hop for CVT[3] (Compound Value Type) entities in Freebase to obtain the extracted knowledge subgraphs (this expansion is applicable if the KG includes CVT nodes).
3. **Neural-based techniques** utilize neural networks to build a subgraph that contains facts relevant to a given question. The Pullnet [22] fulfills an iterative process to construct a subgraph. In each iteration, a graph convolutional network (graph CNN) is used to identify nodes that should be expanded using the pull operation. The pull operation retrieves the top facts from the KG around entity e which are constrained to have e as their subject or object. The retrieved facts are ranked based on the similarity between the fact's relation and the question using a classifier. Thus, the classifier predicts which retrieved facts are more relevant to the question. The major challenge of these techniques is the requirement for question-answer pairs as training data.

Current KGQASs can be classified as (1) rule-based, (2) information retrieval-based and (3) semantic parsing-based systems. In rule-based systems, much manual work is required in the preparation phase due to mappings from

[1] The surface form of an edge is the value of rdfs:label if the edge does not have a label, the variable part of its URI is adopted as the surface form.

[2] https://nlp.stanford.edu/projects/glove/.

[3] https://developers.google.com/freebase/guide/basic_concepts#cvts.

recognized entities to predefined queries or rules. Then, those queries or rules are evaluated over the underlying KG to retrieve the expected answer [1,24,27]. Extracting knowledge subgraphs reduces the manual work required for setting up a rule-based KGQAS. The information retrieval-based systems need to retrieve all candidate answers and then rank them to select the most pertinent answer. So, building a small knowledge subgraph can help pruning the candidate entities and improving the performance of the system [20,22,23]. KGQASs based on semantic parsing basically convert questions to executable queries. In these systems, the unstructured question is mapped to intermediate logical forms and then the intermediate forms are transformed into queries, such as SPARQL. Obviously, reducing the search space on KGs through constructing a pruned knowledge subgraph based on the input question makes improvements in mapping stages of semantic-parsing KGQASs [4].

Although the stream of research on QA over KGs has gained the solutions for building knowledge subgraphs, the recall and size of knowledge subgraphs still need to be improved. For example, filtering-based techniques are not effective in reducing size from a large KG such as DBPedia, PRN fails in building high-recall knowledge subgraphs once there are no directed paths from topic entities to answer entities, and neural-based techniques demand training question-answer pairs which are not available in many practical settings. This paper proposes a bi-directed propagation technique based on PPR (BiDPPR) to build high-recall knowledge subgraphs by considering incoming edges of nodes as well as outgoing edges while the size of the extracted subgraphs not being larger than those constructed by PRN.

3 The Approach

This section presents the proposed approach for constructing high-recall knowledge subgraphs according to the generic architecture shown in Fig. 2 that comprises three main stages including topic entity identification, neighborhood retrievals and knowledge subgraph retrieval.

3.1 Topic Entity Identification

The task of EL is to link an entity mentioned in a text corpus to the corresponding entity in a knowledge base [15]. Here, given a KG containing a set of entities and a set of questions, the goal of EL is to map each entity mentioned in questions to its corresponding entity in the KG [16,21]. The corresponding entities (known as topic entities) generally show the topic of the given question sentences. In this paper, the topic entities of questions are identified through existing EL tools including DBpedia Spotlight and S-MART. The DBpedia Spotlight system [17] automatically annotates questions' sentences with DBpedia URIs, and S-MART is applied for entity linking in FreeBase. This paper assumes that there is at least an entity mentioned in each question (known as topic mention), which shows the main focus of the question and EL identifies its mapping entity in the

KG. As an example, given the question “Give me all the companies with more than 1000 employees that were founded in the US from 1986 to 2000”, the named entity “the US” is the topic mention which is mapped to <http://dbpedia.org/resource/United_States> as the topic entity.

3.2 Neighborhood Retrieval

Once the topic entity of the question is identified, all the entities in the underlying KG which have a distance (distance between two nodes is the number of edges in a shortest undirected path) smaller or equal n are extracted. The extracted entities along with relations among them are defined as neighborhood graph which consists of the n-hop neighbors around the topic entity (according to Definition 1). Generally, according to the number of required hops for reasoning over facts, questions can be grouped into two categories: simple questions and complex questions. A simple question, namely single-hop question, can be answered through only one fact whereas a complex question, called multi-hop question, requires reasoning over two or more facts of the KG [11,19]. Since, in real scenarios, the maximum length of path starting at topic entity do not exceed 3 in general [4], this paper considers $n = 3$.

