A Joint Model for Question Answering over Multiple Knowledge Bases

Yuanzhe Zhang, Shizhu He, Kang Liu, Jun Zhao
National Laboratory of Pattern Recognition (NLPR)
Institute of Automation, Chinese Academy of Sciences, Beijing, 100190, China
{yzzhang, shizhu.he, kliu, jzhao}@nlpr.ia.ac.cn

Abstract

As the amount of knowledge bases (KBs) grows rapidly, the problem of question answering (QA) over multiple KBs has drawn more attention. The most significant distinction between multiple KB-QA and single KB-QA is that the former must consider the alignments between KBs. The pipeline strategy first constructs the alignments independently, and then uses the obtained alignments to construct queries. However, alignment construction is not a trivial task, and the introduced noises would be passed on to query construction. By contrast, we notice that alignment construction and query construction are interactive steps, and jointly considering them would be beneficial. To this end, we present a novel joint model based on integer linear programming (ILP), uniting these two procedures into a uniform framework. The experimental results demonstrate that the proposed approach outperforms state-of-the-art systems, and is able to improve the performance of both alignment construction and query construction.

Introduction

With the continued growth of knowledge bases (KBs) on the web, how to access such precious intellectual resources becomes increasingly important (Unger, Freitas, and Cimiano 2014). Knowledge base based question answering (KB-QA) just focuses on this problem and is able to use natural language as query language. Therefore, it has received more attention in recent years.

The key problem in KB-QA is to convert natural language questions into structured queries, such as SPARQL queries. There are many researches that focus on this problem, and most of them are single KB-QA (Frank et al. 2007; Zettlemoyer and Collins 2005; 2007; 2009; Kwiatkowski et al. 2011; 2013). They often assume that the answers could be acquired from a single KB. However, it is almost unpractical that using a single KB could cover all questions. A plenty of KBs exist on the web and they could focus on different domains. It is not rare that a natural language question involves many aspects, and each aspect is covered by a relevant KB. Such question would be answered by using multiple KBs. We name this task as multiple KB-QA, which is seldom investigated before, except for (Lopez et al. 2012; Shekarpour et al. 2014; Fader, Zettlemoyer, and Etzioni 2014).

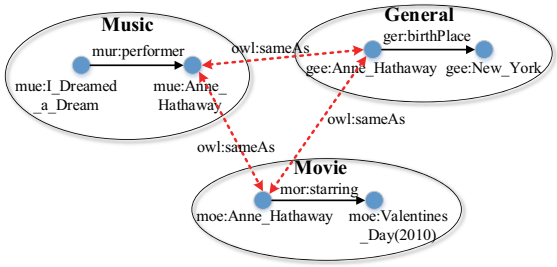

Figure 1: Three KBs should be used to answer the question "Which songs are performed by person who was born in New York and played a role in Valentine's Day?".

This is a challenging task. For example, consider the following question:

Which songs are performed by person who was born in New York and played a role in Valentine's Day?[1]

As illustrated in Figure 1, the answer to "songs performed by" is in a music domain KB, and the answer to "born in New York" is in a general domain KB, and answering "played a role in Valentine's Day" should turn to a movie domain KB. The final structured query is generated by uniting different fragments as follows:

SELECT ?v1 WHERE {
⟨?v1, mur:perfomer, ?v2⟩[2]
⟨?v2, owl:sameAs, ?v3⟩
⟨?v3, mor:starring, moe:Valentines_Day(2010)⟩
⟨?v3, owl:sameAs, ?v4⟩
⟨?v4, ger:birthPlace, gee:New_York⟩ }

From this example, we can see that the most significant difference between multiple KB-QA and single KB-QA is that the former needs to consider the interconnections

[1]This is a real case in Chinese QA scenario, and there is no such a Chinese KB could answer it alone.

[2]This triple pattern means that ?v2 is the performer of ?v1. The first two letters of the prefix represent the source KB (mo: movie, mu: music and ge: general), and the last letter represents the type (e: entity, c: class and r: relation). E.g., **mur** means the resource is from **mu**sic KB and is a **r**elation.

between heterogeneous KBs, such as ⟨?v2, owl:sameAs, ?v3⟩ and ⟨?v3, owl:sameAs, ?v4⟩. This kind of links (owl:sameAs) are also called *alignments*. Only by constructing such links, the triple patterns from different KBs could be integrated and generate the final answer.

