Mapping Factoid Adjective Constraints to Existential Restrictions over Knowledge Bases

Jiwei Ding, Wei Hu(✉), Qixin Xu, and Yuzhong Qu(✉)

State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, China
jwdingnju@outlook.com, qxxunju@outlook.com, {whu,yzqu}@nju.edu.cn

Abstract. The rapid progress of question answering (QA) systems over knowledge bases (KBs) enables end users to acquire knowledge with natural language questions. While mapping proper nouns and relational phrases to semantic constructs in KBs has been extensively studied, little attention has been devoted to adjectives, most of which play the role of factoid constraints on the modified nouns. In this paper, we study the problem of finding appropriate representations for adjectives over KBs. We propose a novel approach, called Adj2ER, to automatically map an adjective to several existential restrictions or their negation forms. Specifically, we leverage statistic measures for generating candidate existential restrictions and supervised learning for filtering the candidates, which largely reduce the search space and overcome the lexical gap. We create two question sets with adjectives from QALD and Yahoo! Answers, and conduct experiments over DBpedia. Our experimental results show that Adj2ER can generate high-quality mappings for most adjectives and significantly outperform several alternative approaches. Furthermore, current QA systems can gain a promising improvement when integrating our adjective mapping approach.

Keywords: Factoid adjective constraint · Question answering · KBQA

1 Introduction

With the rapid development of question answering (QA) systems over structured data [1,4,8,11], end users are able to query knowledge bases (KBs) with natural language questions, a.k.a. KBQA. Semantic parsing [5,24] is a key technique widely-used in these systems, which transforms natural language questions to structural queries like SPARQL queries. Entity linking [7,9] and relation mapping [16] are two major steps in most of the semantic parsing approaches, which map proper nouns and relational phrases to entities and relations (or relation chains) in given KBs, respectively. However, current approaches pay little attention to understanding adjectives over KBs. An example question with adjectives and its corresponding SPARQL query are shown in Table 1.

C. Ghidini et al. (Eds.): ISWC 2019, LNCS 11778, pp. 164–181, 2019.
https://doi.org/10.1007/978-3-030-30793-6_10

Table 1. An example for understanding adjectives by using a KB

Natural language question	List all American actors who are alive.
Existential restriction or its negation	$\exists$ *dbo:nationality*.{*dbr:United_States*} $\neg\exists$ *dbo:deathDate*.$\top$
SPARQL query over DBpedia	select distinct ?s where { ?s *rdf:type* *dbo:Actor* . ?s *dbo:nationality* *dbr:United_States* . filter not exists { ?s *dbo:deathDate* ?o } }

In linguistics, an adjective is a descriptive word, the main syntactic role of which is to qualify a noun or noun phrase, giving more information about the object signified. Except for the adjectives which appear in proper nouns or "how + adjective" questions, the majority of adjectives in questions takes the meanings of factoid constraints. In KBs, such adjectives can be captured by the existence of certain properties or facts. Inspired by this, we consider understanding adjectives by mapping them to existential restrictions or their negation forms in description logics [2], where concepts, roles and individuals used in the existential restrictions come from classes, properties and entities of a given KB, respectively. An example for understanding adjectives in question by using DBpedia is shown in the second row of Table 1. We map "American" to an existential restriction $\exists$*dbo:nationality*.{*dbr:United_States*}, giving the meaning that an "American actor" may have a fact about his *nationality* with value *dbr:United_States*. Similarly, "alive" is mapped to an existential restriction in negation form $\neg\exists$ *dbo:deathDate*.$\top$, meaning that "actors who are alive" should not have the facts about their death dates in DBpedia. Compared with current QA systems [1,26] which map adjectives to specific classes or entities, mapping adjectives to existential restrictions can cover a higher portion of adjectives and capture the meanings of natural language questions more precisely. We believe that, in addition to entity linking and relation mapping, adjective mapping should be another important step in semantic parsing.

According to our observations, there are two main challenges in generating appropriate mappings for adjectives. Firstly, the lexical gap between the input adjectives and vocabulary elements used in the target existential restrictions may be huge. For example, for adjective "alive", the appropriate mapping $\neg\exists$ *dbo:deathDate*.$\top$ cannot be found by a similarity-based searching approach (which is commonly used in mapping proper nouns or relational phrases), since the similarity between "alive" and "death date" is not obvious. Although some neural network based approaches [4,12] enhance the ability of semantic similarity calculation, they suffer from the lack of training data on this task. Secondly, the search space of this task is quite large, since many facts in the KB may express the meaning of an input adjective. Some adjectives require the representations in negation forms, making the search space even larger.

In order to cope with the above challenges, we propose a new adjective mapping approach, called Adj2ER, based on the following observation: entities that embody the meaning of an input adjective should have a different fact distribution compared to entities that do not. For example, most dead actors have a fact about their death date, while actors who are alive do not. Thus, the facts about *dbo:deathDate* are considered as discriminative for "alive", and $\neg\exists$ *dbo:deathDate*.$\top$ can be generated as a candidate existential restriction. Generating existential restrictions from such discriminative facts can reduce the search space and overcome the lexical gap at the same time. In our approach, the set of entities that embodies the meaning of an input adjective is collected by retrieving Wikipedia, and candidate existential restrictions are generated from related facts by using statistic measures, followed by a supervised filtering step to improve the accuracy. We created two question sets with adjectives from QALD and Yahoo! Answers, and conducted experiments over DBpedia. Our experimental results turn out to be promising.

The rest of this paper is structured as follows. In Sect. 2, we define the adjective mapping problem. Our approach to solving the adjective mapping problem is proposed in Sect. 3. In Sect. 4, we report the experimental results on adjective mapping and QA tasks. In Sect. 5, we discuss several findings in our experiments. Related work is presented in Sect. 6. Finally, Sect. 7 concludes this paper with future work.