Definition 1. *A neighborhood graph is defined as $G_N = (N_N, E_N)$ where N_N is a set of entities around the topic entity T_e with distance $d <= n$ from T_e (distance between two nodes is the number of edges in a shortest undirected path), E_N is a set of edges with distance $d < n$ from T_e and n is the depth (the longest undirected path) of the graph.*

To build neighborhood graphs with maximum depth n, SPARQL[4] patterns are defined according to n and T_e. Basically, the total number of possibilities to construct SPARQL patterns around the topic entity T_e with depth n is 2^n. Therefore, 2, 4 and 8 SPARQL patterns can be defined for depths 1, 2 and 3, respectively (the topic entities are shown in blue colour). Figure 4 illustrates all the possible states to construct SPARQL patterns with depth $n <= 3$ and Fig. 5 shows the SPARQL patterns when n is equal to 2.

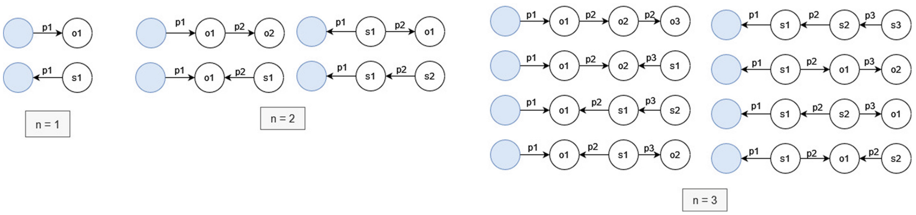

Fig. 4. Possible states to construct SPARQL patterns

[4] https://www.w3.org/TR/rdf-sparql-query/.

```
SELECT <T_e> ?p1 ?o1. ?o1 ?p2 ?o2
            WHERE {
          <T_e> ?p1 ?o1 .
          ?o1 ?p2 ?o2
               }
```

```
SELECT <T_e> ?p1 ?o1. ?s1 ?p2 ?o1
            WHERE {
          <T_e> ?p1 ?o1.
            ?s1 ?p2 ?o1
               }
```

```
SELECT ?s1 ?p1 <T_e>. ?s1 ?p2 ?o1
            WHERE {
          ?s1 ?p1 <T_e>.
            ?s1 ?p2 ?o1
               }
```

```
SELECT ?s1 ?p1 <T_e>. ?s2 ?p2 ?s1
            WHERE {
          ?s1 ?p1 <T_e>.
            ?s2 ?p2 ?s1
               }
```

Fig. 5. SPARQL patterns with depth 2

3.3 Knowledge Subgraph Retrieval

After creating a neighborhood graph for a given input question, a knowledge subgraph is retrieved around the topic entity across the neighborhood graph according to BiDPPR. The formal definition of a knowledge subgraph is provided in Definition 2.

Definition 2. *A knowledge subgraph is a subset of the neighborhood graph which can be defined as $G_K = (N_K, E_K)$ where $N_K \subset N_N$ and $E_K \subset E_N$ and N_K includes the entities which are more likely to be answer entities.*

The proposed technique, BiDPPR, tackles the issue of lacking a directed path from topic entity to answer entity in PRN through a bi-directed propagation process which is summarized as following:

- To consider the impact of incoming edges of a node during the propagation process as well as its outgoing edges, a linear combination of propagation along outgoing edges and incoming edges is utilized to find the BiDPPR score of nodes. If M denotes the adjacency matrix of the neighborhood G_N which presents the edge weights then the transpose of M can be considered as a matrix that includes the inverse relations between entities and let this matrix be M^T. Thus, the calculation of BiDPPR is formulated as:

$$\begin{aligned} pr_v^{(t)} = (1-\alpha)pr_v^{(t-1)} + \alpha\big(\omega_1 \sum_{r} \sum_{<n,r,v>} w_r . pr_n^{(t-1)} + \\ \omega_2 \sum_{r} \sum_{<v,r,n>} w_r^{(t)} . pr_n^{(t-1)}\big) \end{aligned} \tag{1}$$

  Where w_r and w_r^t denote the weights of the edge r in both directions based on the adjacency matrix M and transpose of adjacency matrix M^T, respectively. Also, ω_1 and ω_2 are assumed as coefficient ratio for the incoming edges and outgoing edges, respectively.
- To compute the adjacency matrix, similar to [23], pretrained word embeddings GolVe is applied to generate the embedding of the question and the edges' surface forms. The cosine similarity between the embeddings of the given question and the edge is considered as the weight of that edge.