Unfortunately, such alignments are usually not explicitly presented. Therefore, for multiple KB-QA, it is natural to adopt a pipeline strategy including two steps: alignment construction and query construction. First, the alignments between heterogeneous KBs are identified. Then, multiple KBs could be linked together and be regarded as a large single KB. Thus, existing single KB-QA methods could be applied. However, such pipeline strategy has two limitations.

First, constructing these alignments is not a trivial task (Choi, Song, and Han 2006; Euzenat, Shvaiko, and others 2007; Ngo, Bellahsene, and Todorov 2013). The reported F-measure of large-scale alignment construction is approximately 80% (Dragisic et al. 2014). As a result, the alignments obtained by automatic methods are inevitably noisy. Moreover, such noises would be passed on to the subsequent step and have negative effects on the final answer generation.

Second, existing KBs usually grow fast and update frequently, for example, DBpedia (Bizer et al. 2009) increased by 110 million facts in 2014. The constructed alignments are prone to be out of date and new alignments need to be added. However, it is not necessary to identify all alignments between KBs for a question. In most cases, only identifying the question related alignments is sufficient, whereas pipeline methods are unable to capture the question related information.

Therefore, we wonder whether performing alignment construction and query construction jointly can alleviate these problems. We notice that alignment construction and query construction could influence each other. On the one hand, the identified alignments obviously have impacts on query construction. For example, if we have a correct alignment ⟨moe: Anne_Hathaway, owl:sameAs, gee:Anne_Hathaway⟩, and it links triple pattern ⟨?v1, mor:starring, moe:Valentines_Day(2010)⟩ and ⟨?v1, ger:birthPlace, gee:New_York⟩, we are strongly implied that the two triple patterns should be selected to construct a query. On the other hand, the query construction part can help identify alignments. For example, when we build alignments independently, the entity moe:Anne_Hathaway in the movie KB may be matched to two entities both named "Anne Hathaway" in the general KB. One of them was born in New York and the other was born in Shottery. If we know that ⟨?v1, ger:birthPlace, gee:New_York⟩ is selected for query construction, the one born in New York is implied to be matched.

Based on the above considerations, we propose a joint method by encoding alignment construction and query construction into a unified model. In specific, we employ an integer linear programming (ILP) model, where the candidate triple patterns and the potential alignments are the variables restricted by several designed constraints, and they could be determined simultaneously through global joint inference. In this way, alignment construction and query construction could affect each other. To the best of our knowledge, this is the first work to jointly consider alignment construction and query construction.

The experimental results demonstrate that the proposed approach outperforms state-of-the-art systems, and is able to improve both the performance of alignment construction and query construction compared with the pipeline system.

The contributions of this paper are as follows:

1) We propose a novel approach that jointly considers alignment construction and query construction, treating them as interactive procedures for the first time.
2) The proposed approach achieves better performance compared with state-of-the-art multiple KB-QA systems.
3) The proposed joint model improves the performance of both alignment construction and query construction compared with the pipeline model.

Our Method

Given multiple KBs that usually represented as subject-property-object (*SPO*) triples, our objective is to translate a natural language question into a formal language query. To achieve this goal, we need to translate the input question into several candidate triple patterns, where each triple pattern corresponds to a query triple in the formal query. Then, different from single KB-QA, we need to construct alignments (owl:sameAs links) between the variables of different triple patterns from multiple KBs. The alignments indicate the ways of linking two triple patterns. In specific, we design four ways: `1↔1`, `1↔2`, `2↔1` and `2↔2`, where `i↔j` denotes that the *i-th* variable of the left-side triple pattern is linked to the *j-th* variable of the right-side triple pattern by owl:sameAs links. Figure 2 illustrates how a question can be solved by the triple patterns and the alignments.

In general, our approach has five steps as follows. The first and second step aim at extracting candidate triple patterns from the given questions. The third step aims at constructing potential alignments between variables from different triple patterns. The fourth step performs global joint inference. The final step constructs structured queries according to the inference results and finds the correct answers from KBs.

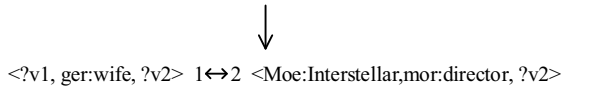

Figure 2: How questions are solved by triple patterns and the ways they are linked.