2 Problem Definition

To see the usage of adjectives in natural language questions, we investigated the adjective occurrences in the 5^{th} challenge on question answering over linked data (QALD-5) [21] dataset. Among all the 349 non-hybrid questions, 117 adjective occurrences are contained in 107 questions. We classified the 117 adjective occurrences into four categories (see Table 2).

In this paper, we mainly focus on the adjectives which take the meanings of factoid constraints (33.3%). For example, "American actor" means a subclass of "Actor" whose nationality is United States. In KBs, these constraints can be described as the existence of certain facts, so the primary target of our work is to map adjectives to existential restrictions in description logics. As for the adjectives used as the names of entities/relations, they should not be

Table 2. Classification of adjectives appeared in QALD-5

Categories	Percentage	Examples
Factoid constraint	33.3%	Give me all Swedish holidays
Name of entity/relation	32.5%	Himalayan mountain system
How + adjective	27.4%	How many companies [...]
Structural constraint	6.8%	Which other weapons [...]

Table 3. Forms of existential restrictions and examples

Logic form	Examples	
$\exists r.\top$	(Married people):	$dbo{:}Person \sqcap \exists dbo{:}spouse.\top$
$\exists r.\{a\}$	(Chinese cities):	$dbo{:}City \sqcap \exists dbo{:}country.\{dbr{:}China\}$
$\neg\exists r.\top$	(People who are alive):	$dbo{:}Person \sqcap \neg\exists dbo{:}deathDate.\top$
$\neg\exists r.\{a\}$	(Hot food):	$dbo{:}Food \sqcap \neg\exists dbo{:}servingTemperature.\{\text{"}Cold\text{"}\}$

interpreted alone and have already been considered in the entity linking and relation mapping tasks. Also, "how + adjectives" questions can be interpreted using template-based or rule-based parsing approaches [8,20]. For the remaining 6.8% of adjectives such as "same" and "other", they do not express the meanings of certain facts, but may influence the structures of target query graphs. We will consider these structural adjective constraints in the future.

The work in [10] showed that the meanings of adjectives vary when modifying nouns from different classes, e.g., "American actors" means actors who have nationality United States, while "American cities" means cities that are located in the United States. In this sense, we consider the *class* for the noun that an adjective modifies as an important factor for the adjective mapping problem. Since many studies [1,12] have been done to map natural language phrases to classes in KBs, we skip the class mapping part and mainly focus on the problem of mapping adjectives to existential restrictions. We define the adjective mapping problem as follow:

Definition 1 (Adjective Mapping Problem). *Given an adjective adj and a class C (which stands for the class of the modified noun) in a KB, the adjective mapping problem is to map (adj, C) to existential restrictions or negation forms in description logics, such as $\exists r.\top$, $\exists r.\{a\}$, $\neg\exists r.\top$ or $\neg\exists r.\{a\}$, where r, a are a specific role and an individual in the KB, respectively. The resulted restrictions should reflect the meaning of the given adjective on the class.*

Due to the diversity of knowledge representations, there may exist more than one candidate mapping, e.g., both $\exists$*dbo:nationality.*{*dbr:United_States*} and $\exists$*dbp:nationality.*{*"American"*} are appropriate for "American" on *dbo:Actor*. Our study aims to find all suitable mappings for an input adjective.

To simplify the adjective mapping problem, we only consider the existential restrictions (or their negation forms) that can be determined by the existence of one certain fact. The considered forms of existential restrictions are shown in Table 3, which cover 92.3% of the factoid adjective constraints in QALD-5. For the adjectives that can be mapped to more complex structures, such as $\exists r_1.\exists r_2.\top$, we discuss them in Sect. 5.

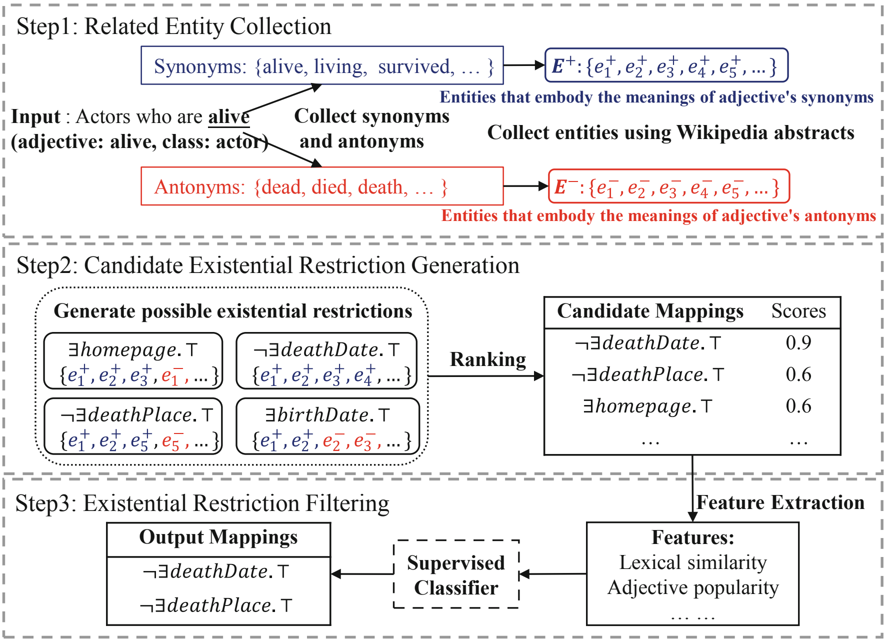

Fig. 1. Framework of the proposed approach

3 The Proposed Approach

In this section, we propose an approach, Adj2ER, to automatically map adjectives to several existential restrictions or their negation forms. The framework of the approach is shown in Fig. 1, which contains the following three steps:

1. **Related entity collection.** Two sets of entities (denoted by E^+ and E^-) are automatically collected by retrieving an adjective (or its synonyms and antonyms) in Web corpora such as Wikipedia abstracts, where E^+ denotes entities that embody the meanings of the input adjective's synonyms, and E^- denotes entities that embody the meanings of the adjective's antonyms.
2. **Candidate existential restriction generation.** Several candidate existential restrictions are generated from the facts about these entities in a given KB. Each candidate should cover most entities in E^+, and its negation should cover most entities in E^-.
3. **Existential restriction filtering.** A supervised learning method is designed to refine the candidate existential restrictions. The remaining restrictions are returned as the output of our approach.

