Pattern-Based Keyword Search on RDF Data[⋆]

Hanane Ouksili[1,2], Zoubida Kedad[1], Stéphane Lopes[1], and Sylvaine Nugier[2]

[1] DAVID lab., Univ. Versailles St Quentin, Versailles France
`firstname.lastname@uvsq.fr`
[2] EDF R&D, Departement STEP, Chatou, France
`firstname.lastname@edf.fr`

Abstract. An increasing number of RDF datasets are available on the Web. Querying RDF data requires the knowledge of a query language such as SPARQL; it also requires some information describing the content of these datasets. The goal of our work is to facilitate the interrogation of RDF datasets, and we present an approach for enabling users to search in RDF data using keywords. We propose the notion of pattern to include some external knowledge during the search process which increases the quality of the results.

Keywords: RDF graph, Keyword search, Patterns, Domain knowledge.

1 Introduction

A huge volume of RDF datasets is available on the Web, enabling the design of novel intelligent applications. In order to query these datasets, users must have information about their schema to target the relevant resources and properties. They must also be familiar with a formal query language such as SPARQL. An alternative approach to query RDF data is keyword search which is a straightforward and intuitive way of querying datasets.

We can distinguish between two categories of approaches for keyword search in RDF data. The first one uses information retrieval techniques on documents which are previously defined. For example, a document can be defined as a triple [1]. Documents are indexed, then keywords are mapped into the graph to identify relevant documents. Finally, the result is returned, consisting in a ranked list of documents. These works do not try to combine parts of the graph which correspond to different keywords. In the second category, the structure of the graph is used to construct the results [2, 3]. Keyword-to-graph mapping functions are used to identify a set of elements which contains the query keywords. The ranked list of subgraphs is then returned as a result. In these works, only the structure of the graph is considered during the construction of the result, unlike our approach, which integrates external knowledge formalized as patterns.

In this paper, we introduce a novel keyword search approach which returns graphs as a result to a keyword query (see Figure 1). The key contribution of this paper is the use of external knowledge, expressed as patterns, during the extraction of the relevant graph fragments, the construction of the result and the ranking step. Experiments show that our approach delivers high-quality search results.

[⋆] This work is supported by EDF and french ANR project CAIR.

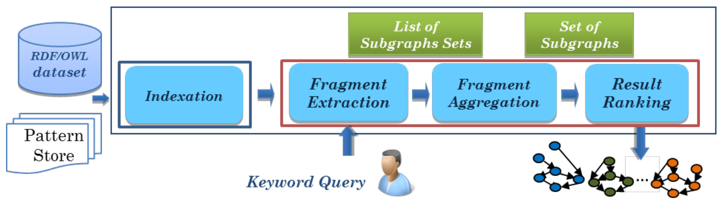

Fig. 1. Keyword search process

2 Expressing Knowledge with Patterns

According to the knowledge available for a specific domain, or to the user's point of view, some equivalence relations can be defined between properties and paths in the dataset. In our approach, they are expressed through patterns.

• A *pattern* represents an equivalence between one property expression and one path expression. It is defined as a pair $[exp,\ exP]$ where $exp =\ (X, p, Y)$ is a property expression, $exP = (X, P, Y)$ is a path expression, p is a property, P is a SPARQL 1.1 path expression[3], and X and Y are either resources or variables.

For example, consider the following pattern: $[(X,\ swrc{:}isAbout,\ Y),\ (X,\ swrc{:}isAbout/(owl{:}sameAs|\hat{}owl{:}sameAs)^{+},\ Y)]$. This pattern means that the property *swrc:isAbout* is equivalent to a path composed of a property *swrc:isAbout* followed by a sequence of *owl:sameAs* properties. Its evaluation on the RDF graph of Figure 2 extracts the value *D.B.* for the property *swrc:isAbout* of the resource *Art1*, *Database* is therefore considered as value for this property.

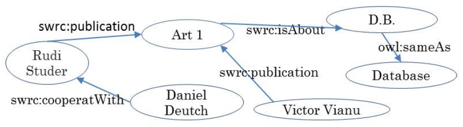

Fig. 2. Subgraph from SWRC

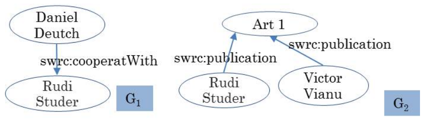

Fig. 3. Final Results

3 Extracting Relevant Fragments

We use an inverted index to extract the relevant fragments. To this end, we consider a representative keyword from the local name of URI for resources and properties as document content; for example, a document is created for the property *swrc:hasAuthor*, containing the word *Author* (it can be generated using Information Extraction techniques). For literals, we consider their content as a document.

Let G be an RDF graph and $Q = \{k_1, ..., k_n\}$ a keyword query. The system first matches the keywords and the graph elements (i.e. classes, instances, properties and literals) using the index, a mapping function and some standard transformation functions such as abbreviation and synonym. This step returns a set of lists El=$\{El_1, ..., EL_n\}$ where each El_i is a list of elements el_{ij} that matches the keyword k_i. We refer to these as *keyword elements*. Patterns are

[3] http://www.w3.org/TR/sparql11-property-paths/

then used to extract, for each keyword element el_i, subgraphs that are relevant to the query. We denote these subgraphs and keyword element as relevant fragments. If no pattern is applicable, the relevant fragment is reduced to el_i.

