Sematch: Semantic Entity Search from Knowledge Graph

Ganggao Zhu and Carlos A. Iglesias

Grupo de Sistemas Inteligentes,
Universidad Politécnica de Madrid, Spain
{gzhu,cif}@dit.upm.es

Abstract. As an increasing amount of the knowledge graph is published as Linked Open Data, semantic entity search is required to develop new applications. However, the use of structured query languages such as SPARQL is challenging for non-skilled users who need to master the query language as well as acquiring knowledge of the underlying ontology of Linked Data knowledge bases. In this article, we propose the Sematch framework for entity search in the knowledge graph that combines natural language query processing, entity linking, entity type linking and semantic similarity based query expansion. The system has been validated in a dataset and a prototype has been developed that translates natural language queries into SPARQL.

Keywords: Entity Search, Semantic Search, Query Expansion, Semantic Similarity, Knowledge Graph

1 Introduction

Increasing amounts of structured data are published as Linked Open Data (LOD) in the form of Resource Description Framework (RDF). The Knowledge Graph (KG) such as DBpedia [1] and YAGO2 [9] are examples that have succeeded in creating large general purpose RDF knowledge graphs on the Web of Data, whose knowledge is extracted from Wikipedia. Those initiatives have enabled the KG to change the web from a web of documents into a web of entities. Hence, apart from identifying a single entity based on its textual description, retrieving a list of entities from KG conforming user's specific information needs is also important for both web users and web applications. For example, when a student wants to compare universities in Spain or a web application needs to display all the universities in Spain, both cases require a list of entities of type University with the restriction of Location Spain.

However, querying a list of entities from these heterogeneous structured KGs is challenging for non-skilled users who need to master the syntax of a structured query language (such as SPARQL) and to acquire sufficient knowledge of the underlying ontology (schema and vocabulary). The ideal way for casual users to query from KGs is using Natural Language Interfaces (NLI), where users can express their information needs using Natural Language (NL) without being aware of the heterogeneous LOD vocabulary. The research in NLI for KGs has its roots in the application of traditional keyword-based information retrieval techniques to indexed RDF data such as the works in semantic search [22, 6]. Recent researches such as [25, 19, 8, 13, 7, 23, 21] have focused on advanced Question Answering (QA) techniques over KGs by translating NL queries into formal SPARQL queries. In this paper, we have restricted the queries to queries with just one relation, called Single Relation Type-based Queries (SRTQs) such as full sentence query *Give me all the universities located in Spain.* An abbreviated version of SRTQ can be expressed with keywords, i.e. *universities Spain.* This example of SRTQ can be rewritten as an equivalent conjunctive formal logic expression $?x \leftarrow (?x, is, University) \cap (?x, ?relation, Spain)$ where ontology class *University*, and instance *Spain* are restrictions on the variable x.

To clarify the task of semantic entity search for SRTQ, we give the formal definitions as follows. A Knowledge Graph K is a directed graph $G_k = \langle C, I, R, L, \tau \rangle$ [25], where C and I define the sets of *class* and *instance*; R and L are the sets of *relation* and *literal*; and τ is a function $(C \cup I) \times (C \cup I \cup L) \rightarrow R$ that defines all triples in K. Let Q a SRTQ expressed in NL. $Q = (q_1, q_c, q_i..q_n)$ is a bag of terms containing entity type mention q_c and entity instance mention q_i. Entity Linking is defined as $f_e : q_i \rightarrow e \in I$ and Type Linking is defined as $f_t : q_c \rightarrow t \in C$. The formal query $F : \langle e, t, \tau' \rangle$ over K is a graph G_f subsumed by G_k. From the definitions above, the entity search task for SRTQ can be modeled as: given Q, detect and link entity type t and entity instance e to K via f_e and f_t, constructing and executing formal queries $\{F\}$ over K to get desired entities.

For example, in the query described above *query(Spain, university)*, the results of this query are the entities whose entity type is *University* and have semantic relatedness (*located-in*) with the mentioned entity instance *Spain*. By linking *university* and *Spain* to their proper URIs in K, the formal query $< Spain, university, ?relation >$ can be translated into SPARQL query. By executing this query in a specific SPARQL endpoint, a list of university entities can be retrieved from a specific KG. Note that the relation terms such as *located-in* in the user query is not detected and mapped to R. The relation is used as a variable ($?relation$) in the query construction. In the current work, both the desired entities and the corresponding relation with the mentioned entity are returned as search results, where the relations are implemented as facets for faceted browsing for end users. One of our future works is to include relation information for improving the search performance.