Details for each step are described in the following three subsections.

3.1 Related Entity Collection

In this step, entity sets E^+ and E^- can be automatically collected by retrieving the co-occurrence of each entity and the adjective in Web corpora. In our approach, we choose Wikipedia abstracts, due to their high quality and good coverage, and they can be directly linked to entities in KBs through relations like *dbo:wikiPageID*.

For each entity e_i of class C in KB, we consider e_i as an element of E^+ if the input adjective appears in a sentence of its Wikipedia abstract. The following constraints are employed to ensure the accuracy:

- If the input adjective co-occurs with a negative word such as "never" or "not" in a sentence of e_i's Wikipedia abstract, this sentence is ignored.
- Sentences which begin with other entities may not describe e_i directly, thus they should not be considered during retrieving. These sentences can be detected and filtered out using page link information.
- Due to the incompleteness of KB, some unpopular entities may not have proper facts to embody the meaning of the adjective. Considering this, entities with less than 10 facts are not collected.

For some adjectives, e.g., "alive" and "dead", they rarely appear in Wikipedia abstracts, we automatically generate some alternative words for retrieving by using the following lexicons:

PPDB [17] is an automatically-extracted paraphrase database, which provides some equivalence and entailment relations between natural language words and phrases. For example, it provides both "died" and "death" for adjective "dead". We consider all the words that have a high-confidence equivalence or entailment relation[1] with the input adjective as alternative words.

WordNet [14] is a lexical database of English, where nouns, verbs, adjectives and adverbs are grouped into synsets, each expressing a different meaning. Each adjective participates in several synsets in WordNet, and it can be considered as an alternative word if it shares the same synset with the input adjective. However, some synsets contain rarely-used word senses of the input adjective, e.g., "dead" and "stagnant" share the same synset "not circulating or flowing", which is a rarely-used word sense for "dead". It may lower the precision if we consider "stagnant" as an alternative word for "dead". In this sense, we only consider top-2 synsets of each adjective when generating alternatives.

The method for collecting entities in E^- is very similar to E^+ after we fetch the antonyms of the input adjective in WordNet. For adjectives without any antonyms, we randomly sample some entities in class C which are not covered by E^+ to build E^-. The entities appearing in both E^+ and E^- should be removed from both sets, since it is hard to determine whether they embody the meaning of the adjective or not. Finally, a uniform random sampling method is used to make E^+ and E^- approximately the same in size.

[1] We used the S size PPDB downloaded from http://paraphrase.org/#/download.

3.2 Candidate Existential Restriction Generation

In this step, our approach generates some candidate existential restrictions using a statistical learning method. As shown in the middle of Fig. 1, our approach firstly generates all possible existential restrictions (or existential restrictions in negation forms) from the facts that are related to entities in E^+ and E^-, and then ranks them by a combined measurement based on the supporting degrees of E^+ and E^-.

For each fact about entity $e_i \in E^+$ in KB, this fact may describe the embodiment of the input adjective. Considering this, our approach generates two possible existential restrictions, $\exists r.\top$ and $\exists r.\{a\}$, for each fact about e_i in form of $(e_i, a) : r$. As for the facts in form of $(a, e_i) : r$, the approach also generates two possible existential restrictions, $\exists \bar{r}.\top$ and $\exists \bar{r}.\{a\}$, where $\bar{r}$ denotes the inverse of relation r. This step repeats for all $e_i \in E^+$. Particularly, we regard type as a role to make our approach unified, e.g., the fact "$Al\_Pacino : Actor$" is regarded as "$(Al\_Pacino, Actor) : type$".

If the input adjective has an antonym, each fact about entity $e_j \in E^-$ in KB may describe the embodiment of the antonym. Two possible existential restrictions in negation forms, $\neg\exists r.\top$ and $\neg\exists r.\{a\}$, are generated for each fact about e_j in form of $(e_j, a) : r$, which indicates that the entities embodying the meaning of the input adjective should not have such a fact in KB. Similarly, $\neg\exists \bar{r}.\top$ and $\neg\exists \bar{r}.\{a\}$ are generated for each fact about e_j in form of $(a, e_j) : r$.

After generating all the possible existential restrictions, our approach ranks them by a combined measurement. Let R_i be an existential restriction or a negation of existential restriction, we define the supporting degree on entity set E (E can be E^+ or E^-) as:

$$Sup(R_i, E) = \begin{cases} \dfrac{|\{e_i | e_i \in E \wedge e_i : R_i\}|}{|E|}, & \text{if } R_i \text{ contains no negation} \\ 1 - \dfrac{|\{e_i | e_i \in E \wedge e_i : \neg R_i\}|}{|E|}, & \text{otherwise} \end{cases} . \quad (1)$$

To calculate the supporting degrees, we also consider the facts inferred from sub-class and sub-property axioms, which are conducted in advance. Thus, the procedure of calculating the supporting degrees for R_i only needs to judge whether there is a certain fact for each $e_i \in E$. Since each fact is only related to two restrictions, it only needs to go through all the facts about entities in E^+ and E^- once the supporting degrees for all existential restrictions are calculated. Let F denote all the facts about entities in E^+ and E^-, the time complexity of calculating the supporting degrees is $O(2 \times |F|) = O(|F|)$, due to there are at most $2 \times |F|$ restrictions.

