Exploring Term Networks for Semantic Search over RDF Knowledge Graphs

Edgard Marx[1,3(✉)], Konrad Höffner[1], Saeedeh Shekarpour[4], Axel-Cyrille Ngonga Ngomo[1], Jens Lehmann[2], and Sören Auer[2]

[1] AKSW, University of Leipzig, Leipzig, Germany
marx@infomatik.uni-leipzig.de, konrad.hoeffner@uni-leipzig.de
[2] Computer Science Institute, University of Bonn, Bonn, Germany
{lehman,auer}@cs.uni-bonn.de
[3] Instituto de Pesquisa e Desenvolvimento Albert Schirmer, Teófilo Otoni, Brazil
[4] Knoesis Research Center, Fairborn, OH, USA
saeedeh@knoesis.org

Abstract. Information retrieval approaches are considered as a key technology to empower lay users to access the Web of Data. A large number of related approaches such as Question Answering and Semantic Search have been developed to address this problem. While Question Answering promises more accurate results by returning a specific answer, Semantic Search engines are designed to retrieve the best top-K ranked resources. In this work, we propose `*path`, a Semantic Search approach that explores term networks for querying RDF knowledge graphs. The adequacy of the approach is evaluated employing benchmark datasets against state-of-the-art Question Answering as well as Semantic Search systems. The results show that `*path` achieves better F_1-score than the currently best performing Semantic Search system.

1 Introduction

The growth of Semantic Web technologies has led to the publication of large volumes of data. Approximately 10000 *Resource Description Framework (RDF)*[1] datasets are available via public data portals.[2] However, retrieving desired information from datasets still poses a significant challenge. Lay users cannot be expected to make themselves familiar with the underlying query languages and modeling structures.

A major challenge is the efficient retrieval of the resource that best represents the user's intent via natural language (NL) keyword queries. Relying solely on off-the-shelf triple stores or document retrieval may lead to poor performance or precision (see Sect. 5). To address this problem, we propose an approach for Semantic Search RDF knowledge graphs by exploring its Term Network. A Term Network (see Sect. 4) is a graph whose vertices are labeled terms. Overall, our contributions are as follows:

[1] http://www.w3.org/RDF.
[2] http://lodstats.aksw.org/.

E. Garoufallou et al. (Eds.): MTSR 2016, CCIS 672, pp. 249–261, 2016.
DOI: 10.1007/978-3-319-49157-8_22

- We develop a new formal model for Semantic Search (SemS) based on Term Networks;
- We present a ranking method that increases the precision on retrieving RDF data;
- We compare our approach with state of the art SemS techniques on the QALD-4 [17] benchmark and show that we achieve a higher F_1-score.

The rest of this paper is organized as follows: The related work is reviewed in Sect. 2. Section 3 defines the preliminaries. Section 4 describes the `*path` model. Section 5 outlines the evaluation and discusses the results. Finally, Sect. 6 concludes giving an outlook of potential future work.

2 Related Work

Information retrieval (IR) over Linked Data is an active and diverse research field with many existing related work focusing on designed for different environments, diverging in complexity and precision. The related work can be mainly categorized in two types of approaches that recover information from Linked Data Knowledge Graphs (KGs, see Definition 1): (1) by using conventional IR techniques and (2) by answering natural language questions. While the use of time efficient traditional IR systems lacks the ability to deal with complex queries, they are usually faster. Wang et al. [19] shows that pure traditional IR engines are faster than the combination of a triple store with a full-text index. However, both models explore the semantics of an NL query for delivering the response by applying statistics measures and heuristics in the KG.

Semantic Search (SemS) approaches aim to retrieve the top-k ranked resources for a given NL input query. Swoogle [3], introduces a modified version of PageRank that takes into account the types of the links between ontologies. Sindice [10], Falcons [2] and Sig.ma [16] explores traditional document retrieval to index and locate relevant sources and/or resources. Sindice is a search engine that can retrieve documents containing a given statement. Falcon, uses a built-in ranking mechanism for entity ranking while Sig.ma allows the use of constraints to query for particular classes and/or properties. In all cases, the structure and semantics are not taken into account during the matching phase.

YAHOO! BNC [4] used a local, per property, term frequency as well as a global term frequency. It also applied a boost based on the number of matched query terms. Umass [4] explored existing ranking functions applied to four field types: (1) title; (2) name; (3) dbo:title, and; (4) all others. The fields were weighted separately with a specific boost applied to each of them.