The first step: phrase detection & resource mapping

In this step, we aim to detect phrases in the question, and map them into the resources in KBs. In specific, we first exploit the labels of all resources in the employed KBs to build a resource dictionary. Next, for all word sequences contained

Resource	Triple Pattern	Type
Relation (Rel)	⟨?v1, Rel, ?v2⟩	R
Class (Cla)	⟨?v1, rdf:type, Cla⟩	ER
Entity (Ent) + Relation (Rel)	⟨?v1, Rel, Ent⟩ / ⟨Ent, Rel, ?v2⟩	ER

Table 1: Candidate triple pattern generation method, where ?v1 and ?v2 are variables.

in the question, we consult the dictionary. If the similarity based on Levenshtein distance sim between the label of a resource and the word sequence is greater than a certain threshold θ[3], we output this word sequence as the detected phrase and select this resource as the corresponding candidate resource. Meanwhile, we set sim as the confidence value of the resource, and the occurrence frequency of the resource is also recorded. Note that a phrase can be mapped to more than one resources from different KBs. Disambiguation is not performed in this step and will be performed in the joint inference step.

The second step: candidate triple pattern generation

The objective of this step is to generate candidate triple patterns from the acquired candidate resources. In contrast to linguistic methods (which use linguistic tools to parse the structrue of an input question), we adopt a KB-based manner to organize the candidate resources. This generating method could not only reduce errors brought by linguistic tools, but also generate more candidate triple patterns, preserving more possibilities. Specifically, we exploit the KB facts to find out the possible triple patterns, and each triple pattern is a basic query unit in the final query, as indicated in Table 1. If a candidate resource is a relation, it can generate a triple pattern of type *R* (Relation). If a candidate resource is a class, it will generate a triple pattern of type *ER* (Entity & Relation). If an entity and a relation appear in the same triple in a KB, they can be combined to form an *ER* type triple pattern. Note that the resources in a triple pattern are from the same KB. Although there may be conflicts between the triple patterns, such as containing the same resources, we resolve them in the global joint inference step.

The third step: potential alignment generation

In this step, our goal is to generate potential alignments based on the acquired candidate triple patterns. Different triple patterns are linked by variables, e.g., ⟨?v1, mor:starring, moe:Valentines_Day(2010)⟩ is a triple pattern from the movie KB, and ⟨?v1, ger:birthPlace, gee:New_York⟩ originates from the general KB. Each triple pattern has only one variable, so if a link exists between them, it will be a $1 \leftrightarrow 1$ link. Now, we should find out that whether there are alignments between these two triple patterns. Concretely, we investigate whether the entities represented by these two variables can be matched. We first turn to the KB facts to figure out the entities represented by the variables. Next, we adopt a string-based matching algorithm using Levenshtein distance similarity to acquire the alignments. In specific, if two entities from different KBs are similar enough (similarity is greater than 0.65), they are identified as an alignment. The alignment confidence is the matching similarity. This is a common baseline matching method in the area of ontology matching. Unlike most simple ontology matching tasks, e.g., OAEI benchmarks (Euzenat et al. 2011), we allow that one entity matches multiple entities in other KBs, because we intend to recognize all potential alignments, whereas 1-1 matching sometimes omits correct alignments.

No alignments exist between those triple patterns from the same KB. To link these triple patterns, we only need to find out the shared entities represented by the variables. For example, ⟨?v1, ger:playForTeam, ?v2⟩ and ⟨?v1, ger:birthPalce, gee:New_York⟩ both come from the general KB. The shared entities of both the variables (e.g., gee:Kareem_Abdul_Jabbar, gee:Alex_Arias, etc.) indicate that the two patterns can be linked by them ($1 \leftrightarrow 1$). We also call these shared entities alignments for convenience, and their confidence is set to 1.0.

The fourth step: global joint inference

This step is the key of our approach. By now, we already have all the candidate triple patterns and the potential alignments, and we can combine them together to get a disambiguation graph, as presented in Figure 3. To obtain the final formal query, we need to know: i) which triple patterns should be selected and ii) how the selected triple patterns are linked, i.e., which alignments should be selected. Meanwhile, the selected triple patterns and alignments should be consistent. To this end, we design an integer linear programming (ILP) based model, collectively considering the mutual influence between these two parts, and design several constraints to restrict them. A global joint inference will be performed to determine the triple patterns and alignments simultaneously. The details will be provided in the following section.