In this paper, we propose a framework for semantic entity search in SRTQ over heterogeneous KGs. Since both the entity types mentioned in a user query and the ontology classes for annotating entities in KG (rdf:type) may be too general or too specific, a semantic similarity based type expansion algorithm is proposed and implemented for ontology class enrichment in SPARQL query construction in order to bridge this vocabulary gap. A dataset for SRTQ has been collected to evaluate both the Sematch framework and the proposed algorithm. The source code of Sematch prototype together with the implemented query expansion algorithm is published in github[1] including a working demo using DBpedia SPARQL endpoint. Consequently, the paper is organized as follows: In Section 2, we present the architecture and the whole process of Sematch framework. Then, in Section 3 we elaborate on our experimental setup and analyze our evaluation results. The related works are reviewed in Section 4. We close with concluding remarks and an outlook on future work in Section 5.

2 Sematch Framework

The overall architecture of Sematch framework is shown in Fig.1. The NL query processing component performs Natural Language Processing (NLP) tasks of tokenization, Part of Speech Tagging and Name Entity Recognition (NER) using NLTK[2]. Then, the entity linking component detects the named entity and maps it to instance URI of the KG. In the type expansion component, the type mentioned in the query is mapped to WordNet synsets and expanded based on WordNet taxonomy. Then, type synsets are mapped to ontology class URIs of the KG through *Synset ID Linkers*. Finally, SPARQL queries are generated based on the type and entity URIs obtained before in the *Query Engine*. In this section, we describe the details of entity linking, type expansion and the query graph generation.

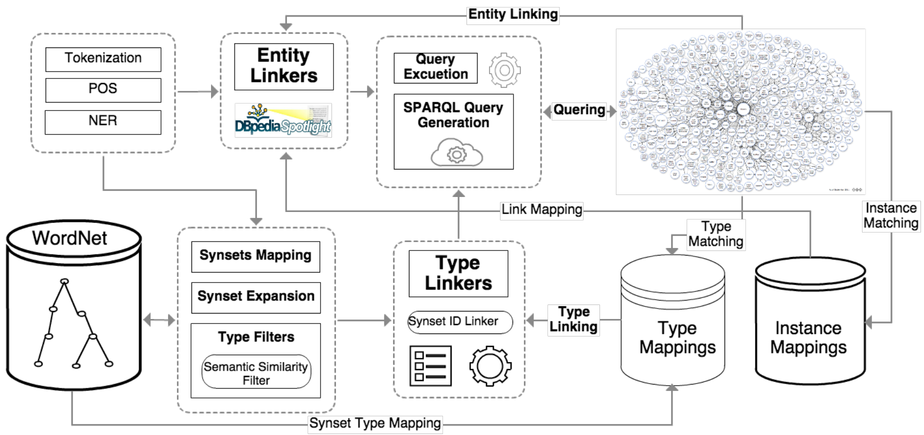

Fig. 1. Sematch Framework Overview

[1] https://github.com/gsi-upm/sematch
[2] http://www.nltk.org/

2.1 Entity Linking

The entity linking [17] component takes all the tokens except for stopwords. Those tokens are required because the task of entity linking not only links entity mentions that occur in query tokens to entries in the KG but also disambiguates entity mentions. Nevertheless, only the links of entities (Location, Person, etc.) recognized by the NER will be sent to the query construction engine. In the example query described above, the entity mention *Spain* is detected and mapped to URI *DBpedia:Spain*. The current Sematch prototype uses DBpedia Spotlight [14] web service for entity linking.

Entity linking annotates the name entities with URIs of a specific KG. In order to make our system available to different KGs, instance link mapping (owl:sameAs) data[3] is used to transform the URIs from a specific entity linking system to the URIs that is used in other KGs. The proper entity URI is selected according to the configuration of the SPARQL endpoints. In case of multiple entity URIs are given, all of them are sent to the query engine.