We adopt some measurements frequently used in information retrieval to rank all possible existential restrictions. First, a candidate existential restriction R_i should be supported by most of entities in E^+, while its negation $\neg R_i$ should be supported by most of entities in E^-. This assumption meets the goal of

Table 4. Features used in filtering for restriction R_i

Categories	#	Descriptions
Statistic	1, 2	Supporting degrees of R_i on E^+ and E^-, calculated by Eq. (1)
	3	Accuracy of R_i on E^+ and E^-, calculated by Eq. (2)
	4	Precision of R_i on E^+ and E^-, calculated by Eq. (3)
	5	Combined score of R_i on E^+ and E^-, calculated by Eq. (4)
Adjective popularity	6	Popularity of adjective in class C
	7	Popularity of adjective's antonym in class C (= 0 if no antonym)
Similarity	8	Lexical similarity between adjective (or its antonym) and R_i
	9	Semantic similarity between adjective (or its antonym) and R_i
Restriction form	10	Indicator: whether R_i contains negation
	11	Indicator: whether R_i contains an individual
	12	Indicator: whether R_i uses a reverse relation as role
	13	Indicator: whether R_i uses relation *type* as role

accuracy, which can be calculated as follow:

$$Acc(R_i, E^+, E^-) = \frac{Sup(R_i, E^+) + Sup(\neg R_i, E^-)}{2}. \tag{2}$$

Second, for any entity e_i in E^+ or E^-, if $e_i : R_i$ holds, the probability for $e_i \in E^+$ should be high. This assumption meets the goal of *precision*, which can be calculated as follow:

$$Prec(R_i, E^+, E^-) = \frac{Sup(R_i, E^+)}{Sup(R_i, E^+) + Sup(R_i, E^-)}. \tag{3}$$

The overall score for R_i is a combination of the above two measurements:

$$Score(R_i, E^+, E^-) = Acc(R_i, E^+, E^-) \times Prec(R_i, E^+, E^-). \tag{4}$$

We consider the top-M existential restrictions with *Score* larger than β as candidates. The whole time complexity is $O(|F|) + O(k \times 2 \times |F|) = O(|F|)$.

3.3 Existential Restriction Filtering

Although the existential restriction generation step generates several meaningful mappings, it only considers measurements on the supporting degrees of E^+ and E^-, which means that the precision of the candidates may highly associate with the quality of these two entity sets. Also, the meaning of the adjective itself is not considered. For example, we observed that our candidates contain $\exists$*foaf:homepage*.$\top$ for adjective "alive" on class *dbo:Person*, which means "people who are alive should have a homepage". This mapping is not precise but reasonable, since it captures a distribution characteristic for the facts on E^+ and E^-, which implies that most living people have a homepage (because people in the

KB are usually famous, and famous people usually have homepages), while most dead people do not (because there is even no computer when they were alive).

In this step, we design a supervised learning method to filter out inaccurate candidate mappings. We use a linear kernel SVM classifier, and consider four types of features for each existential restriction R_i, as shown in Table 4. *Statistic* features contain measurements considered in the candidate generation, such as the supporting degrees, accuracy and precision for R_i on E^+ and E^-. *Adjective popularity* features capture the frequency of the input adjective (or its antonym) in Wikipedia abstracts for entity with type C. Adjectives with low popularity may not have a proper mapping in a given KB. *Similarity* features consider the similarity between the adjective and the vocabulary elements used in R_i:

$$Similarity(adj, R_i) = \max_{w \in W(adj), l \in L(R_i)} Similarity(w, l), \tag{5}$$

where $W(adj)$ denotes all synonyms and antonyms of *adj* collected in the first step, and $L(R_i)$ denotes all labels for the role and individual appeared in R_i. We use the Levenshtein distance to calculate lexical similarity, and a pre-trained word embedding [18] to calculate semantic similarity (cosine similarity of word vectors). *Restriction form* features indicate which form of existential restriction R_i belongs to, which is defined in Table 3. Additionally, we add a feature to indicate whether R_i uses relation *type* as role, since some KBs usually use a subclass to capture the meaning of an adjective-modified class.

We manually label some existential restrictions for each input as training data. In the training procedure, we firstly execute Steps 1 and 2 to generate some candidates, and then calculate features mentioned above. To balance the number of positive and negative examples, we treat all candidates that appear in the answer set as positive examples, and select a part of remaining candidates according to the descending order of *Score* as negative examples. During testing, all the existential restrictions that are labeled positive are treated as the output.

4 Experiments

We evaluated Adj2ER over DBpedia, with adjectives used in questions over structural data and community QA questions. Our experiments want to verify two hypotheses: (i) our approach can generate accurate mappings for most of adjectives in natural language questions, and (ii) current QA systems can benefit from integrating our adjective mapping approach. The question sets, source code, and experimental results are available at http://ws.nju.edu.cn/Adj2ER/.

4.1 Experiments on Adjective Mapping

Question Sets. Before building the question sets for evaluation, we implemented a simple method to identify questions with factoid adjective constraints. We first found all the questions that contain adjectives using the Stanford CoreNLP POS tagging module [13], and filtered out the questions in which adjectives only appear after “how”. Also, we removed the questions in which adjectives only appear in proper nouns or relational phrases using EARL [9], a joint entity and relation linking system. Finally, we removed the questions with adjectives which represent structural constraints, such as “same”, “other” and “different”, using a manually-collected word list. This approach achieved an overall accuracy of 87% when we built the question sets for the following experiments, and the errors are mainly caused by relational phrase recognition failure.