The final step: formal query generation

This step aims to generate the formal query from the global joint inference results. In our example, the selected triple patterns are as follows.

t1: ⟨?v1, mur:performer, ?v2⟩
t2: ⟨?v1, mor:starring, moe:Valentines_Day(2010)⟩
t3: ⟨?v1, ger:birthPlace, gee:New_York⟩

The selected alignments are ⟨mue:Anne_Hatha- way, owl:sameAs, moe:Anne_Hathaway⟩ that links t1 and t2 ($2 \leftrightarrow 1$), and ⟨moe:Anne_Hathaway, owl:sameAs, gee:Anne_Hathaway⟩ that links t2 and t3 ($1 \leftrightarrow 1$). We use SELECT ?v WHERE to generate SPARQL query, where ?v is the variable most close to the question word, and this turns out to be an effective assumption. So we can obtain the formal query as follows.

SELECT ?v1 WHERE {
⟨?v1, mur:perfomer, ?v2⟩

[3] θ is assigned an empirical value (0.84) to ensure that i) sufficient candidate resources are acquired and ii) improbable candidate resources are excluded to reduce the candidate space.

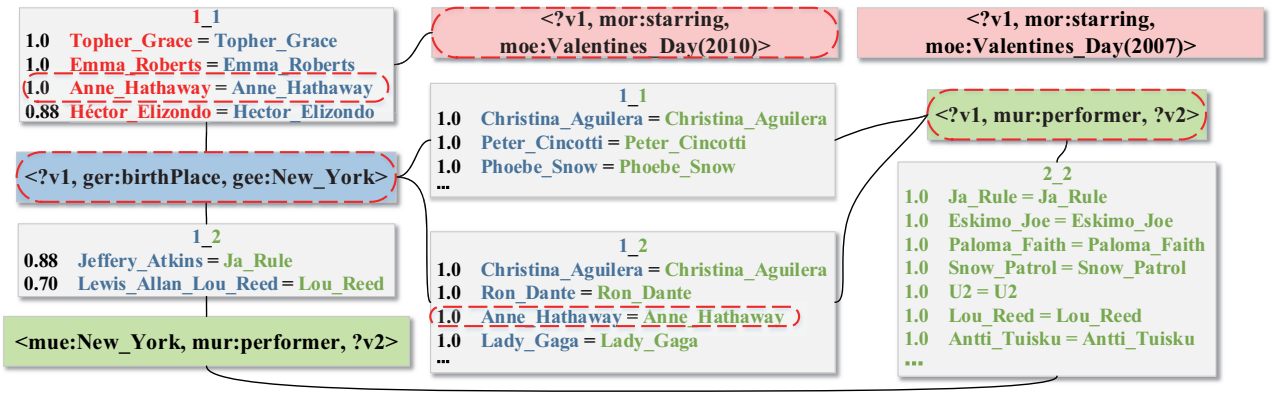

Figure 3: The disambiguation graph of the running example. The colorful boxes are the candidate triple patterns, and the items in the gray boxes are the potential alignments (prefixes are omitted for clarity, and "=" means owl:sameAs). The finally selected triple patterns and alignments are indicated by the dotted boxes.

⟨?v2, owl:sameAs, ?v3⟩
⟨?v3, mor:starring, moe:Valentines_Day(2010)⟩
⟨?v3, owl:sameAs, ?v4⟩
⟨?v4, ger:birthPlace, gee:New_York⟩ }

The alignment sets that the selected alignments belonging to are used together with the KBs to answer the SPARQL query. Concreatly, as shown in Figure 3, the left upper and middle lower alignment sets are used. Note that although these alignments are provided, only those that could derive an answer are considered as the constructed alignments, and are evaluated in our experiments.

Global Joint Inference

Objective Function

We use binary variable $t_i \in \{0, 1\}$ to denote the i-th triple pattern. 1 indicates that the triple pattern is selected and 0 means the triple pattern is not selected. w_i^t denotes the importance of t_i. Here, we measure the importance by the average confidence and frequency of their contained resources.

$a_j \in \{0, 1\}$ is used to represent the j-th alignment. 1 indicates that the alignment is selected and 0 means the alignment is not selected. w_j^a denotes the confidence of a_j.

Intuitively, more important triple patterns and more reliable alignments should be selected. We also consider another factor, namely, the selected triple patterns should cover as many as words in the input question. We use c_k to denote the number of words that t_k covers.

According to the above analysis, the objective function can be given as follows.

$$\text{maximize } \alpha \sum_{i=1}^{N_t} w_i^t t_i + \beta \sum_{j=1}^{N_a} w_j^a a_j + \gamma \sum_{k=1}^{N_t} c_k t_k$$

Here α, β and γ are the parameters of the ILP. N_t is the total number of the candidate triple patterns, and N_a is the total number of the potential alignments.