2.2 Semantic Similarity Based Type Expansion

This subsection presents the details of translating q_c into entity type t. The query q_c is first mapped to a list of WordNet [15] synsets based on their specific sense in the query through Word Sense Disambiguation (WSD) using an adaptation of Lesk Algorithm [5]. WordNet provides a taxonomy of synsets representing the meaning of words. A set of words that share one common sense is called a synset. Unlike conventional IR using synsets for synonym expansion, *synsets mapping* reconciles words to synsets with specific meaning. Thus, the types for describing things are processed at the semantic level (meanings) rather than at the lexical level (terms). WordNet provides relations between synsets such as hypernymy/hyponymy (i.e., the relation between a sub-concept and a super-concept) and holonymy/meronymy (i.e., the relation between a part and the whole). The synset type seeds from *synsets mapping* are expanded based on WordNet hypernyms/hyponyms.

Though the recall can be increased by expanding with hypernyms/hyponyms, it is also important to guarantee a certain level of precision. Since semantic similarity measures the proximity between synsets mainly based on hierarchical relation (Is-A), semantic similarity is applied in type expansion for optimizing its precision. Let Σ_{synset} be all the noun synsets in WordNet. The *semantic similarity function* $sim: \quad \Sigma_{synset} \times \Sigma_{synset} \rightarrow [0,1]$ is defined as a list of the state of art semantic similarity measures including edge counting based measures *path* [16], *wup* [24], *lch* [11], and information content based measures *res* [18], *jcn* [10], *lin* [12]. In this work, the information content (IC) is computed as $IC(w) = -\log P(w)$ where $P(w)$ is the probability in finding w in Brown Corpus of American English [18]. A threshold $\eta \in [0,1]$ is used to establish the

[3] http://wiki.dbpedia.org/Downloads2014

semantic similarity between two synsets: $sim(s_1, s_2) >= \eta$. Let Σ_{seeds} denote the synset type seeds from *synsets mapping* component, the semantic similarity based type expansion algorithm is defined in Algorithm 1. The final algorithm returns a list of expanded synsets which are also merged into a synset type list. A synset type list is a set of synsets including seed synsets and expanded synsets.

Algorithm 1 Semantic Similarity Based Synset Expansion

procedure EXPANSION($\Sigma_{seeds}, \eta, sim$)
  $\Sigma_{result} \leftarrow \emptyset$
  for all $s \in \Sigma_{seeds}$ **do**
    EXPAND($s, s, \eta, sim, \Sigma_{result}$)
  end for
  return Σ_{result}
end procedure
procedure EXPAND(c, s, η, sim, Σ)
  $\Sigma \leftarrow c$
  for all $x \in hypernyms(c)$ **do**
    if $x \notin \Sigma$ and $sim(s, x) >= \eta$ **then**
      EXPAND(x, s, η, sim, Σ)
    end if
  end for
  for all $y \in hyponyms(c)$ **do**
    if $y \notin \Sigma$ and $sim(s, y) >= \eta$ **then**
      EXPAND(y, s, η, sim, Σ)
    end if
  end for
end procedure

Before constructing the query, expanded synsets have to be transformed into proper URIs with *Synset ID Linkers*. A *Synset ID Linker* is an implementation of the Type Linking function $f_t : q_c \rightarrow t \in C$, which links synsets to the Linked Data ontology classes by looking up the type mapping data[4]. The type mapping data[5] is derived from *yagoDBpediaClasses* and *yagoWordnetIds* in YAGO2. In this form, URIs of ontology classes from different knowledge graphs are unified by WordNet synsets based on their meanings. Some DBpedia ontology[6] classes are aligned to the type mapping data based on the data[7] provided by YAGO2. Ontology classes in other knowledge graphs can also be aligned to WordNet synsets based on the current type mapping data using ontology alignment techniques [3]. After type expansion, the entity mention *university* is expanded into a

[4] 108286163,university.n.01,http://dbpedia.org/ontology/University, http://dbpedia.org/class/yago/University108286163
[5] Mapping Data contains 68423 entries of synsets and YAGO ontology classes.
[6] 145 DBpedia ontology classes are aligned to the mapping data.
[7] http://www.mpi-inf.mpg.de/departments/databases-and-information-systems/research/yago-naga/yago/linking/

list of ontology class URIs. In the next subsection, we describe how to construct the formal query F using e and t based on predefined graph patterns.