- To preserve the origin direction of edges, the impact of propagation along outgoing edges ω_1 should be greater than the impact of propagation along incoming edges ω_2.
- In the first initialization, BiDPPR scores are set to $\frac{1}{|N_N|}$ for all non-topic entities and the BiDPPR scores of topic entities are set to $1 + \frac{1}{|N_N|}$ (Eq. 2). Furthermore, the scores are normalized after each iteration to prevent any explosion.

$$pr_v^{(0)} = \begin{cases} \frac{1}{|N_N|} + 1 & topic\,entities \\ \frac{1}{|N_N|} & otherwise \end{cases} \tag{2}$$

Similar to PRN, the k-top nodes by BiDPPR score, along with edges among them are selected to make the knowledge subgraph after T iterations. It is noticeable that the sizes of extracted knowledge subgraphs do not increase in comparison to the extracted subgraphs by PRN. The size of the knowledge subgraph is dependant on K as well as ϵ. In Sect. 4 the coverage of PRN and BiDPPR for different values of K are compared.

Figure 6 illustrates the propagation process in PRN and BiDPPR in a sample neighborhood graph without any directed path from the topic entity A to the answer entity F. As shown in Fig. 6, in the first iteration $t = 1$, the PRN score will be 0 for all nodes except the topic entity A, and the propagation will only happen from node A (the edge are shown in blue colour). For $t = 2$, the PRN score will be non-zero for node A and its neighbors including B, C, D and E, and the propagation will happen from these nodes. For next iterations, the PRN score will be non-zero for the nodes A, B, C, D, E, G and H. Since H and G as dead nodes have no outgoing edges, their scores can not be propagated in the graph. Thus, the PRN score for the node F will stay at 0 by the end due to lack of a directed path from node A to node F. While in BiDPPR, the propagation does not start from a specific node (as the initial scores are not zero) however node A as the topic entity (with the initial score $\frac{1}{8} + 1$ according to 2) significantly impacts on its neighbors. Since the propagation spreads out in both directions in BiDPPR, the score of node F will increase remarkably in the next iteration ($t = 2$) due to happening propagation along incoming edge of node B (note that the weight of the edge between F and B has to be high because its weight is computed based on cosine similarity between the embedding of the question sentence and the edge's label).

4 Experiments

In this section, the proposed approach is evaluated on Freebase, DBpedia and WikiMovies [18] with three QA-benchmarks separately. The code[5] is implemented in python and Stardog[6] is utilized to set up SPARQL endpoints. The PRN technique with $\epsilon = 1e - 6$ is performed.

[5] The GrafNet repository on the Github is reused according to the proposed approach.

[6] https://www.stardog.com/get-started/.

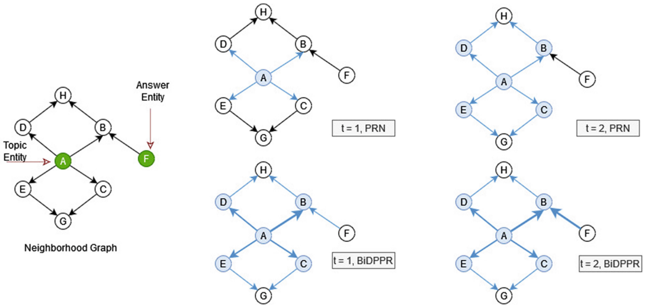

Fig. 6. Propagation process from the topic entity to the answer entity using PRN and BiDPPR

4.1 Knowledge Graphs

Freebase is a practical, scalable KG used to structure general human knowledge [8]. It was launched by Metaweb as an open, public and collaborative KG with schema templates for most kinds of possible entities such as persons, cities, movies, etc. in 2007.

DBPedia is extracted from structured data in Wikipedia through a crowd sourcing community that the main idea behind the extraction is using the key-value pairs in the Wikipedia infoboxes.

WikiMovies is the QA part of the Movie Dialog dataset and supports three different settings of knowledge including (1) using a traditional knowledge base (KB), (2) using Wikipedia as the source of knowledge, and (3) using information extraction over Wikipedia.

4.2 QA Datasets

WebQuestionsSP(WebQSP) dataset [26] includes 4737 natural language questions that were produced by crawling the Google suggest API [7] and are answered through Freebase entities. The questions need up to 2-hop reasoning from the KG. Moreover, the questions are more colloquial and biased towards topics that are frequently asked from Google [6,23].