QALDadj65 contains 65 (adj, C) pairs appearing in the questions from QALD-1 to QALD-9 [22]. We first extracted all the questions that contain factoid adjective constraints, and leveraged a type linking method [12] based on word embedding similarity to find a class in DBpedia for the noun that each adjective modifies. In some cases, the adjective modifies an entity, we took a minimal type of the entity as its class. Finally, 65 distinct (adj, C) pairs were generated. Furthermore, five Semantic Web majored graduate students were asked to build existential restrictions for each input. An existential restriction was considered as a reference answer only if more than three assessors had mentioned it in their answer sets. The agreement score between the assessors is good (Fleiss’ $\kappa = 0.76$), and the average size of reference answers for each input is 3.72.

YAadj396 contains 396 frequently-used (adj, C) pairs in Yahoo! Answers Comprehensive Questions[2]. We firstly extracted the top-$1,000$ frequently-used (adj, C) pairs in questions. However, 412 of them contain adjectives like “good” and “favourite”, whose meanings are related to personal preferences. Also, the assessors failed to achieve consistency answers for another 192 (adj, C) pairs, such as “new book” and “small city”, and the meanings of most of these adjectives are not captured by DBpedia. For the remaining 396 input pairs, the average size of reference answers is 2.93. Considering the difficulty of this task, the agreement score between different assessors is acceptable (Fleiss’ $\kappa = 0.62$) .

Metrics and Settings. For each input, we adopted precision (P), recall (R) and F1-score (F1) as the metrics. Especially, when the approach provided no answer for the input, we set P = R = F1 = 0. For each question set, we reported the average P, R and F1 values for all inputs. We performed 5-fold cross-validation on each question set. The following settings of Ajd2ER were evaluated:

Adj2ER-w/o filtering. We selected the top-K candidates according to the descending order of $Score$ in Eq. (4), where K is a hyperparameter.

Adj2ER-full. We set parameter M to 20, and threshold β to 0.1 in the candidate generation step.

Comparative Approaches. We compared Adj2ER with the following two alternative approaches, which are commonly used in current QA systems:

Linking-based. Some existing QA systems [1,8,26] directly link adjectives or “adjective + noun” phrases to entities, classes or literals in the given KB. For

[2] https://webscope.sandbox.yahoo.com/.

Table 5. Evaluation results for adjective mapping

	QALDadj65			YAadj396			Time
	P	R	F1	P	R	F1	
Linking-based	31.90%	43.88%	32.66%	40.40%	34.18%	33.49%	**2.53s**
Network-based	40.26%	43.92%	36.48%	40.50%	40.54%	37.27%	89.15s
Adj2ER-w/o filtering	52.30%	36.89%	38.36%	39.98%	39.73%	36.54%	7.12s
Adj2ER-full	**71.30%**	**58.44%**	**59.65%**	**56.79%**	**46.29%**	**47.97%**	8.41s

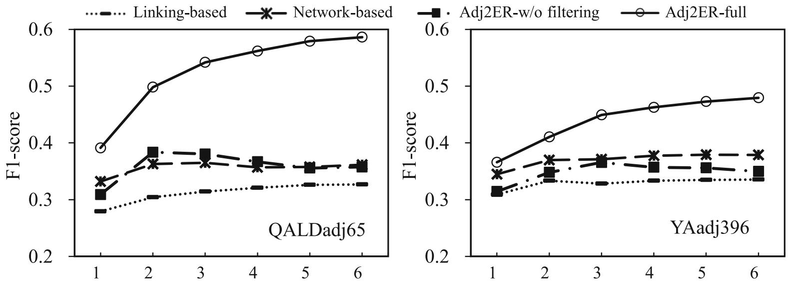

Fig. 2. F1-scores for top-K results of different approaches

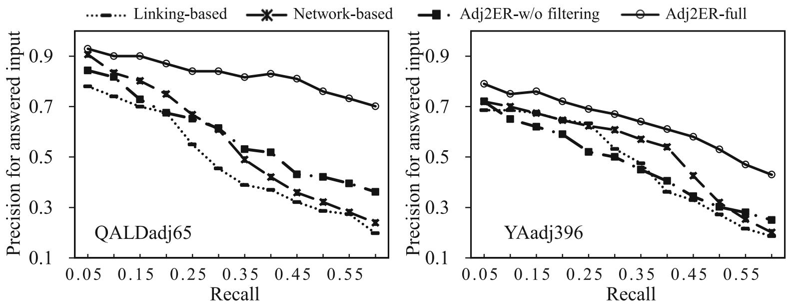

Fig. 3. Precision-recall curves for different approaches

each input (adj, C) pair, the linking-based approach links adj to some entities and classes using EARL [9], as well as some literals using Lucene search. Let a be the linking result from the above process. An existential restriction $\exists r.\{a\}$ is considered as a candidate if there is at least one entity $e : C$ which holds $(e, a) : r$ in KB. Then, a similarity score is calculated using cosine similarity between the averaged word embedding of each existential restriction and the "adjective + class" phrase. Finally, this approach considers the top-K existential restrictions satisfying similarity score larger than θ as the final output.

Network-based. Some recent QA systems [4,12] utilize neural networks to learn the semantic similarity between a natural language constraint expression and a target query component. For each input (adj, C) pair, the network-based approach randomly samples at most 1,000 entities of class C, and generates a large amount of candidate existential restrictions from all the facts about these entities. Then, all the candidates are ranked according to the semantic similarity with the input "adjective + noun" phrase, calculated by an encode-and-compare model [4]. The model consists of two convolutional neural networks which map the phrase and the existential restriction to 200-dimensional vectors, and the semantic similarity is calculated by the cosine similarity of the vectors. Finally, the top-K existential restrictions with similarity score larger than θ were considered as output.