2.3 Query Graph Generation

Given URIs of e and t, SPARQL queries can be constructed using Graph Pattern Collection (GPC) for SRTQ derived from the graph patterns defined in [19]. GPC is a set of triple patterns and is defined as: $GPC = \big\{(s,p,o)|(s \in I \vee s = variable) \wedge (p = variable) \wedge (o \in I \vee o \in C \vee o = variable)\big\}$. The Graph Pattern Set (GPS) is a set of all GPCs and is represented as $GPS = \{g|g = GPC\}$ which are $\{GPC_1, GPC_2, GPC_3, GPC_4, GPC_5, GPC_6\}$. The details of the graph patterns for each GPC are illustrated in Fig.2. In these pattern collections, symbols preceded by question marks denote variables and symbols without question marks are t (entity type) and e (entity instance).

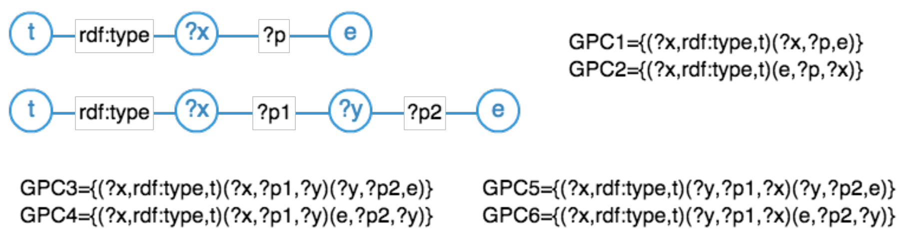

Fig. 2. Graph Pattern Collections

Algorithm 2 Query Generation and Execution

procedure ENGINE(t, e, GPS, G_k)
    $\Sigma_{result} \leftarrow \emptyset$
    $T \leftarrow Union(t)$
    for all $GPC \in GPS$ **do**
        $F \leftarrow construct(GPC, T, e)$
        $\Sigma_{result} \leftarrow query(F)$
    end for
    return $HashSet(\Sigma_{result})$
end procedure

Those patterns are only valid for certain combinations with t. The goal of type expansion is to generate adequate type URIs. The *Union* syntax of SPARQL query language is used to combine all the available type URIs such as (?x, rdf:type, t1) Union (?x, rdf:type, t2). GPC_1 and GPC_2 represent direct semantic relation with the mentioned entity, which is shown in the first pattern graph of Fig. 2. Semantic relation expansion is represented by $\{GPC_3, GPC_4\}$ and $\{GPC_5, GPC_6\}$. The relation expansion is included because the relations between entities in the KG can be transitive relations. Finally, t and e are constructed into F by being filled into all GPCs. The queries are sent to the user specified SPARQL endpoint and the results are unified by removing repetitions. The query construction and execution process are illustrated in Algorithm 2. The example of GPC_1 for constructing the query *university Spain* is illustrated as below:

```
SELECT DISTINCT ?x ?p WHERE {
{ ?x rdf:type dbpedia:University> } UNION
{ ?x rdf:type yago:University108286163 } UNION
{ ?x rdf:type yago:CityUniversity103036244 } UNION
{ ?x rdf:type dbpedia:EducationalInstitution> } UNION
{ ?x rdf:type yago:EducationalInstitution108276342 } .
?x ?p <http://dbpedia.org/resource/Spain> .
} GROUP BY ?x
```

3 Evaluation

In this section, we evaluate the performance of Sematch framework. The evaluation aims to achieve three goals: 1) compare the effectiveness of different semantic similarity methods for type expansion 2) evaluate the feasibility of semantic similarity based type expansion; 3) compare the effectiveness of relation expansion by using different numbers of GPCs.

3.1 Datasets

We have collected a dataset for SRTQs from a dataset for entity search in DBpedia [4] which contained data from several campaigns, including INEX-XER, TREC Entity, SemSearch ES, SemSearch LS, QALD-2, and INEX-LD. Table.1 illustrates our 29 SRTQs. For convenience, we have also shown the queries with detected entity type mention and entity instance mention.