Constraints

We define a series of constraints on the binary variables in the ILP. They are divided into two categories as follows.

Constraints on triple patterns

Constraint 1 Conflicts may occur between triple patterns, if they contain resources from the same or overlapping phrases. For example, in triple pattern ⟨?v1, mor:starring, moe:Valentines_Day(2010)⟩ and ⟨?v1, mor:starring, moe:Valentines_Day(2007)⟩, moe:Valentines_Day(2010) and moe:Valentines_Day (2007) are from the same phrase "Valentine's Day", which only has one exact meaning. Thus, these two triple patterns should not be selected simultaneously. The original phrases of a triple pattern are explicit, so it is easy to put the overlapping triple patterns into a set o. At most one triple pattern in this set can be selected. Many sets of this type may exist, represented by $\mathbb{O} = \{o_1, o_2, ..., o_n\}$. This type of constraints can be formulated as follows.

$$\sum_{t_i \in o} t_i \leq 1, \ \forall o \in \mathbb{O} \tag{1}$$

Constraint 2 At least one triple pattern typed *ER* should be selected. If all the selected triple patterns are of type *R*, no legal formal query can be formed. We use $Type(t_i)$ to represent the type of t_i. The constraint can be given as follows.

$$\sum_{Type(t_i)=ER} t_i \geq 1 \tag{2}$$

Constraints on alignments

Constraint 3 If an alignment a_j is selected, then the two triple patterns it links should also be selected. We use $t_{left(a_j)}$ to denote the left-side triple pattern that a_j links, and $t_{right(a_j)}$ represents the right-side triple pattern. The constraints of this type are given as follows.

$$a_j \leq t_{left(a_j)} \ \ and \ \ a_j \leq t_{right(a_j)} \tag{3}$$

Constraint 4 Because two triple patterns can ultimately only be linked in one way, we can only select at most one alignment between two triple patterns. Thus, we include all the alignments that link two certain triple patterns in a set l. Many sets of this type may exist, represented by $\mathbb{L} = \{l_1, l_2, ..., l_n\}$. The constraints are formulated as follows.

$$\sum_{a_j \in l} a_j \leq 1, \ \forall l \in \mathbb{L} \tag{4}$$

Constraint 5 Conflicts may occur between alignments when they link the same triple pattern. Thus, the consistency of the selected alignments should be taken into account. The following two illustrative examples explain two types of conflict.

Example 1. Consider the following triple patterns.

t1: ⟨?v1, mor:starring, moe:Valentines_Day(2010)⟩

t2: ⟨?v1, ger:birthPlace, gee:New_York⟩

t3: ⟨?v1, mur:performer, ?v2⟩

Assume we select alignment ⟨moe:Anne_Hath- away, owl:sameAs, gee:Anne_Hathaway⟩ to link t1 and t2 (1↔1), and select ⟨gee:Lady_Gaga, owl:sameAs, mue:Lady_Gaga⟩ to link t2 and t3 (1↔2). In this case, two alignments both link t2, and ?v1 in t2 concurrently represents Anne Hathaway and Lady Gaga, which is not allowed.

Example 2. Consider the following triple patterns.

t4: ⟨?v1, mor:starring, moe:Valentines_Day(2010)⟩

t5: ⟨?v1, mur:performer, ?v2⟩

t6: ⟨?v1, ger:birthPlace, ?v2⟩

Assume triple pattern t4 is linked to t5 with alignment ⟨moe:Anne_Hathaway, owl:sameAs, mue:Anne_Hathaway⟩ (1↔2). Meanwhile, t5 is linked to t6 with alignment ⟨mue:New_York, owl:sameAs, gee:New_York⟩ (1↔2). This situation will lead to a non-existent triple ⟨mue:New_York, mur:performer, mue:Anne_Hathaway⟩, which is not allowed.

To model this type of constraints, we also include such conflicting alignments in a set c to avoid their simultaneous selection. Many sets of this type may exist, represented by $\mathbb{C} = \{c_1, c_2, ..., c_n\}$. The constraints are formulated as follows.

$$\sum_{a_j \in c} a_j \leq 1, \ \forall c \in \mathbb{C} \tag{5}$$

Thus far, the ILP problem has been well defined. The global joint inference is performed by solving the optimization problem. We use *Gurobi*[4] to implement ILP. Although the problems are usually complicated, they still can be solved in a few seconds. The most time consuming part of the proposed method is generating the disambiguation graph, which could be improved by distributed computing. The ILP solving part is quite efficiency. The average speed is 2.2 second/question in Chinese dataset.