Results. Table 5 shows the results for each adjective mapping approach, and Fig. 2 shows the F1-scores for top-K answers. The results on both question sets are quite similar, and our Adj2ER-full approach achieved the best performance. Compared with Adj2ER-w/o filtering, Adj2ER-full gained an improvement of more than 15% in precision, which indicated that the supervised filtering step successfully filtered out some inaccurate mappings. The average time cost by Adj2ER-full was 8.41 seconds, and approximately 80% of time was spent on getting data from the local DBpedia endpoint. We considered the time cost acceptable since we can prepare a lexicon for all adjectives before we integrate this approach into QA systems.

The results of the linking-based approach are not good (F1-scores are lower than 35% on both question sets), which indicated that the lexical gap between the adjectives and the target existential restrictions is huge. The network-based approach achieved similar results compared with Adj2ER-w/o filtering. However, it required labeled training data to train the semantic similarity model, while the latter only took weak supervisions from text. Also, the average time cost by the network-based approach was 10 times longer, since it took every fact about entities in class C as a possible representation for the adjective. By contrast, our approach leveraged statistic measures for generating candidates, which largely reduced the search space.

Also, we tested the performance change of each approach by setting varied thresholds for parameters. Figure 3 shows the precision-recall curves for the four approaches. The precision for Adj2ER-full kept larger than 60% when recall varied from 5% to 40%, which indicated that our approach can generate mappings of high quality for a considerable part of adjectives in natural language questions.

4.2 Experiments on Question Answering

Question Set. We built **AdjQuestions**, a question set containing 120 questions over structured data to verify if existing QA systems can benefit from our adjective mapping approach. We extracted all 70 questions that contain factoid adjective constraints from QALD-1 to QALD-9, and manually sampled 50 questions from Yahoo! Answers. For each question, a standard SPARQL query and its execution result over DBpedia-201510 were also provided.

Algorithm 1. Integrating Adj2ER into existing QA systems

Input: A natural language question Q
Output: A SPARQL query for the input question Q

procedure GENERATESPARQL(Q)
  S := ExistingQASystem.GENERATESPARQL(Q);
  for all (adj, C) pair in Q **do**
    Restrictions := Adj2ER(adj, C);
    if S contains a *restriction* $\in$ **Restrictions then** continue;
    S' := S;
    Remove all restrictions in S' which have semantic similarity ≥ 0.6 with adj;
    for all *restriction* $\in$ **Restrictions do**
      S^* := Resulting SPARQL for adding *restriction* to S';
      if S^* is a non-empty query **then** S := S^*; break;
  return S;

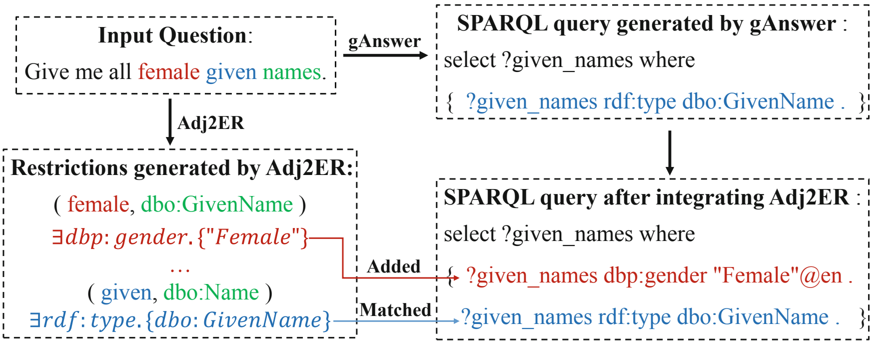

Fig. 4. Example for integrating Adj2ER into existing QA system

Table 6. QA results on AdjQuestions

	70 QALD questions			50 YA questions			Overall
	P	R	F1	P	R	F1	F1
gAnswer	30.49%	55.30%	29.75%	16.56%	36.26%	13.97%	23.18%
gAnswer + Adj2ER	**44.03%**	**62.25%**	**43.02%**	37.32%	**56.59%**	38.59%	**41.18%**
WDAqua	21.10%	26.64%	17.79%	23.53%	28.10%	22.04%	19.56%
WDAqua + Adj2ER	33.28%	43.86%	32.05%	**42.70%**	44.88%	**40.99%**	35.77%

Integrating Strategy. We integrated our adjective mapping approach into two state-of-the-art QA systems, namely gAnswer [11,26] and WDAqua [8], which ranked first and second in the QALD-9 challenge, respectively. The procedure for integrating Adj2ER into them is shown in Algorithm 1. An example to illustrate the procedure is shown in Fig. 4.

QA Results. Table 6 shows that, by integrating Adj2ER, gAnswer and WDAqua gained an improvement of 18.00% and 16.21% in macro F1-scores, respectively. Moreover, our integrating strategy modified 48.33% (58.33%) of SPARQL queries originally generated by gAnswer (WDAqua), and 33.33% (29.17%) questions gained an improvement in final F1-scores. Only 4.17% (5%) of questions had their F1-score decreased after integrating Adj2ER. It should be noticed that, sometimes even if the system understood the question correctly, it still cannot obtain the same result as the gold standard due to the difference in knowledge representations. For example, the final SPARQL query shown in Fig. 4 precisely expresses the meaning of the input question, and can be successfully executed over DBpedia with 728 answers returned. However, the SPRQAL query in gold standard expresses "female" in another way ($\exists$*dbo:gender.*{*dbr:Female*}), and its answer set is greatly different from ours. In order to test the real impact of our adjective understanding approach, we manually assessed whether each adjective is correctly interpreted. As a result, after integrating Adj2ER, the accuracy for interpreting adjectives improved from 32.50% (35.83%) to 61.66% (58.33%), which indicated that Adj2ER can greatly improve the performance of adjective understanding module in current QA systems.