Later, Blanco et al. [1] proposed a modified version of BM25F ranking function adapted for RDF data. The function was applied to a horizontal pairwise index structure composed of the subject and its property values. However, the most important feature in the proposed structure is the possibility to assign different weights to predicates. The proposed adaptation is implemented in the $\text{Glimmer}_{\text{Y!}}$ engine and is shown to be time efficient as well as outperforms other state-of-the-art methods in ranking RDF resources.

Recently, Virgilio et al. [18] introduced a distributed technique for SemS on RDF data using MapReduce. The method uses a distributed index of RDF paths. The proposed strategy returns *the best top-k answers in the first k generated results.* The retrieval is done by evaluating the paths containing the terms of the query using two strategies: (1) Linear and (2) Monotonic. (1) The Linear strategy uses only the high ranked path(s). As a consequence, it does not produce an optimum solution but has linear complexity with respect to the size of matched entities. (2) The Monotonic strategy uses all matched paths and, thus, produces better results. Intuitively, measuring all suitable paths from all entities is less time efficient. Please refer to the work of Mangold et al. [8] for a more detailed analysis of SemS approaches.

One of the biggest challenges in SemS method lies in evaluating the relatedness between the terms in a KG and an NL query. Document retrieval engines rely on term frequency weighting, which is based on the assumption, that the more frequently a term occurs, the more related it is to the topic of the document [7]. While good retrieval performance needs to take the frequency into account, it suffers from frequent yet unspecific words such as "the", "a" or "in". Inverse document frequency corrects this by diminishing the weight of words that are frequently occurring in the corpus, leading to the combined term frequency–inverse document frequency (tf-idf) [15] to score documents for a query.

3 Preliminaries

We begin by introducing a formal definition of the RDF model. Thereafter, we introduce fundamental concepts that are required for full understanding of the rest of the paper.

RDF[3] is a standard for describing Web resources. A resource can refer to any physical or conceptual thing, such as a Web site, a person or a device. The RDF data model expresses statements about resources in the form of subject-predicate-object triples. The subject denotes a resource; the predicate expresses a property (of the subject) or a relationship (between subject and object); the object is either a resource or literal. Resources are identified with IRIs, a generalization of URIs, while literals are used to identify values such as numbers and dates by means of a lexical representation.

Definition 1 (RDF knowledge Graph, KG). *Formally, let K be a finite RDF knowledge graph (KG). K can be regarded as a set of triples $(s, p, o) \in (\mathcal{I} \cup \mathcal{B}) \times \mathcal{P} \times (\mathcal{I} \cup \mathcal{L} \cup \mathcal{B})$, where $\mathcal{R} = \mathcal{I} \cup \mathcal{B}$ is the set of all RDF resources $r \in \mathcal{R}$ in the KG, $\mathcal{I}$ is the set of all IRIs, $\mathcal{B}$ is the set of all blank nodes, $\mathcal{B} \cap \mathcal{I} = \emptyset$. $\mathcal{P}$ is the set of all predicates, $\mathcal{P} \subseteq \mathcal{I}$. $\mathcal{L}$ is the set of all literals, $\mathcal{L} \subset \Sigma^*$ and $\mathcal{L} \cap \mathcal{I} = \emptyset$, where Σ is the unicode alphabet. $\mathcal{E}$ is the set of all entities, $\mathcal{E} = \mathcal{I} \cup \mathcal{B} \setminus \mathcal{P}$. An RDFTerm φ refers to any edge label $p \in P$ or vertex in the KG $\varphi \in (\mathcal{I} \cup \mathcal{B} \cup \mathcal{L})$. A KG is modeled as a directed labeled graph $\mathcal{G} = (\mathcal{V}, \mathcal{D})$, where $\mathcal{V} = \mathcal{E} \cup \mathcal{L}$, $\mathcal{D} \subseteq \mathcal{E} \times (\mathcal{E} \cup \mathcal{L})$ and the labeling function[4] of the edges is a mapping $\lambda : \mathcal{D} \mapsto \mathcal{P}$. We disregard literal language tags and data types.*

[3] https://www.w3.org/TR/REC-rdf-syntax/.

[4] Not to be confused with rdfs:label.

Figure 1 shows an excerpt of a KG where a literal vertex $v_i \in \mathcal{L}$ (respectively a resource vertex $v_i \in \mathcal{R}$) is illustrated by a rectangle, respectively an oval. Each edge between two vertices corresponds to a triple, where the first vertex is called the subject, the labeled edge the predicate and the second vertex the object. For example, (e2) $\xrightarrow{\texttt{rdfs:label}}$ [Mona Lisa] corresponds to the triple <e2, rdfs:label, "Mona Lisa">.