[4]http://www.gurobi.com/

Experiments

Datasets

We perform our experiments on two open datasets to make fair comparisons, namely, the benchmark over interlinked KBs (Shekarpour et al. 2014), and the Question Answering over Linked Data 4 (QALD-4) TASK2 dataset (Unger et al. 2014). Note that these two datasets have already given the perfect alignments between the KBs. To verify the effectiveness of our proposed approach, we remove these alignments when using our model.

We also provide a novel Chinese dataset[5]. To increase the diversity of the questions, we asked five questioners to pose questions independently. The question set was constructed based on three Chinese KBs. The movie KB is extract from *Mtime*[6], and the music KB comes from *Douban Music*[7], and the general KB is extracted from *Baidu Baike*[8].

Settings

Our experiments are conducted on a standard computer with 2.7GHz quad-core CPU and 4GB memory. The proposed joint model has three parameters: α, β and γ. The benchmark over interlinked KBs does not contain training set, so we set the parameters to 1.0 by default. For the QALD-4 TASK2 dataset and the Chinese dataset, we use the provided training set to tune the parameters.

Results & Discussion

Comparisons with state-of-the-art To demonstrate the overall effectiveness of the proposed method, we compare the performance of our system with that of state-of-the-art systems on the three datasets mentioned above. Our system is compared with SINA (Shekarpour et al. 2014) in all the three datasets. The performance of SINA in benchmark dataset is reported in their original research, and we re-implemented it in the other two datasets to acquired the performance. For the QALD-4 TASK2 dataset, we compare our system with the participants in the evaluation: POMELO (Hamon et al. 2014) and RO_FII (Built by The Faculty of Computer Science at Alexandru Ioan Cuza University of Iasi).

Datasets	Systems	Prec.	Rec.	F_1
Benchmark	SINA	0.95	0.90	0.92
	Ours	**0.96**	**0.96**	**0.96**
QALD-4 TASK2	POMELO	0.82	0.87	0.85
	RO_FII	0.16	0.16	0.16
	SINA	0.80	0.78	0.79
	Ours	**0.89**	**0.88**	**0.88**
Chinese	SINA	0.64	0.63	0.63
	Ours	**0.77**	**0.78**	**0.77**

Table 2: Comparisons with state-of-the-art systems.

Table 2 presents our results. From the results, we can observe that for benchmark, *Ours* could improve F-measure by

[5]https://goo.gl/ziMZ8w

[6]http://www.mtime.com/

[7]http://music.douban.com/

[8]http://baike.baidu.com/

4%. For QALD-4 TASK2 dataset, *Ours* improves F-measure by 3%. Note that there is another participant system GFMed (Marginean 2014) in QALD-4 TASK2, and it can achieve a high F-measure of 0.99. We do not compare with it because GFMed follows a controlled language approach, which is built on Grammatical Framework grammar, and is designed manually for this specific task. For Chinese dataset, *Ours* improves F-measure by 14% compared with SINA. The results demonstrate that our method outperforms the state-of-the-art systems.

Joint vs. Pipeline To further verify the effectiveness of the proposed joint model, we design a pipeline system for comparison. The pipeline model is constructed as follows. First, the alignments are determined independently of query construction, and only the alignments with the highest confidence are reserved as correct ones. Then, we remove the alignment relevant item in the objective function ($\beta \sum_{j=1}^{N_a} w_j^a a_j$), and remove the constraints relevant to alignment construction (constraint 4 and 5). The query construction procedure is performed using the remained ILP model.

Table 3 presents the results, where **Pipe** represents the pipeline method that performs alignment construction and query construction independently and **Joint** represents our proposed joint model. The performance of query construction (**QC**) is measured by the results of the final QA. The performance of alignment construction (**AC**) is judged manually. Because no gold standard has been defined, the recall cannot be evaluated, and we only evaluate precision.

Datasets (Method)	**QC**			**AC**
	Prec.	Rec.	F_1	Prec.
benchmark (Pipe)	0.76	0.76	0.76	0.80
benchmark (Joint)	**0.96**	**0.96**	**0.96**	**0.92**
QALD-4 TASK2 (Pipe)	0.65	0.64	0.64	0.72
QALD-4 TASK2 (Joint)	**0.89**	**0.88**	**0.88**	**0.92**
Chinese (Pipe)	0.72	0.72	0.72	0.84
Chinese (Joint)	**0.77**	**0.78**	**0.77**	**0.94**

Table 3: Comparisons between joint and pipeline methods on three datasets.