5 Discussions

Limitations of Adj2ER. For the majority of adjectives used in the questions, our approach can generate the existential restrictions of high quality. However, there are still some adjectives that cannot be resolved by our approach, due to the following reasons: (i) a portion (3.04%) of compound adjectives rarely appear in Web corpora, such as "Chinese-speaking" and "non-extinct". They are also not covered by lexicons like WordNet and PPDB. Thus, Adj2ER did not collect enough related entities; (ii) a few adjectives should be mapped to more complex logic forms. For example, "widowed people" may be mapped to $dbo{:}Person \sqcap \exists dbo{:}spouse.\exists dbo{:}deathDate.\top$. Currently, our approach cannot generate such complex structures, due to the much larger search space and the difficulty in collecting related entities. We would like to consider dealing with these adjectives in the future.

It should also be noticed that, in this paper, we mainly focus on understanding adjectives for the KBQA task. From the perspective of adjective semantics [19], Adj2ER can handle most of intersective and subsective adjectives, but currently cannot handle scalar adjectives (e.g., "big cities") and intentional adjectives (e.g., "alleged criminal"), since these adjectives' meanings are not expressed in KBs, or the meanings are related to personal preferences. For example, when we say "big cities", we know that it represents a numerical constraint on cities' area or population, but there is no standard interpretation for "big". Such a question is considered inappropriate as a KBQA question, since even experts cannot provide consistent reference answers.

Error Analysis. There are several reasons causing the unsatisfactory result in QA. Firstly, 65% of error cases occurred in entity linking, relation mapping or query type detection, which are beyond the scope of this work. Particularly, several relational phrases, such as "official language", which should be mapped to simple relations, were detected as (adj, C) pairs by mistake. Secondly, in 23% of error cases Adj2ER generated inappropriate mappings for adjectives. Most of these adjectives are about time (e.g., past, current), and perhaps their meanings cannot be captured by existential restrictions. Finally, 12% of error cases were caused by our integrating strategy. The restrictions generated by Adj2ER were correct but added to wrong variables when there are multiple variables in the queries generated by the existing QA systems.

Other Findings. An interesting finding is that our approach can provide existential restrictions for different word senses of an adjective. For example, it provided both ∃*dbo:servingTemperature*.{*"Hot"*} and ∃*dbo:ingredient*.{*dbr:Chili_pepper*} for "hot" on class *dbo:Food*. The former existential restriction captures the meaning of "high temperature", while the latter one means "spicy". A word sense disambiguation method may be useful for better integration of Adj2ER in QA systems. Another interesting finding is that some existential restrictions, which were generated as candidates (before filtering), were not exact interpretation but entailed facts for most of the embodied entities, such as ∃*dbo:utcOffset*.{*"+8"*} for (Chinese, dbo:City). We would like to study how to exploit these candidates for information extraction in the future.

6 Related Work

This work is closely related to KBQA. Some existing systems [1,8,26] directly link adjectives or "adjective + noun" phrases to entities, classes or literals in the given KB, by using a similarity-based searching method or a pre-built lexicon. Although this solution can interpret "American actors" as "actors who are related to *dbr:United_States*", it cannot determine whether the phrase means "actors who have nationality *dbr:United_States*" or "actors who have death place *dbr:United_States*". Some recent systems [4,12] leverage the encode-and-compare networks to learn the semantic similarity between the natural language constraint expression and the target query component (in this work, the existential restriction). Our experiments showed that such solution may suffer from a huge search space and limited performance due to the lack of training data. Current QA systems cannot process adjectives in natural language questions precisely.

Semantic parsing is an important step in most QA systems, which translates natural language questions to structural queries, such as SPARQL, λ-DCS [5], CCG [6] and description logics [2]. Hu et al. [11] exploited semantic query graphs for question analysis based on the dependency structures of questions. Abujabal et al. [1] mapped questions to SPARQL templates automatically generated from question-answer pairs. However, all these approaches mainly focus on understanding proper nouns and relation phrases in questions, and do not have a specific step to recognize and interpret adjective constraints. Zhang et al. [25] studied the semantic interpretation of superlative expressions, by mapping each superlative adjective to a single relation in the KB with a neural network. However, this work cannot be applied to factoid adjective constraints, due to the diversity of knowledge representations and the lack of training data.

Entity linking and relation mapping are two essential steps in most of the semantic parsing approaches. Common entity linking approaches, e.g., [7,9], firstly generate candidates for possible entity mentions by running an exact string matching or lexical similarity based method over pre-built lexicons, and then focus on the problem of entity disambiguation. These approaches cannot be easily applied to adjective mapping, since the lexical gap between adjectives and structural knowledge is huge. The distant supervision method [15], which is frequently used to find mappings between relational phrases and KB relations, cannot handle the adjective mapping problem directly, as it is difficult to determine which facts in the KB express the meaning of the adjective.

In natural language processing area, there have been much work on clustering or identifying the meanings of adjectives [10,14]. Bakhshandeh and Allen [3] studied the problem of finding the aspect that an adjective describes through the WordNet glosses, such as "price" for "expensive". However, to the best of our knowledge, there is little work in studying the representations of adjectives over KBs. Walter et al. [23] proposed an approach for extracting adjective lexicalizations by analyzing the labels of objects occurring in DBpedia. However, this work only considers representations in form of $\exists r.\{a\}$, and requires a to be a meaningful string containing words related to the adjective. Our approach considers four forms of existential restrictions and is universal for all facts.

7 Conclusion

In this paper, we studied the problem of mapping factoid adjective constraints to existential restrictions over KBs. Our main contributions are listed below:

- We proposed a novel approach Adj2ER, which maps adjectives to existential restrictions or their negation forms in description logics.
- We leveraged statistic measures for generating candidate existential restrictions and supervised learning for filtering the candidates, which can largely reduce the search space and overcome the lexical gap.
- We created two question sets with adjectives used in QALD and Yahoo! Answers, and conducted experiments over DBpedia. Our experiments showed that Adj2ER generated mappings of high quality for most adjectives and significantly outperformed several alternative approaches. Furthermore, current QA systems gained an improvement of over 16% in F1-score by integrating our approach.