In this work, we address the problem of SemS systems that aim to retrieve the top-k ranked entities representing the intention behind an NL user query.

Definition 2 (Natural Language Query). *A NL query $q \in \Sigma^*$ is a user given keyword string expressing a factual information needed.*

4 Approach

For many years, scientists from the most diverse fields of cognitive science have tried to explain and reproduce the human cognition system, including psychology, neuroscience, philosophy, linguistics and artificial intelligence. While diverse theories have been developed, a commonly shared idea is that knowledge is organized as a network [12]. Hudson et al. [6] go further and states that grammar is organized as a network as well. According to Hudson's work, the syntactic structure of a sentence consists of a network of dependencies between single terms. Thus, everything that needs to be said about the syntactic structure of a sentence can be represented in such a network. Hudson explores Saussure's [13] idea that "*language is a system of interdependent terms in which the value of each term results solely from the simultaneous presence of the others*". He also argues about the psycholinguistic evidence for the use of *spreading activation* in supporting knowledge reasoning. However, according to Hudson et al., the main challenge is finding out how the activation occurs in mathematical terms [6]. Our intuition is that as the KG contains a network of terms formed by the label (e.g. rdfs:label) of the RDFTerms—properties, classes and entities—they can be used to query.

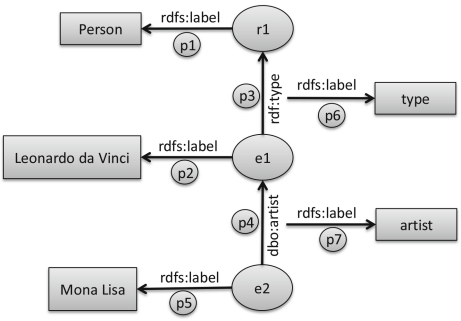

Fig. 1. An excerpt of a KG. The label of rdfs:label properties were omitted for simplification.

Definition 3 (Term). *A term*[5] *can be a* word or a phrase used to describe a thing or to express a concept [11]. *In this work, we consider as term any literal* ($l \in L$) *in a KG.*

Definition 4 (RDFTerm Label). *A term associated with an RDFTerm* φ*, denoted by* $\mathrm{L}(\varphi)$*, is the literal respectively the label of* φ*. Considering the* rdfs:label[6] *as labeling property:*

$$\mathrm{label}(r) := \{l \in L \mid (r, \texttt{rdfs:label}, l) \in K\}$$

$$L(\varphi) := \begin{Bmatrix} \{\varphi\} & \text{if } \varphi \in L, \\ \mathrm{label}(\varphi) & \text{otherwise.} \end{Bmatrix}$$

Although there is no evidence that the previous works were influenced by Hudson's theory, there are models that make use of the KG in order to evaluate the answer [14,20]. Figure 1 shows a set of literals associated with the resources in the KG sample. Each resource contains a set of terms $LR(r)$. This terms are called *Resource-Associated Terms* and are defined as follows:

Definition 5 (Resource-Associated Terms). *The set of terms associated with a resource* r *denoted by* $LR(r)$ *is the union of all literals as well as labels of each property and object in the triples in which* r *is the subject.*

$$LR(r) := \{l \in L \mid \exists (r, p, o) \in K : \exists \varphi \in \{p, o\} : l = L(\varphi)\}$$

Example 1 (Resource-Associated Terms). Considering the KG depicted in Fig. 1, the triples having the entity e_2 as subject are as follows:

```
1. e2 rdfs:label"Mona Lisa".
2. e2 dbo:artist e1.
```

The associated terms for e_2 are: $LR(e_2) = \{$ `"label"`, `"Mona Lisa"`, `"artist"`, `"Leonardo da Vinci"`$\}$

Definition 6 (Term Network). *A Term Network is a graph whose vertices are labeled with terms.*

A KG can be converted to a TN by visiting all vertices and edges executing the following operations (Fig. 2 shows the TN for Example 1):

1. Labeling edges and non-literal vertices by a copy of their respective labels defined by the labeling property rdfs:label;
2. Converting edges to vertices.

The TN of a KG is connected and its paths can have cycles as well as an arbitrary length. In order to simplify the TN and eliminate its ambiguity, the proposed model works on a simplified version of the TN extracted from a structure called Semantic Connected Component (SCC), defined as follows:

[5] Not to be confused with an RDFTerm.
[6] Other labeling properties may also be used.