We can make the following observations based on the results: i) The performance of query construction and alignment construction is related. If query construction performs well, the alignments will also be of high quality, and vice versa. This result indicates that the two procedures have mutual influence. ii) For all three datasets, the joint model achieves superior results in both alignment construction and query construction. We believe the reason is that the two procedures are not independent, and thus, the joint model is beneficial for both of them.

Impacts of the constraints To determine the impacts of the constraints, we remove one constraint at a time in joint inference. Then, we evaluate the performance of alignment construction and query construction.

Figure 4 presents the results. We can observe the following: i) removing any constraint will lead to a decrease in performance in both alignment construction and query construction, and ii) constraint 3 has the greatest influence on performance. This finding indicates that the interactions between triple patterns and alignments are of vital importance. The results demonstrate that the constraints bridging triple patterns and alignments are more significant than other constraints, which further proves that joint inference is useful.

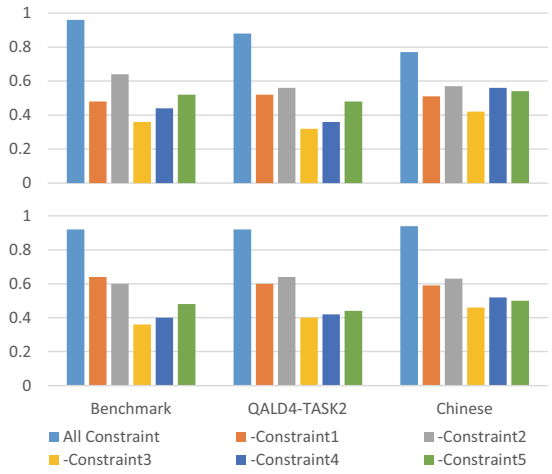

Figure 4: The impacts of constraints. Upper: F-measure of query construction. Lower: Precision of alignment construction.

Related Work

Several QA systems have been developed to address the multiple KB-QA problem.

SINA (Shekarpour et al. 2014) is a system that draws attention to answering questions over interlinked KBs. Interlinked means that the interconnections (mostly owl:sameAs links) between the KBs are already given. However, such perfect alignments are rare in reality. SINA employs a hidden Markov model (HMM) to determine the resources. Then, the resources are organized into incomplete query graphs (IQGs) according to several heuristic rules. Finally, the IQGs are combined using the given interconnections.

Fader et al. (2014) propose an open-domain KB-QA system (OQA) that also involves multiple KBs. However, OQA does not focus on answering questions that need align multiple KBs. OQA puts the employed four KBs together, generating a single large KB, and the KBs are not actually integrated. The facts from different KBs are simply accumulated, and OQA uses an approach similar to information retrieval to query the KB.

PowerAqua (Lopez et al. 2012) focuses on answering questions across multiple ontologies, without any existing alignments. It first uses a linguistic tool to parse the input natural language question. Using the syntactic structure of the question, triple queries are generated. Then, the answers of each triple query are acquired by searching the ontologies. Finally, the answers are merged (the merging step includes aligning resources in different ontologies) and ranked. Our

approach surpasses PowerAqua in two aspects: i) We consider the interaction between alignment construction and query construction. By contrast, PowerAqua generates triple queries independently, neglecting the alignments construction part. ii) Our approach does not depend on external linguistic tools, which may introduce errors. Moreover, PowerAqua also adopts pipeline structure, and would pass on these errors.

Yahya et al. (2013) also utilize ILP to implement KB-QA; Schwarte et al. (2011) provide join processing and grouping techniques for federated query processing on linked data, but the input is SPARQL query instead of natural language query.

Conclusions

In this paper, we present a joint method to address the QA problem over multiple KBs. The novelty of the proposed method lies in jointly considering two interacting procedures in multiple KB-QA, i.e., alignment construction and query construction, and encoding them into an ILP model. We conduct several experiments on three datasets to verify the effectiveness of the proposed approach. The results demonstrate that the proposed joint model is able to improve the performance of both alignment construction and query construction. The comparisons with state-of-the-art systems further demonstrate the superiority of the proposed approach.

Acknowledgements

The work was supported by the Natural Science Foundation of China (No. 61533018), the National High Technology Development 863 Program of China (No. 2015AA015405) and the National Natural Science Foundation of China (No. 61272332). And this research work was also sponsored by CCF-Tencent Open Research Fund. We thank the anonymous reviewers for their insightful comments.