3.2 Evaluation Metrics

Precision and recall were used as our metrics. Assuming A is the relevant set of entities for the query that is provided in dataset, and B is the set of retrieved entities by running Sematch, the precision and recall can be defined as follows:

$$Recall = \frac{|A \cap B|}{|A|} \tag{1}$$

Table 1. The Query Dataset Used in Evaluation.

ID	Source	Query	Type	Entity
1	INEX_LD-20120131	vietnam travel national park	park	dbpedia:Vietnam
2	INEX_LD-20120132	vietnam travel airports	airports	dbpedia:Vietnam
3	INEX_LD-2010004	Indian food	food	dbpedia:India
4	INEX_XER-62	Neil Gaiman novels	novels	dbpedia:Neil_Gaiman
5	INEX_XER-72	films shot in Venice	film	dbpedia:Venice
6	INEX_XER-79	Works by Charles Rennie Mackintosh	works	dbpedia:Charles_Rennie_Mackintosh
7	INEX_XER-86	List of countries in World War Two	countries	dbpedia:World_War_II
8	INEX_XER-91	Paul Auster novels	novels	dbpedia:Paul_Auster
9	INEX_XER-108	State capitals of the United States of America	capitals	dbpedia:United_States
10	INEX_XER-124	Novels that won the Booker Prize	novels	dbpedia:Man_Booker_Prize
11	INEX_XER-125	countries which have won the FIFA world cup	countries	dbpedia:FIFA_World_Cup
12	INEX_XER-133	EU countries	countries	dbpedia:European_Union
13	INEX_XER-139	Films directed by Akira Kurosawa	film	dbpedia:Akira_Kurosawa
14	INEX_XER-140	Airports in Germany	airports	dbpedia:Germany
15	INEX_XER-141	Universities in Catalunya	university	dbpedia:Catalonia
16	QALD2_te-6	Give me all professional skateboarders from Sweden	skateboarders	dbpedia:Sweden
17	QALD2_te-17	Give me all cars that are produced in Germany	car	dbpedia:Germany
18	QALD2_te-28	Give me all movies directed by Francis Ford Coppola	movie	dbpedia:Francis_Ford_Coppola
19	QALD2_te-39	Give me all companies in Munich	companies	dbpedia:Munich
20	QALD2_te-60	Give me a list of all lakes in Denmark	lakes	dbpedia:Denmark
21	QALD2_te-63	Give me all Argentine films	film	dbpedia:Argentina
22	QALD2_te-82	Give me a list of all American inventions	invention	dbpedia:United_States
23	QALD2_tr-16	Give me the capitals of all countries in Africa	capitals	dbpedia:Africa
24	QALD2_tr-53	Give me all presidents of the United States	presidents	dbpedia:United_States
25	QALD2_tr-63	Give me all actors starring in Batman Begins	actors	dbpedia:Batman_Begins
26	QALD2_tr-68	Which actors were born in Germany?	actors	dbpedia:Germany
27	QALD2_tr-70	Give me all films produced by Hal Roach	film	dbpedia:Hal_Roach
28	QALD2_tr-78	Give me all books written by Danielle Steel	book	dbpedia:Danielle_Steel
29	QALD2_tr-84	Give me all movies with Tom Cruise	movies	dbpedia:Tom_Cruise

$$Precision = \frac{|A \cap B|}{|B|} \tag{2}$$

where $|.|$ gives the size of the set and $|A \cap B|$ is the set of entities that are both relevant and retrieved. Fig.3 illustrates the counts of expanded synsets using different semantic similarity methods as threshold varying from 0.6 to 1 with interval of 0.05. The semantic similarity methods *wup* and *path* have the same performance in expanding synsets so we only compare the method of *wup*, *lch*, *res*, *jcn*, and *lin*. In order to limit the maximum number of expanded synsets under 50, the thresholds of 0.9, 1.0 are chosen where 1.0 represents the baseline without expansion and 0.9 represents the type expansion. Furthermore, we use two sets of GPCs for comparing which are $gp1 = \{GPC_1, GPC_2\}$ and $gp2 = \{GPC_1, GPC_2, GPC_3, GPC_4\}$. The direct relation between desired entity and mentioned entity is represented by $gp1$, while $gp2$ represents relation expansion. We use the DBpedia SPARQL endpoint[8] to execute SPARQL queries. The experiment results are shown in the following section.