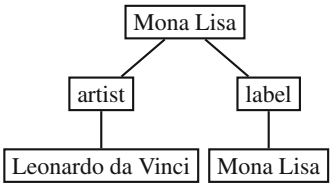

Fig. 2. Representation of a TN extracted from the triples that have e_2 as subject from the KG depicted in Fig. 1.

Definition 7 (Semantic Connected Component). *The Semantic Connected Component (SCC) of an entity e in an RDF graph G under a consequence relation $\models$ is defined as $SCC_{G,\models}(e) := \{(e,p,o) \mid G \models \{(e,p,o)\}\} \cup \{(p, rdfs{:}label, l) \in G\} \cup \{(o, rdfs{:}label, l) \in G\}\}$. If the graph and consequence relation is clear from the context, we use the shorter notation $SCC(e)$. Within this paper, we use the RDFS entailment consequence relation as defined in its specification*[7].

Example 2 (Semantic Connected Component). For instance, by RDFS entailment, the entity dbr:Australia is a dbo:PopulatedPlace. The inference is due to dbr:Australia being typed as dbo:Country which is a subclass of dbo:PopulatedPlace. Considering the running example, the SCC of the entity e_2 is $SCC(e_2) = (\{e_2, e_1, \texttt{"Mona Lisa"}\}, \{p_5, p_4\})$.

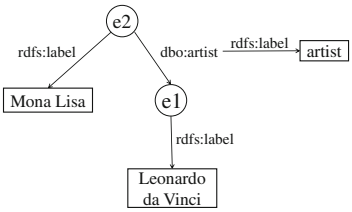

Fig. 3. Representation of the SCC of the entity e_2 extracted from the KG depicted in Fig. 1.

The structure used for ranking is called Semantic Unit (SU). The SU is a tree, where the nodes starting from its root node are labeled with tokens and have only one child. Tokens are sub-strings extracted from another string, they are formally defined as follows.

Definition 8 (Token). *A token $t \in \mathcal{T}$ is the result from a tokenizing function $\mathcal{T} : \Sigma^* \rightarrow \Sigma^{**}$, which converts a string to a set of tokens.*

[7] http://www.w3.org/TR/rdf-mt/.

The root node sub-trees of the SU form a set of paths starting from the resource to which the SCC is associated, see Fig. 4. The SU is defined as follows:

Definition 9 (Semantic Unit (SU)). *The Semantic Unit is a tree where:*

- *The root node is an entity;*
- *All vertices in the root node sub-trees only have one child, and;*
- *Vertices in the root node sub-trees are labeled with tokens.*

Example 3 (Semantic Unit (SU)). Considering the running example, the SU of the entity e_2 is $SU(e_2) = (\{\ e_2,v_1,v_2,v_3,v_4,v_5,v_6,v_7\},\ \{(e_2, v_1),\ (e_2, v_5),\ (v_1, v_2),\ (v_2, v_3),\ (v_3, v_4),\ (v_5, v_6),\ (v_6, v_7)\})$[8] and is depicted in Fig. 4.

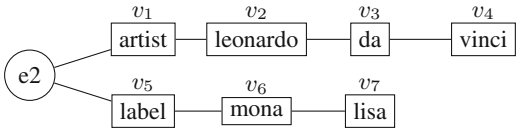

Fig. 4. Representation of the SU of the entity e_2 extracted from the KG depicted in Fig. 1.

An SCC can be converted into an SU as follows:

1. Converting the sub-trees starting from the root node of the SCC into TN;
2. Converting the literal vertices to a graph where there is an edge starting from each token to its subsequent one, defined as follows:

$$\mathcal{G}(l) := (\mathcal{T}(l), \mathcal{D}(l))$$
$$\mathcal{D}(l) := \{(t_1, t_2) \in \mathcal{T}(l) \mid \exists i \in \mathbb{N} : (\pi_i(\mathcal{T}(l)) = t_1) \wedge (\pi_{i+1}(\mathcal{T}(l)) = t_2)\}$$

Example 4 (Literal to graph). Converting the term `"mona lisa"` to a graph.

$$\mathcal{G}(\texttt{"mona lisa"}) = (\{\texttt{"mona"}, \texttt{"lisa"}\}, \{(\texttt{"mona"}, \texttt{"lisa"})\})$$

In the following sections, we start by describing how we retrieve SU in the KG using the query terms. Later, we discuss how we can efficiently rank it.