Understanding adjectives with KBs is still a difficult problem and deserves more attention. In future work, we plan to study other complex logic forms for adjective mapping. Also, we want to apply our approach to other tasks such as information extraction and question generation.

Acknowledgments. This work was supported by the National Key R&D Program of China (No. 2018YFB1004300), the National Natural Science Foundation of China (No. 61772264), and the Collaborative Innovation Center of Novel Software Technology and Industrialization. We would like to thank Xinqi Qian, Yuan Wang and Xin Yu for their helps in preparing evaluation.

References

1. Abujabal, A., Yahya, M., Riedewald, M., Weikum, G.: Automated template generation for question answering over knowledge graphs. In: WWW, pp. 1191–1200 (2017)
2. Baader, F., Calvanese, D., McGuinness, D.L., Nardi, D., Patel-Schneider, P.F.(eds.): The Description Logic Handbook: Theory, Implementation, and Applications. Cambridge University Press (2003)
3. Bakhshandeh, O., Allen, J.: From adjective glosses to attribute concepts: learning different aspects that an adjective can describe. In: IWCS, pp. 23–33 (2015)
4. Bao, J., Duan, N., Yan, Z., Zhou, M., Zhao, T.: Constraint-based question answering with knowledge graph. In: COLING, pp. 2503–2514 (2016)
5. Berant, J., Chou, A., Frostig, R., Liang, P.: Semantic parsing on freebase from question-answer pairs. In: EMNLP, pp. 1533–1544 (2013)
6. Bernardi, R.: The syntactic process: language, speech, and communication, mark steedman. J. Logic Lang. Inform. **13**(4), 526–530 (2004)
7. Deng, D., Li, G., Feng, J., Duan, Y., Gong, Z.: A unified framework for approximate dictionary-based entity extraction. VLDB J. **24**(1), 143–167 (2015)
8. Diefenbach, D., Singh, K., Maret, P.: WDAqua-core0: a question answering component for the research community. In: Dragoni, M., Solanki, M., Blomqvist, E. (eds.) SemWebEval 2017. CCIS, vol. 769, pp. 84–89. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-69146-6_8
9. Dubey, M., Banerjee, D., Chaudhuri, D., Lehmann, J.: EARL: joint entity and relation linking for question answering over knowledge graphs. In: Vrandečić, D., et al. (eds.) ISWC 2018. LNCS, vol. 11136, pp. 108–126. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-00671-6_7
10. Hartung, M., Frank, A.: Exploring supervised LDA models for assigning attributes to adjective-noun phrases. In: EMNLP, pp. 540–551 (2011)
11. Hu, S., Zou, L., Yu, J.X., Wang, H., Zhao, D.: Answering natural language questions by subgraph matching over knowledge graphs. IEEE Trans. Knowl. Data Eng. **30**(5), 824–837 (2018)
12. Luo, K., Lin, F., Luo, X., Zhu, K.Q.: Knowledge base question answering via encoding of complex query graphs. In: EMNLP, pp. 2185–2194 (2018)
13. Manning, C.D., Surdeanu, M., Bauer, J., Finkel, J.R., Bethard, S., McClosky, D.: The Stanford CoreNLP natural language processing toolkit. In: ACL, pp. 55–60 (2014)
14. Miller, G.A.: WordNet: a lexical database for English. Commun. ACM **38**(11), 39–41 (1995)
15. Mintz, M., Bills, S., Snow, R., Jurafsky, D.: Distant supervision for relation extraction without labeled data. In: ACL-IJCNLP, pp. 1003–1011 (2009)
16. Nakashole, N., Weikum, G., Suchanek, F.M.: PATTY: a taxonomy of relational patterns with semantic types. In: EMNLP-CoNLL, pp. 1135–1145 (2012)

17. Pavlick, E., Rastogi, P., Ganitkevitch, J., Durme, B.V., Callison-Burch, C.: PPDB 2.0: better paraphrase ranking, fine-grained entailment relations, word embeddings, and style classification. In: ACL, pp. 425–430 (2015)
18. Pennington, J., Socher, R., Manning, C.D.: GloVe: global vectors for word representation. In: EMNLP, pp. 1532–1543 (2014)
19. Pustejovsky, J.: Inference patterns with intensional adjectives. In: Joint ISO-ACL SIGSEM Workshop on Interoperable Semantic Annotation, pp. 85–89 (2013)
20. Unger, C., Bühmann, L., Lehmann, J., Ngomo, A.C.N., Gerber, D., Cimiano, P.: Template-based question answering over RDF data. In: WWW, pp. 639–648 (2012)
21. Unger, C., et al.: Question answering over linked data (QALD-5). In: CLEF (2015)
22. Usbeck, R., Gusmita, R.H., Ngomo, A.N., Saleem, M.: 9th challenge on question answering over linked data (QALD-9) (invited paper). In: ISWC Workshop on SemDeep-4/NLIWOD-4, pp. 58–64 (2018)
23. Walter, S., Unger, C., Cimiano, P.: Automatic acquisition of adjective lexicalizations of restriction classes: a machine learning approach. J. Data Semant. **6**(3), 113–123 (2017)
24. Yih, W., Chang, M., He, X., Gao, J.: Semantic parsing via staged query graph generation: question answering with knowledge base. In: ACL-IJCNLP, pp. 1321–1331 (2015)
25. Zhang, S., Feng, Y., Huang, S., Xu, K., Han, Z., Zhao, D.: Semantic interpretation of superlative expressions via structured knowledge bases. In: ACL-IJCNLP, pp. 225–230 (2015)
26. Zou, L., Huang, R., Wang, H., Yu, J.X., He, W., Zhao, D.: Natural language question answering over RDF: a graph data driven approach. In: SIGMOD, pp. 313–324 (2014)