3.3 Results

Within the experimental configuration defined in the previous subsections, each query in Table 1 has been executed 20 times with two thresholds (th=0.9 and th=1.0), two sets of GPCs (gp1 and gp2), and five semantic similarity measures.

[8] http://dbpedia.org/snorql/

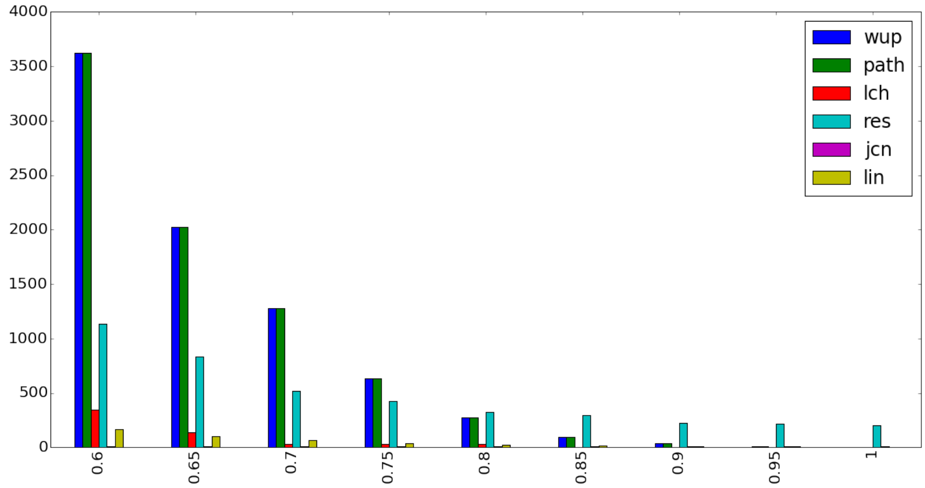

Fig. 3. Synset Expanding based on Thresholds

However, among those queries, the current prototype of Sematch is unable to answer the queries 5, 6, 8, 11, 22, 23 and 28. Thus, we have collected the results of 76% queries in the evaluation dataset. For each of those queries, 20 precision and recall values are collected. The average of those values have been illustrated in Table 2 with the corresponding settings. Each column of this table represents the specific semantic similarity measures which are *wup* [24], *lch* [11], *res* [18], *jcn* [10] and *lin* [12]. Each row of the table represents the specific settings of threshold and GPCs. For each cell, the average precision and recall are presented as (precison, recall) correspondingly.

Table 2. Average Recall and Precision

settings	wup	lch	res	jcn	lin
th=0.9 gp1	(0.33,0.42)	(0.46,0.41)	(0.40,0.42)	(0.40,0.42)	(0.39,0.42)
th=0.9 gp2	(0.003, 0.66)	(0.007,0.66)	(0.004,0.7)	(0.006,0.66)	(0.006,0.66)
th=1.0 gp1	(0.46,0.4)	(0.46,0.41)	(0.41,0.41)	(0.40,0.42)	(0.42,0.40)
th=1.0 gp2	(0.007,0.66)	(0.007,0.66)	(0.005,0.67)	(0.006,0.66)	(0.007,0.66)

The results have shown that the Sematch Framework can answer a moderate proportion of SRTQs (76%) and have promising performance in retrieving entities from KG. Each column of Table.2 has shown that as type or relation expanding the recall increases while the precision decreases. The semantic similarity based type expansion algorithm can improve recall and guarantee a certain level of precision. Since there is no control in relation expansion, though the recall has improved a lot, the precision becomes unacceptable by including too many irrelevant entities. Nevertheless, due to significant improvement of the recall, further research will focus on limiting irrelevant entities by automatically filtering those irrelevant relations in order to guarantee the precision. By comparing each row, it has been shown that the semantic similarity measure *lch, jcn* is better in keeping better precision, but with lower improvement of recall. While *wup, res, lin* are promising in improving recall. Fig.4 has shown that decreasing the threshold resulted in tremendous synsets and longer execution time. Further research is also required to keep reducing the irrelevant types and decreasing the execution time.