QALD-6 [25] is the sixth installment of the QALD (Question Answering over Linked Data challenge) and focuses on questions which need up to 3-hop reasoning from the DBPedia. QALD-6 includes 350 training questions and 100 test questions which the test dataset is applied in this experiment.

MetaQA dataset [18] is a large-scale multi-hop dataset in the domain of movies. It includes more than 400k 1-hop, 2-hop and 3-hop questions, containing three individual datasets namely, MetaQA-1hop, MetaQA-2hop and MetaQA-3hop [20].

4.3 Metric

The number of entities in knowledge subgraphs is considered as a metric to compare sizes of knowledge subgraphs. Furthermore, recall as a classical metric to evaluate the effectiveness is adopted for showing the coverage of the constructed knowledge subgraphs. Here, recall is the fraction of the answers that are successfully retrieved by the subgraph as the following:

$$recall = \frac{retrieved\ entities \cap answer\ entities}{answer\ entities} \tag{3}$$

4.4 Results

The experimental results for WebQSP, QLAD-6 and MetaQA datasets with 500 entities ($k = 500$) are shown in Table 1. On WebQSP dataset, the number for recall in PRN is 89.9%, this increased to 92.2% in BiDPPR. The BiDPPR is comparable to the PRN on QLAD-6, the recall improves by 22.1%. On MetaQA dataset, BiDPPR shows the recall improvement around 10% over 3-hop questions. In the case of MetaQA-1hop and MetaQA-2hop, both techniques achieve fully-coverage knowledge subgraphs.

Table 1. Results on WebQSP and MetaQA with 500 entities

Dataset	NPR	BiDPPR
WebQSP	89.9	92.2
QLAD-6	62.7	84.8
MetaQA-1hop	100	100
MetaQA-2hop	100	100
MetaQA-3hop	83.0	92.2

To illustrate that BiDPPR obtains higher recall knowledge subgraphs with fewer number of entities in comparison to NPR, WebQSP is selected as (1) WebQSP includes much more questions in compassion to QALD-6, and (2) Freebase is far larger than WikiMovies. Figure 7 presents the recalls of NPR and BiDPPR on WebQSP over the size of knowledge subgraphs. As the graph shows, the coverage of BiDPPR retrieval knowledge subgraphs is relatively quickly in comparison to NPR. For example, the number of entities to archive the recall 92.0% in NPR is $k = 1200$ while BiDPPR is able to achieve the same recall with $k = 500$.

One point to note is that BiDPPR uses the transpose of the adjacency matrices to consider the inverse direction of the relations. Since the transpose of a matrix can be done in $O(1)$ time (and space), BiDPPR does not affect the time complexity of NPR[7].

[7] Time complexity of NPR is O(m * n) [m = no. of iterations, n= no. of nodes].

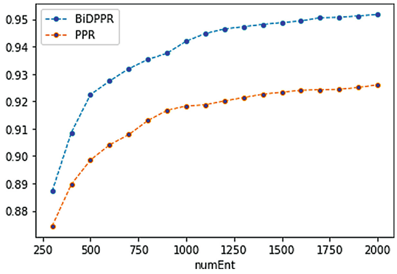

Fig. 7. Recall of BiDPPR and NPR on WebQSP with different number of entities.

4.5 Compared Approaches

BiDPPR is compared with NPR (used in [14,23]) according to Table 1. It is seen that BiDPPR finds higher coverage and smaller knowledge subgraphs for questions.

According to [14], the recall of NPR on WebQSP can increase to 94.9 with 2000 entities once the extracted subgraphs are expanded one hop for CVT entities in Freebase however BiDPPR gains the recall 95.2 with the same number of entities as shown in Table 2.

Table 2. Results on WebQSP with 2000 entities

Technique	NPR	NPR+CVT	BiDPPR
Coverage	92.6	94.9	95.2

The results reported by GAQA in [4] give the answer recall when answer entities are retrieved over the extracted knowledge subgraphs[8] and the coverage of the constructed knowledge subgraphs is not shown in its paper. Basically, GAQA can achieve full-coverage knowledge subgraphs if the query patterns are correctly identified due to it only filters the obvious unnecessary items (e.g., the predicates which are mainly used to link the KGs) however the knowledge graphs are significantly larger than those generated by BiDPPR. Since GAQA's source code is not publicly available, this research study re-implements GAQA's solution. In this re-implementation, 15 questions are randomly selected from each dataset (WebQSP and QLAD-6) and their query patterns are identified based on their SPARQL queries[9]. Figure 8 depicts the average size of the knowledge subgraphs (in terms of the number of entities) for the randomly selected questions and it is clearly shown that BiDPPR builds substantially smaller knowledge subgraphs.