4.1 Retrieving

The idea is to perform the selection of SUs which have a term in intersection with the query terms. For instance, one possible solution for {`"mona"`, `"lisa"`, `"artist"`} is the co-occurrence of all terms in a SU. The next possible solution is the co-occurrence of two of the three terms and so on. Thus, it is necessary to check for the existence of the query terms in different paths. For example, one SU may contain the token `"artist"` and another with the tokens (`"mona"`, `"lisa"`), see Example 5.

[8] The output of the tokenizer used in this example are lowercase lexemes from a literal.

Example 5 (Retrieving "Mona Lisa artist"). In the KG in Fig. 1, the SCC containing the answer for the query {"mona", "lisa", "artist"} is SCC(e_2) and can be retrieved by a simple lookup with a SPARQL query.

Query and Resource Labels Analysis. Information retrieval systems for RDF are commonly designed to support full or keyword NL queries. However, converting keywords to full queries is a more challenging task. The `*path` query approach is designed to deal with keyword or full queries by converting the latter into keyword queries. The process of conversion of a NL input query to a tuple of keywords consists of applying known techniques, in order: (1) lowercase and (2) lemmatization. In order to increase the number of matched SUs, the same analysis is applied to the SU labels.

After extracting the SUs, the SCC of the SU's entity is used for ranking.

4.2 Ranking

Document retrieval approaches are not suitable for RDF because the most important feature of RDF is not the terms, but the relation of the concepts underlying its graph structure. The challenge of adapting the ranking method is measuring the relatedness between the resources in the target KG and the input query terms. As a query rarely exactly matches the resource associated terms, both are first converted into tokens. Thereafter, the proposed ranking assumes that the probability of a resource being part of an answer correlates with the number of matched tokens between the query and the resource associated terms. For instance, a query containing *birth date* should be more related to the property dbo:birthDate than to the property dbo:deathDate or dbpprop:date. The strength is measured by the number of query tokens matching with the resource tokens.

Definition 10 (Resource Matching). *A resource matching is a function* $MT : \mathcal{T} \rightarrow 2^{\mathcal{R}}$ *that maps query tokens* $\mathcal{T} = \{t_1, t_2, t_3 ... t_n\}$ *to resources, formally defined by* MT(t)*, where* δ *is a string dissimilarity function and* $\theta \in [0, 1] \subset \mathbb{R}$*:*

$$MT(t) := \{r \in \mathcal{R} \mid \exists t' \in \mathcal{T}(LR(r)) : \delta(t, t') < \theta\}$$

Example 6 (Resource Matching). Let $\mathcal{T}(q)$ = {"mona", "lisa", "artist"}. According to Fig. 1, the tokens are mapped to: MT("mona")={$e2$}, MT("lisa")= {$e2$}, MT("artist")={p_4}.

As the knowledge base is a graph, the resources and literal values are connected by paths formed by edges and vertices, see Fig. 3.

Example 7 (Path). In the SCC shown in Fig. 3, there are two paths starting from the entity e_2 as follows: γ_1= ((e_2, "Mona Lisa")) and γ_2 = ((e_2, e_1)).

Furthermore, resources belonging to a path between one resource to another are labeled (e.g. rdfs:label). Therefore, it is possible to explore the terms associated to the entity's paths to determine its relevance.

Definition 11 (Path terms). *Path terms are the set of all literals in the path* γ*, defined as follows:*

$$LP(\gamma) := \{l \mid \exists \varphi \in \gamma : l \in L(\varphi)\}$$

Example 8 (Path terms). For Example 7, the set of associated terms for the two given paths are as follows: $LP(\gamma_1)$= {"label", "Mona Lisa"} and $LP(\gamma_2)$ = {"artist", "Leonardo da Vinci"}.

Thus, the relevance score of an entity depends on the number of matched terms in its associated paths. The higher the number of matched terms, the higher the relevance of the entity. Furthermore, if a term matches multiple paths of an entity, it is only attributed to the path with the highest number of matched terms. The relevance score of an entity is the sum of all individual path scores; it is measured by the *Semantic Weight Model (SWM)*, which is formally defined as follows.