4 Related Work

Several NLI systems have been developed for keyword-based search or QA over KG. Semantic keyword-based search system Sindice [22] is an adaptation of conventional document retrieval approach for RDF data. Keyword-based entity search system Falcons [6] relies on matching query keywords in indexed terms. SPARK [25] translates keyword queries into formal logic queries to facilitate end users to perform semantic search. Treo [8] combined entity search, semantic relatedness and spreading activation search to query over LOD using NL queries. PowerAqua [13] is an ontology-based QA system which can combine information from heterogeneous LOD. FREyA [7] uses syntactic parsing in combination with the ontology-based lookup, as well as user interaction in order to interpret the question. Unger et al. [23] presented a QA system relying on deep linguistic analysis in generating SPARQL templates for answering more complex questions. SINA [21] is a keyword search system that can perform QA tasks by transforming keywords or NL queries into conjunctive SPARQL queries over LOD sources.

Sematch is a keyword-based entity search system especially for answering SRTQs aiming to retrieve a list of entities. It followed the approach [19] in which SPARQL queries are constructed from mapping keywords to LOD URIs and filling URIs into predefined graph patterns. Sematch adopted the idea of using WordNet taxonomy for interlinking entity type vocabulary like the work [3] and proposed semantic similarity based type expansion algorithm for enriching type information in generating SPARQL queries. Query expansion for LOD has also been proposed in [2] and [20]. Augenstein et al. [2] mainly focused on mapping keywords to LOD and relying on KG for query expansion. Shekarpour et al. [20] used machine learning approaches to combine expansion features from both WordNet and LOD and applied them in semantic search. Sematch focused on expanding entity types with WordNet hypernyms/hyponyms and using semantic similarity measures to optimize precision.

5 Conclusions and Future Work

In this paper, we have defined SRTQ and entity search tasks for SRTQ. A framework for answering SRTQ has been proposed by combining conventional NLP techniques NER, WSD and LOD techniques such as Entity Linking. The framework is designed to be extensible for including more advanced approaches both in NLP and LOD for solving SRTQ. A prototype system Sematch has been implemented and evaluated under the SRTQ evaluation dataset, which have been collected from several LOD campaigns in semantic search and QA. The evaluation results have shown that the Sematch system has promising performance in answering SRTQ and the proposed semantic similarity based type expansion algorithm can improve the entity search recall while keeping certain level of precision. Moreover, it has been shown that the relation expansion in query graph generation has a significant improvement in search recall though precision become unacceptable. Consequently, one of the future works will be developing advanced approaches to guarantee the search precision while expanding relations. Furthermore, more researches will be followed in refining the semantic similarity based type expansion algorithm to optimize both the execution time and search performance. Developing approaches to combine WSD, NER, and Entity linking for disambiguation jointly are also possible future works.

Acknowledgments. This work has been partially funded by the European project SmartOpenData (FP7-ENV-2013 Project 6038) under grant no. 603824.

References

1. Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., Ives, Z.: Dbpedia: A nucleus for a web of open data. In: The Semantic Web, LNCS, vol. 4825, pp. 722–735. Springer Berlin Heidelberg (2007)
2. Augenstein, I., Gentile, A., Norton, B., Zhang, Z., Ciravegna, F.: Mapping keywords to linked data resources for automatic query expansion. In: The Semantic Web: ESWC 2013 Satellite Events, LNCS, vol. 7955, pp. 101–112. Springer Berlin Heidelberg (2013)
3. Ballatore, A., Bertolotto, M., Wilson, D.C.: Linking geographic vocabularies through wordnet. Annals of GIS 20(2), 73–84 (2014)
4. Balog, K., Neumayer, R.: A test collection for entity search in dbpedia. In: 36th International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 737–740. SIGIR '13, ACM, New York (2013)
5. Banerjee, S., Pedersen, T.: An adapted lesk algorithm for word sense disambiguation using wordnet. In: Computational linguistics and intelligent text processing, pp. 136–145. Springer (2002)
6. Cheng, G., Qu, Y.: Searching linked objects with falcons: Approach, implementation and evaluation. Int. J. Semantic Web Inf. Syst. 5(3), 49–70 (2009)
7. Damljanovic, D., Agatonovic, M., Cunningham, H.: Freya: An interactive way of querying linked data using natural language. In: The Semantic Web: ESWC 2011 Workshops, LNCS, vol. 7117, pp. 125–138. Springer Berlin Heidelberg (2012)
8. Freitas, A., Oliveira, J.G., Curry, E., ORiain, S., da Silva, J.C.P.: Treo: combining entity-search, spreading activation and semantic relatedness for querying linked data. In: Proc. of 1st Workshop on Question Answering over Linked Data (QALD-1) at the 8th Extended Semantic Web Conference (ESWC 2011) (2011)
9. Hoffart, J., Suchanek, F.M., Berberich, K., Lewis-Kelham, E., de Melo, G., Weikum, G.: Yago2: Exploring and querying world knowledge in time, space, context, and many languages. In: 20th International Conference Companion on World Wide Web. pp. 229–232. ACM, New York (2011)