[8] After constructing knowledge subgraphs, GAQA obtains answers of given questions over the extracted subgraphs based a graph-alignment method and then reports the results.

[9] The task of identifying query pattern needs end users' assistance in GAQA.

Furthermore, the recalls of the retrieved subgraphs by BiDPPR are 95.0% and 0.89 for the selected questions in WebSQP and QALD-6, respectively.

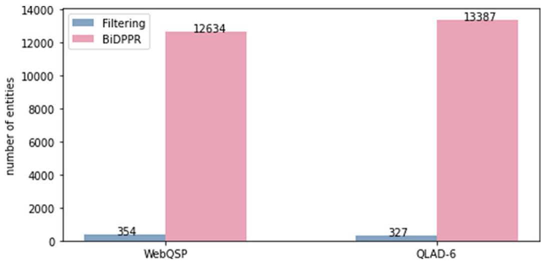

Fig. 8. Comparison of the knowledge subgraphs's size in BiDPPR and GAQA.

Compared to PullNet, BiDPPR needs no training data. PullNet has to train a classifier based on question-answer pairs to predict the relevant facts to questions. The results of PullNet are not directly comparable to the results of this paper due to PullNet's results show the recall after obtaining answers over the knowledge subgraphs and the source code is not available as well. However, according to [22], PullNet is able to retrieve far fewer entities with higher recall in comparison to NPR.

5 Conclusion

With the increasing growth of KGs, QA over KGs can be seen as the most promising approach to make the KGs easily accessible for end users. Since a KG is typically large and stores millions of facts, accessing the KG for each given question in KGQASs is difficult or even impossible. Extracting a small subset from the KG (known as knowledge subgraph) that is likely to contain the answer entity, defiantly reduce the search space and make the final answer extraction process easier. This paper proposes an approach including three major stages: topic entity identification, neighborhood retrieval and knowledge subgraph retrieval. The main focus of the approach is to introduce a new derivation of the PPR technique called BiDPPR to construct the knowledge subgraphs. Once there is no directed path from topic entities to answer entities, PPR technique fails to construct knowledge subgraphs which contain the answer entities. To address this problem, BiDPPR suggests propagating along the incoming edges as well as the outgoing edges. The proposed approach finds higher recall knowledge subgraphs with fewer entities than the ones created before. The effectiveness of the proposed approach in terms of recall and size is illustrated on WebQuestionsSP, QLAD-6 and MetaQA datasets which apply Freebase, DBPedia and WikiMovie as KGs to answer questions, respectively.

In the future, given a natural language question, a syntactic-semantic representation is created as question graph and the number of hops to retrieve the neighborhood graph is calculated based on the longest path in the question graph. Then, the task of QA over KGs is reduced to finding subgraph matches of the question graph over the knowledge subgraph.

Acknowledgement. This research has been supported by the project WordLiftNG within the Eureka, Eurostars Programme (grant agreement number 877857 with the Austrian Research Promotion Agency (FFG)) and the project KI-NET within the Interreg Osterreich-Bayern 2014–2020 programme (grant agreement number AB 292).