Definition 12 (Semantic Weight Model (SWM)). *Each* token t *in* $\mathcal{T}(q)$ *is first mapped to the* paths *of the* SCC S*. The set of matched tokens from a path* γ *is returned by the function* $TP(\gamma, q)$*. A* path match *of an* SCC S *is evaluated by the function* $\mathrm{MTP}(\gamma, q, S)$ *using a path weighting function* $\mathrm{w} : D^+ \rightarrow R$*.*

$$TP(\gamma, q) := \{t \in \mathcal{T}(LP(\gamma)) \mid \exists t' \in \mathcal{T}(q) : \delta(t, t') < \theta\}$$
$$\mathrm{MTP}(\gamma, q, S) := \{t \in TP(\gamma, q) \mid \forall \gamma' \in D(S)^+ : \mathrm{w}(\gamma)|TP(\gamma, q)| \geq \mathrm{w}(\gamma')|TP(\gamma', q)|\}$$

The final score of an SCC S *is a sum of its* n path-scores *and is measured by the function* $\mathrm{score}(S)$*, as follows:*

$$\mathrm{score}(S) = \sum_{\gamma \in D(S)^+} \begin{cases} \mathrm{w}(\gamma)|TP(\gamma, q)| & \text{if } \mathrm{MTP}(\gamma, q, S) \neq \emptyset, \\ 0 & \text{otherwise.} \end{cases}$$

In case there are terms matching multiple paths and the paths have equal number of matched terms and equal score, only one of the path scores is added to the SCC *score.*

The SWM assigns different weights based on the RDF properties on the path. This means that the weight of a term in a path is determined by the type of the properties (label, is-a relation, other) on that path and it acts as a tiebreaker for the paths with equal number of tokens. The weight hierarchy of paths is constructed to allow the exploration of the KG by querying entities by type, label, predicates and objects. Since terms extracted from resources can have overlaps, there is a need for providing a disambiguation method.

Weighing. Following we start explaining the rationality behind the defined weights, later we use examples to better illustrate it.

Is-a Relation. The problem is that tokens can exist in different paths of an SCC. Thereafter, a token in an is-a relation property can also exists in other properties. However, a property as an entity label references the entity itself while an is-a relation references classes of entities. In this case, if a query intends to select a specific class of entities, other entities can be retrieved by mistake. Thus, it is important to provide an efficient method to disambiguate between classes and entities. To alleviate this problem, the weight of the paths containing an is-a relation property are set higher than other paths. Thereafter, the selection of a specific entity can be done by building a more precise query. The reason is that beside the entity's label, other properties can be used to disambiguate. For instance, in the case of a class and an entity have the same label, the user can use other entity property's term. Therefore, the highest weight is assigned to paths with an is-a relation property γ_t—i.e. the paths containing rdf:type.

Entity Label. The second highest weight is assigned to labeling property paths γ_l—i.e. the paths containing the rdfs:label property—and those are assigned higher values than other property paths γ_o. Entities can be referenced multiple times in a KG, but when a query contains an entity label, it is more likely that it is looking for the entity than for its references—an object instance. Therefore, to prevent entities with references to be higher ranked than the entity itself, the weight of the path with an labeling property is set higher than a path with another property.

Despite the different weights, we still want a higher number of matched tokens to score higher in practical cases, i.e. $n+1$ matched tokens should score higher than n matched tokens for reasonably low n:

$$\begin{aligned}(n+1)\,\mathrm{w}(\gamma_t) > (n+1)\,\mathrm{w}(\gamma_l) > \\ (n+1)\,\mathrm{w}(\gamma_o) > n\,\mathrm{w}(\gamma_t) > \\ n\,\mathrm{w}(\gamma_l) > n\,\mathrm{w}(\gamma_o)\end{aligned} \tag{1}$$

Following, the model is explained using examples.

Case 1: Querying by Entity label. For the query "Rio de Janeiro", the SWM should consider the DBpedia entity dbpedia:Rio_de_Janeiro as the best answer although the DBpedia entity dbpedia:Tom_Jobim has the DBpedia property dbpprop:birthPlace referencing the entity dbpedia:Rio_de_Janeiro. For the term "The" in a query, the model will consider as a possible answer the entities dbpedia:The_Simpsons and dbpedia:The_Beatles rather than the DBpedia property dbpprop:The_GIP.

Case 2: Querying by Is-a Relation. Considering the query "place", the implemented *SWM* will prefer the data type dbo:Place instead of the property dbo:Place.

Case 3: Querying by Another Properties. Let us consider the case that the query is "birth place" rather than "place" as in the previous example. As the number

of matching terms in the property dbo:birthPlace is higher than for the data type dbo:Place, consequently the weight of dbo:birthPlace will be higher than the *data type*.