References

Bizer, C.; Lehmann, J.; Kobilarov, G.; Auer, S.; Becker, C.; Cyganiak, R.; and Hellmann, S. 2009. Dbpedia-a crystallization point for the web of data. *Web Semantics: science, services and agents on the world wide web* 7(3):154–165.

Choi, N.; Song, I.-Y.; and Han, H. 2006. A survey on ontology mapping. *ACM Sigmod Record* 35(3):34–41.

Dragisic, Z.; Eckert, K.; Euzenat, J.; Faria12, D.; Ferrara, A.; Granada, R.; Ivanova, V.; Jiménez-Ruiz, E.; Kempf, A. O.; Lambrix, P.; et al. 2014. Results of the ontology alignment evaluation initiative 2014. In *9th ISWC workshop on ontology matching (OM)*, 61–104.

Euzenat, J.; Meilicke, C.; Stuckenschmidt, H.; Shvaiko, P.; and Trojahn, C. 2011. Ontology alignment evaluation initiative: Six years of experience. In *Journal on data semantics XV*. Springer. 158–192.

Euzenat, J.; Shvaiko, P.; et al. 2007. *Ontology matching*, volume 18. Springer.

Fader, A.; Zettlemoyer, L.; and Etzioni, O. 2014. Open question answering over curated and extracted knowledge bases. In *Proceedings of SIGKDD*, 1156–1165. ACM.

Frank, A.; Krieger, H.-U.; Xu, F.; Uszkoreit, H.; Crysmann, B.; Jörg, B.; and Schäfer, U. 2007. Question answering from structured knowledge sources. *Journal of Applied Logic* 5(1):20–48.

Hamon, T.; Grabar, N.; Mougin, F.; and Thiessard, F. 2014. Description of the pomelo system for the task 2 of qald-2014. In *CLEF 2014 Working Notes Papers*.

Kwiatkowski, T.; Zettlemoyer, L.; Goldwater, S.; and Steedman, M. 2011. Lexical generalization in ccg grammar induction for semantic parsing. In *Proceedings of EMNLP*, 1512–1523.

Kwiatkowski, T.; Choi, E.; Artzi, Y.; and Zettlemoyer, L. 2013. Scaling semantic parsers with on-the-fly ontology matching. In *Proceedings of EMNLP*, 1545–1556.

Lopez, V.; Fernández, M.; Motta, E.; and Stieler, N. 2012. Poweraqua: Supporting users in querying and exploring the semantic web. *Semantic Web* 3(3):249–265.

Marginean, A. 2014. Gfmed: Question answering over biomedical linked data with grammatical framework. In *CLEF 2014 Working Notes Papers*.

Ngo, D.; Bellahsene, Z.; and Todorov, K. 2013. Opening the black box of ontology matching. In *The Semantic Web: Semantics and Big Data*. Springer. 16–30.

Schwarte, A.; Haase, P.; Hose, K.; Schenkel, R.; and Schmidt, M. 2011. Fedx: Optimization techniques for federated query processing on linked data. In *The Semantic Web–ISWC 2011*. Springer. 601–616.

Shekarpour, S.; Marx, E.; Ngomo, A.-C. N.; and Auer, S. 2014. Sina: Semantic interpretation of user queries for question answering on interlinked data. *Web Semantics: Science, Services and Agents on the World Wide Web* 39–51.

Unger, C.; Forascu, C.; Lopez, V.; Ngomo, A.-C. N.; Cabrio, E.; Cimiano, P.; and Walter, S. 2014. Question answering over linked data (qald-4). In *Working Notes for CLEF 2014 Conference*.

Unger, C.; Freitas, A.; and Cimiano, P. 2014. An introduction to question answering over linked data. In *Reasoning Web. Reasoning on the Web in the Big Data Era*. 100–140.

Yahya, M.; Berberich, K.; Elbassuoni, S.; and Weikum, G. 2013. Robust question answering over the web of linked data. In *Proceedings of CIKM*, 1107–1116.

Zettlemoyer, L. S., and Collins, M. 2005. Learning to map sentences to logical form: Structured classification with probabilistic categorial grammars. In *Proceedings of UAI*, 658–666.

Zettlemoyer, L. S., and Collins, M. 2007. Online learning of relaxed ccg grammars for parsing to logical form. In *In Proceedings EMNLP-CoNLL*, 678–787.

Zettlemoyer, L. S., and Collins, M. 2009. Learning context-dependent mappings from sentences to logical form. In *Proceedings of ACL-IJCNLP*, 976–984.