References

1. Abujabal, A., Yahya, M., Riedewald, M., Weikum, G.: Automated template generation for question answering over knowledge graphs. In: Proceedings of the 26th International Conference on World Wide Web, pp. 1191–1200 (2017)
2. Ait-Mlouk, A., Jiang, L.: Kbot: a knowledge graph based chatbot for natural language understanding over linked data. IEEE Access **8**, 149220–149230 (2020)
3. Andersen, R., Chung, F., Lang, K.: Using pagerank to locally partition a graph. Internet Math. **4**(1), 35–64 (2007)
4. Bakhshi, M., Nematbakhsh, M., Mohsenzadeh, M., Rahmani, A.M.: Data-driven construction of sparql queries by approximate question graph alignment in question answering over knowledge graphs. Exp. Syst. Appl. **146**, 113205 (2020)
5. Bao, J., Duan, N., Zhou, M., Zhao, T.: Knowledge-based question answering as machine translation. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 967–976 (2014)
6. Bast, H., Haussmann, E.: More accurate question answering on freebase. In: Proceedings of the 24th ACM International on Conference on Information and Knowledge Management, pp. 1431–1440 (2015)
7. Berant, J., Chou, A., Frostig, R., Liang, P.: Semantic parsing on freebase from question-answer pairs. In: Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, pp. 1533–1544 (2013)
8. Bollacker, K., Evans, C., Paritosh, P., Sturge, T., Taylor, J.: Freebase: a collaboratively created graph database for structuring human knowledge. In: Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data, pp. 1247–1250 (2008)
9. Christmann, P., Roy, R.S., Weikum, G.: Beyond ned: fast and effective search space reduction for complex question answering over knowledge bases. arXiv preprint arXiv:2108.08597 (2021)
10. Fensel, A., Toma, I., García, J.M., Stavrakantonakis, I., Fensel, D.: Enabling customers engagement and collaboration for small and medium-sized enterprises in ubiquitous multi-channel ecosystems. Comput. Ind. **65**(5), 891–904 (2014)
11. Fu, B., Qiu, Y., Tang, C., Li, Y., Yu, H., Sun, J.: A survey on complex question answering over knowledge base: recent advances and challenges. arXiv preprint arXiv:2007.13069 (2020)
12. Fu, K., et al.: Ts-extractor: large graph exploration via subgraph extraction based on topological and semantic information. J. Vis. **24**(1), 173–190 (2021)
13. Haveliwala, T.H.: Topic-sensitive pagerank. In: World Wide Web, pp. 517–526 (2002)

14. He, G., Lan, Y., Jiang, J., Zhao, W.X., Wen, J.R.: Improving multi-hop knowledge base question answering by learning intermediate supervision signals. In: Proceedings of the 14th ACM International Conference on Web Search and Data Mining, pp. 553–561 (2021)
15. Li, J., Sun, A., Han, J., Li, C.: A survey on deep learning for named entity recognition. IEEE Trans. Knowl. Data Eng. **34**, 50–70 (2020)
16. Ling, X., Singh, S., Weld, D.S.: Design challenges for entity linking. Trans. Assoc. Comput. Linguist. **3**, 315–328 (2015)
17. Mendes, P.N., Jakob, M., García-Silva, A., Bizer, C.: Dbpedia spotlight: shedding light on the web of documents. In: Proceedings of the 7th International Conference on Semantic Systems, pp. 1–8 (2011)
18. Miller, A., Fisch, A., Dodge, J., Karimi, A.H., Bordes, A., Weston, J.: Key-value memory networks for directly reading documents. arXiv preprint arXiv:1606.03126 (2016)
19. Qiu, Y., Wang, Y., Jin, X., Zhang, K.: Stepwise reasoning for multi-relation question answering over knowledge graph with weak supervision. In: Proceedings of the 13th International Conference on Web Search and Data Mining, pp. 474–482 (2020)
20. Saxena, A., Tripathi, A., Talukdar, P.: Improving multi-hop question answering over knowledge graphs using knowledge base embeddings. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 4498–4507 (2020)
21. Shen, W., Wang, J., Han, J.: Entity linking with a knowledge base: issues, techniques, and solutions. IEEE Trans. Knowl. Data Eng. **27**(2), 443–460 (2014)
22. Sun, H., Bedrax-Weiss, T., Cohen, W.W.: Pullnet: open domain question answering with iterative retrieval on knowledge bases and text. arXiv preprint arXiv:1904.09537 (2019)
23. Sun, H., Dhingra, B., Zaheer, M., Mazaitis, K., Salakhutdinov, R., Cohen, W.W.: Open domain question answering using early fusion of knowledge bases and text. arXiv preprint arXiv:1809.00782 (2018)
24. Unger, C., Bühmann, L., Lehmann, J., Ngonga Ngomo, A.C., Gerber, D., Cimiano, P.: Template-based question answering over rdf data. In: Proceedings of the 21st International Conference on World Wide Web, pp. 639–648 (2012)
25. Unger, C., Ngomo, A.-C.N., Cabrio, E.: 6th open challenge on question answering over linked data (QALD-6). In: Sack, H., Dietze, S., Tordai, A., Lange, C. (eds.) SemWebEval 2016. CCIS, vol. 641, pp. 171–177. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46565-4_13
26. Yih, W., Richardson, M., Meek, C., Chang, M.W., Suh, J.: The value of semantic parse labeling for knowledge base question answering. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pp. 201–206 (2016)
27. Zheng, W., Yu, J.X., Zou, L., Cheng, H.: Question answering over knowledge graphs: question understanding via template decomposition. Proc. VLDB Endow. **11**(11), 1373–1386 (2018)