5 Experimental Evaluation

We evaluate the performance of `*path` in comparison to the state-of-the-art SemS system as well as QA in terms of Precision, Recall and F-measure. To the best of our knowledge is the first time that the precision of both approaches are measured in the same benchmark.

Benchmark. Several benchmarks can be used to measure the precision of our approach, including benchmarks from the initiatives *SemSearch* [4][9] and *QA Over Linked Data (QALD)*[10]. *SemSearch* is based on user queries extracted from the YAHOO! search log, with an average distribution of 2.2 words per-query. *QALD* provides both QA and keyword search benchmarks for RDF data that aim to evaluate the extrinsic behavior of systems. The QALD benchmarks are the most suitable for our evaluation due to the wide type of queries they contain and also because it makes use of *DBpedia*, a very large and diverse dataset. In this work, we use openQA framework [9] over the newest version of the QALD benchmark compatible with the framework—QALD version 4 (QALD-4) benchmark [17]. The proposed approach was compared with respect to the performance of $\text{Glimmer}_{\text{Y!}}$ because it is the best performing SemS system and it is open-source, which allows to evaluates its performance.

Results. Table 1 shows the performance of `*path` in comparison to $\text{Glimmer}_{\text{Y!}}$, the state-of-the-art SemS system [1], and all participating QA systems in the multilingual challenge of the *QALD-4* benchmark.

Discussion. The proposed approach is faster than the best SemS participating in SemSearch'10. The main reason is that $\text{Glimmer}_{\text{Y!}}$ build an an index without reasoning which imposes constraints on the precision (Table 1). The index without reasoning is a core limitation of $\text{Glimmer}_{\text{Y!}}$, since the user cannot query by using terms from properties as well as from entity objects. For instance, in *Case* 1 in Sect. 4.2, $\text{Glimmer}_{\text{Y!}}$ fails to retrieve dbpedia:Tom_Jobim because the terms of the entity dbpedia:Rio_de_Janeiro belonging to the property dbpprop:birthPlace are not indexed. The same occurs for the data type in *Case* 2 where the type is also given by a non-literal object. However, the `F-measure` of `*path` decreases sensitively (0.42) in comparison with the best performing QA system in QALD-4. The drawback is due to `*path` does not target the treatment of complex queries—i.e., queries that require the use of aggregations, restrictions as well as solution modifiers to be answered.

[9] http://km.aifb.kit.edu/ws/semsearch10/.
[10] http://greententacle.techfak.uni-bielefeld.de/~cunger/qald/.

Table 1. *Precision* (**P**), *Recall* (**R**) and *F-measure* ($\mathbf{F}_1$) achieved by different SemS and QA systems in QALD-4 *Multilingual Challenge*. The systems are Glimmer$_{Y!}$, ***path**, SINA, TBSL and all QALD-4 participating systems.

System	P	R	F_1	Approach
Xser	0.71	0.72	0.72	QA
gAnswer	0.37	0.37	0.37	QA
CASIA	0.40	0.32	0.36	QA
***path**	0.30	0.30	0.30	SemS
Intui3	0.25	0.23	0.24	QA
ISOFT	0.26	0.21	0.23	QA
SINA	0.15	0.15	0.15	QA
RO FII	0.12	0.12	0.12	QA
TBSL	0.10	0.10	0.10	QA
Glimmer$_{Y!}$	0.07	0.07	0.07	SemS

6 Conclusion, Limitations and Future Work

We have presented a novel ranking method for SemS over KGs. The results of an experimental study show a significant improvement in comparison to the state-of-the-art SemS. Furthermore, the approach achieves comparable precision when compared with QA systems.

There are a few challenges not addressed in the current implementation as complex queries [5]. In future work, we plan to extend the precision of this approach by addressing the mentioned challenges. Furthermore, we plan to investigate indexing techniques. We see this work as the first step of a larger research agenda for SemS over Linked Data.

Acknowledgements. This work was supported by a grant from the EU H2020 Framework Programme provided for the projects Big Data Europe (GA no. 644564), HOBBIT (GA no. 688227), and CNPq under the program Ciências Sem Fronteiras.