We use two kinds of patterns: generic and domain specific. Generic patterns are valid for any dataset as they are related to RDF, RDFS or OWL vocabularies. For example, the pattern $[(X,\ pat{:}sameResult,\ Y),\ (X,\ (owl{:}sameAs\ |\hat{}owl{:}sameAs)^{+},\ Y)]$ is defined to consider X as a relevant fragment if it is related to one keyword element Y with a path composed of a sequence of *owl:sameAs* properties. This is true for any dataset. The evaluation of this pattern in the graph of Figure 2 allows to consider that the resource *database* is relevant when the resource *D.B.* is extracted as a keyword element.

Similarly, domain specific patterns can be defined. For example, the property *swrc:cooperateWith* between two researchers in the ontology SWRC[4] states that they have already collaborated. However, this property is not always defined. If we know that they have authored the same paper, we can infer that they have collaborated, even if the property *swrc:cooperateWith* is missing. We formalize this by the following pattern:

$[(X,\ swrc{:}cooperateWith,\ Y),\ (X,\ swrc{:}publication/\hat{}swrc{:}publication,\ Y)]$. This pattern is used during the *fragment extraction* module by considering the subgraph corresponding to the path ($swrc{:}publica\ tion/\hat{}swrc\ {:}publication$) as a relevant fragment if the property *swrc:cooperateWith* is a keyword element. Consider that *"Studer Cooperator"* is a query issued to find the collaborators of the researcher named *Studer* in the graph of Figure 2. Let us consider the following two extracted keyword elements: the node *Rudi Studer* and the property *swrc:cooperateWith.* Without using patterns, the only result will be the subgraph G_1 of Figure 3 representing *Daniel Deutch*, a collaborator of *Rudi Studer*, both linked by the *swrc:cooperateWith* property. However, *Victor Vianu* is also a collaborator of *Rudi Studer* as they published a paper together (*Art1*). Using the pattern defined above, our approach will also return the subgraph G_2 of Figure 3 as result.

Note that if the keyword element is a property as in our example, our approach will not extract all occurrences of the property in the graph but only the ones for which either the object or the subject is a relevant fragment. Otherwise, the result might include irrelevant answers. In our example, the system will return only cooperators of *Studer* and not cooperators of others researchers.

4 Result Construction and Ranking

The result for a keyword query Q is a list of subgraphs. The goal of the aggregation is to merge relevant fragments to form the set of results. Each result is a connected minimal subgraph which contains for each keyword k_i one corresponding relevant fragment. We use a Cartesian product between the different sets of relevant fragments to construct the combinations. Finally, for each combination, we perform a bidirectional expansion search strategy to join different relevant fragments and construct the result. We introduce the defined patterns

[4] http://ontoware.org/swrc/

to calculate the path between resources. If a pattern indicates that one path is equivalent to one property, the distance is reduced to 1.

Since several elements may contain the same keyword, several subgraphs are returned to the user. Hence, we define a ranking function to evaluate the relevance degree of each subgraph. Our approach combines two criteria: (i) *Compactness Relevance* where compact answers are preferred. When the size of the subgraph is smaller, the compactness is higher. Patterns are taken into account to calculate the size. (ii) *Matching Relevance* is calculated using standard IR approaches. For our approach, we have used the TF-IDF function.

5 Evaluations

We have implemented a prototype which provides an interactive interface to keyword search and allows users to express some external knowledge, which is formalized by our system using patterns. We have used AIFB dataset[5] and we have compared our approach to two baseline algorithms which contain the same steps but do not use patterns; the first one considers both node and edge content without any condition on their extraction and the second one considers node content only.

We have used 10 queries that are randomly constructed (with 2 to 5 keywords) and asked four users to assess the top-k results of each query using the three algorithms on a 3-levels scale: 3 (perfectly relevant), 1 (relevant) and 0 (irrelevant). We use the two metrics NDCG@k and precision@k. To calculate the precision@k, we assume that the scores 3 or 1 correspond to a relevant result and 0 to an irrelevant one. Table 1 reports average Precision@k and NDCG@k values over the 10 queries with k=5, 10 and 20. Our approach significantly outperforms the two other algorithms in terms of NDCG and precision values at all levels. Furthermore, since NDCG includes the ranking position of the results, the ranking algorithm of our approach is better because it includes patterns during the ranking step.

Table 1. Evaluation results

Approach	Our Approach			Approach (i)			Approach (ii)		
k	5	10	20	5	10	20	5	10	20
Precision@k	0.99	0.98	0.97	0.73	0.73	0.69	0.66	0.66	0.66
NDCG@k	0.92	0.90	0.90	0.64	0.63	0.58	0.47	0.46	0.46

References

1. S. Elbassuoni and R. Blanco. Keyword search over rdf graphs. In *CIKM*, pages 237–242, 2011.
2. H. F. Wang, K. Zhang, Q. L. Liu, T. Tran, and Y. Yu. Q2Semantic: a lightweight keyword interface to semantic search. In *ESWC*, pages 584–598, 2008.
3. S. Yang, Y. Wu, H. Sun, and X. Yan. Schemaless and structureless graph querying. *VLDB Endow.*, pages 565–576, 2014.

[5] http://www.aifb.kit.edu/web/Hauptseite/en