10. Jiang, J.J., Conrath, D.W.: Semantic similarity based on corpus statistics and lexical taxonomy. Computational Linguistics cmp-lg/970(Rocling X), 15 (1997)
11. Leacock, C., Chodorow, M.: Combining local context and wordnet similarity for word sense identification. WordNet: An electronic lexical database 49(2), 265–283 (1998)
12. Lin, D.: An information-theoretic definition of similarity. In: Fifteenth International Conference on Machine Learning. pp. 296–304. ICML '98, Morgan Kaufmann Publishers Inc., San Francisco (1998)
13. Lopez, V., Fernández, M., Motta, E., Stieler, N.: Poweraqua: Supporting users in querying and exploring the semantic web. Semantic Web 3(3), 249–265 (2012)
14. Mendes, P.N., Jakob, M., García-Silva, A., Bizer, C.: Dbpedia spotlight: Shedding light on the web of documents. In: 7th International Conference on Semantic Systems. pp. 1–8. I-Semantics '11, ACM, New York (2011)
15. Miller, G., Fellbaum, C.: Wordnet: An electronic lexical database, vol. 16. MIT Press Cambridge (1998)
16. Rada, R., Mili, H., Bicknell, E., Blettner, M.: Development and application of a metric on semantic nets. IEEE Transactions on Systems, Man, and Cybernetics 19(1), 17–30 (1989)
17. Rao, D., McNamee, P., Dredze, M.: Entity linking: Finding extracted entities in a knowledge base. In: Multi-source, Multilingual Information Extraction and Summarization, pp. 93–115. Springer (2013)
18. Resnik, P.: Semantic similarity in a taxonomy: An information-based measure and its application to problems of ambiguity in natural language. Journal of Artificial Intelligence Research 11(95), 95–130 (1999)
19. Shekarpour, S., Auer, S., Ngomo, A., Gerber, D., Hellmann, S., Stadler, C.: Keyword-driven sparql query generation leveraging background knowledge. In: Web Intelligence and Intelligent Agent Technology (WI-IAT), 2011 IEEE/WIC/ACM International Conference on. vol. 1, pp. 203–210 (Aug 2011)
20. Shekarpour, S., Hoffner, K., Lehmann, J., Auer, S.: Keyword query expansion on linked data using linguistic and semantic features. In: Semantic Computing (ICSC), 2013 IEEE Seventh International Conference on. pp. 191–197 (Sept 2013)
21. Shekarpour, S., Marx, E., Ngomo, A.C.N., Auer, S.: Sina: Semantic interpretation of user queries for question answering on interlinked data. Web Semantics: Science, Services and Agents on the World Wide Web (2014)
22. Tummarello, G., Delbru, R., Oren, E.: Sindice.com: Weaving the open linked data. In: The Semantic Web, LNCS, vol. 4825, pp. 552–565. Springer Berlin Heidelberg (2007)
23. Unger, C., Bühmann, L., Lehmann, J., Ngonga Ngomo, A.C., Gerber, D., Cimiano, P.: Template-based question answering over rdf data. In: 21st International Conference on World Wide Web. pp. 639–648. ACM, New York (2012)
24. Wu, Z., Palmer, M.: Verbs semantics and lexical selection. In: Proceedings of the 32nd annual meeting on Association for Computational Linguistics. pp. 133–138. ACL '94, Association for Computational Linguistics, Stroudsburg, PA, USA (1994)
25. Zhou, Q., Wang, C., Xiong, M., Wang, H., Yu, Y.: Spark: Adapting keyword query to semantic search. In: The Semantic Web, LNCS, vol. 4825, pp. 694–707. Springer Berlin Heidelberg (2007)