References

1. Blanco, R., Mika, P., Vigna, S.: Effective and efficient entity search in RDF data. In: Aroyo, L., Welty, C., Alani, H., Taylor, J., Bernstein, A., Kagal, L., Noy, N., Blomqvist, E. (eds.) ISWC 2011. LNCS, vol. 7031, pp. 83–97. Springer, Heidelberg (2011). doi:10.1007/978-3-642-25073-6_6
2. Cheng, G., Qu, Y.: Searching linked objects with Falcons: approach, implementation and evaluation. Int. J. Semant. Web Inf. Syst. **5**(3), 49–70 (2009)
3. Ding, L., Finin, T., Joshi, A., Pan, R., Cost, R.S., Peng, Y., Reddivari, P., Doshi, V.C., Sachs, J.: Swoogle: a search and metadata engine for the semantic web. In: Proceedings of the Thirteenth ACM Conference on Information and Knowledge Management (CIKM), pp. 652–659. ACM (2004)

4. Halpin, H., Herzig, D.M., Mika, P., Blanco, R., Pound, J., Thompson, H.S., Tran, D.T.: Evaluating ad-hoc object retrieval. In: Proceedings of the International Workshop on Evaluation of Semantic Technologies (IWEST 2010), 9th International Semantic Web Conference (ISWC 2010), Shanghai, PR China, November 2010
5. Höffner, K., Walter, S., Marx, E., Usbeck, R., Lehmann, J., Ngonga Ngomo, A.C.: Survey on challenges of Question Answering in the Semantic Web. Submitted to the Semant. Web J. (2016). http://www.semantic-web-journal.net/content/survey-challenges-question-answering-semantic-web
6. Hudson, R.A.: Language Networks: The New Word Grammar. Oxford Linguistics, Oxford University Press, Oxford (2007)
7. Luhn, H.P.: A statistical approach to mechanized encoding and searching of literary information. IBM J. Res. Dev. **1**(4), 309–317 (1957)
8. Mangold, C.: A survey and classification of semantic search approaches. Int. J. Metadata Semant. Ontol. **2**(1), 23–34 (2007)
9. Marx, E., Usbeck, R., Ngonga Ngomo, A.C., Höffner, K., Lehmann, J., Auer, S.: Towards an open question answering architecture. In: SEMANTiCS (2014)
10. Oren, E., Delbru, R., Catasta, M., Cyganiak, R., Stenzhorn, H., Tummarello, G.: Sindice.com: a document-oriented lookup index for open linked data. IJMSO **3**(1), 37–52 (2008)
11. Pearsall, J., Hanks, P., Soanes, C., Stevenson, A. (eds.): Oxford Dictionary of English (Kindle Edition) (2010)
12. Reisburg, D.: Cognition: Exploring the Science of the Mind. Norton, New York (1997)
13. de Saussure, F.: Course in General Linguistics. McGraw-Hill, New York (1959). (Translated by Wade Baskin)
14. Shekarpour, S., Marx, E., Ngomo, A.C.N., Auer, S.: SINA: semantic interpretation of user queries for question answering on interlinked data. J. Web Semant. **30**, 39–51 (2015)
15. Sparck Jones, K.: A statistical interpretation of term specificity and its application in retrieval. J. Documentation **28**(1), 11–21 (1972)
16. Tummarello, G., Cyganiak, R., Catasta, M., Danielczyk, S., Delbru, R., Decker, S.: Sig.ma: live views on the web of data. J. Web Semant. **8**(4), 355–364 (2010)
17. Unger, C., Forascu, C., Lopez, V., Ngomo, A.C.N., Cabrio, E., Cimiano, P., Walter, S.: Question answering over linked data (QALD-4). In: Working Notes for CLEF 2014 Conference (2014)
18. Virgilio, R., Maccioni, A.: Distributed keyword search over RDF via MapReduce. In: Presutti, V., d'Amato, C., Gandon, F., d'Aquin, M., Staab, S., Tordai, A. (eds.) ESWC 2014. LNCS, vol. 8465, pp. 208–223. Springer, Heidelberg (2014). doi:10.1007/978-3-319-07443-6_15
19. Wang, H., Liu, Q., Penin, T., Fu, L., Zhang, L., Tran, T., Yu, Y., Pan, Y.: Semplore: a scalable IR approach to search the web of data. J. Web Semant. **7**(3), 177 (2009)
20. Zhang, L., Liu, Q.L., Zhang, J., Wang, H.F., Pan, Y., Yu, Y.: Semplore: an IR approach to scalable hybrid query of semantic web data. In: Aberer, K., et al. (eds.) ASWC/ISWC 2007. LNCS, vol. 4825, pp. 652–665. Springer, Heidelberg (2007). doi:10.1007/978-3-540-76298-0_47