





Keyword Search over RDF Datasets

(Extended Abstract)

Marco A. Casanova  

Department of Informatics, PUC-Rio, Rio de Janeiro, RJ, Brazil
casanova@inf.puc-rio.br

Abstract. This extended abstract first introduces the problem of keyword search over RDF datasets. Then, it expands the discussion to cover the question of serendipitous search as a strategy to diversify answers. Finally, it briefly presents the entity relatedness problem, which refers to the problem of exploring an RDF dataset to discover and understand how two entities are connected.

Keywords: Keyword search · Serendipity · Entity relatedness · RDF · SPARQL

1 Introduction

Keyword search is typically associated with information retrieval systems, especially those designed for the Web. The user just specifies a few terms, called *keywords*, and the system must retrieve the documents, such as Web pages, that best match the list of keywords. Keyword search over relational databases, as well as over RDF datasets, has also been studied for some time. In particular, the adoption of RDF as the underlying data model adds flexibility and imposes no strict distinction between data and metadata, that is, a keyword may match the name or description of a class or of a property in the same way that it may match a data value. An RDF management system may also offer an inference layer so that one may expand the stored RDF data with derived data in ways that surpass (relational) views. Thus, a keyword may match derived data as much as stored data. Lastly, an RDF dataset is equivalent to a labeled graph, called an RDF graph, which allows the use of graph concepts and algorithms for keyword search.

Keyword search over RDF datasets imposes distinct challenges when compared with traditional keyword search. Indeed, in the latter case, an answer for a keyword query is a document that matches as many keywords as possible, and the various answers (documents) are ranked using well-known measures. By contrast, in the former case, keywords select nodes and edges from an RDF graph, and it is up to the system to find a connected subgraph of the RDF graph that covers these nodes and edges to create an answer for the keyword query. Since there might be more than one such subgraph, the system must rank them according to some reasonable measure.

This extended abstract first discusses the problem of keyword search for RDF datasets. Then, it expands the discussion to serendipitous search as a strategy to diversify answers. Finally, it briefly presents the entity relatedness problem, which refers to the problem of exploring an RDF graph to discover how two entities are connected.

2 Classic Keyword Search Over RDF Datasets

An *Internationalized Resource Identifier* (IRI) is a global identifier that denotes a resource. A *blank node* identifier is a local identifier. RDF [3] describes data as triples of the form (s,p,o) , where s is the *subject*, p is the *predicate* (or *property*) and o is the *object* of the triple. The subject of a triple is an IRI or a blank node, the predicate is an IRI, and the object is an IRI, a blank node or a *literal*. An RDF dataset is a set T of RDF triples and is equivalent to a labeled graph G_T whose nodes are the RDF terms that occur as subject or object of the triples in T and there is an edge (s,o) in G_T labeled with p iff $(s,p,o) \in T$. We will use the terms RDF dataset and RDF graph interchangeably.

RDF *Schema* [2] is a specific vocabulary that permits defining classes and properties, and hierarchies thereof, among other constructs. It should be noted that an RDF dataset may not have an RDF schema. SPARQL 1.1 [6] is a query language to access RDF datasets. The `WHERE` clause of a SPARQL query is a set of *triple patterns*, defined like RDF triples, except that the subject, predicate or object can be a variable.

A *keyword query* is simply a set of literals, or *keywords*, $K = \{K_1, \dots, K_n\}$. A keyword K_i *matches* a triple (s,p,o) iff o is a literal and K_i and o are considered similar (according to some criterion). An *answer* for K over an RDF dataset T is a subset A of T such that there are triples in A that match some of the keywords in K . Note that this notion of answer allows keywords to remain unmatched and permits the RDF graph induced by A to be disconnected. However, answers that induce minimal, connected graphs that match as many keywords as possible should be preferred. Also, note that a keyword may match the label or the description of a class or property, which alters the interpretation of a keyword query. For example, if C is a class with a property `rdfs:label` whose value is the literal “city”, then the keyword query $K = \{city, Princeton\}$ can be interpreted as requesting an instance c of class C such that c has a property whose value matches “Princeton”. The problem of keyword search over RDF datasets is then defined as: “Given an RDF dataset T and a keyword query K , find a *minimally connected answer* for K over T that matches as many keywords as possible”.

Given a keyword query K , an RDF keyword query processing tool first matches the keywords in K with literals that occur in the RDF graph and then either directly crawls the RDF graph to find answers for K or compiles a SPARQL query that returns answers for K . Variations of this basic process may adopt an ontology to expand the keyword matching process, and may introduce ranking strategies to order the keyword matches, to improve the crawling or compilation processes, and to order the answers [11].

The tools also differ on the strategy adopted to compile the SPARQL query. *Schema-based* tools [5] explore the RDF schema to compile a SPARQL query with a minimal set of join clauses – and this is a key idea. In fact, the tool described in [9] supports keyword query processing for both relational databases and RDF datasets with schemas. To circumvent the lack of an RDF schema, *graph-based* tools may compile a SPARQL query based on elementary query graph building blocks, such as entity/class nodes and predicate edges, or graph summarizations. We also find a strategy [10] that estimates set similarity measures using KMV-synopses [1], which in turn drive the SPARQL query compilation process, and a strategy based on tensor calculus.

3 Beyond the Basics: Serendipitous Search

Serendipity is defined as “the art of making an unsought finding”. In a seminal work, Van Anandel [12] defined a list of seventeen serendipity patterns, each one representing a different form of serendipity. The problem of *RDF serendipitous search* can then be intuitively defined as: “Given an RDF dataset T and a query Q , find additional answers related to the original answers for Q by some serendipity pattern”.

A strategy to incorporate serendipity into query processing would then be to mimic Van Anandel’s patterns. This strategy was implemented in [4] for four patterns: *analogy*, *surprising observation*, *disturbance*, and *inversion*. To capture the first two patterns, the process explores the answers for a query to invoke secondary queries with the recently acquired data. To capture the disturbance pattern, the process changes the order of the answer list to expose items that the user would normally neglect. To capture the inversion pattern, the process also formulates alternative queries.

When combined with keyword search, which allows considerable latitude in constructing answers, serendipitous search may produce interesting results that enrich the user’s experience. For example, when processing the keyword query $\{Einstein, Gödel, Princeton\}$, the system may return that Einstein and Gödel were neighbors at Princeton, they died in that city and worked at the Institute for Advanced Study (IAS) at Princeton University (which are the expected answers). But the system may expand these answers to include that Gödel won the first Einstein Award in 1951, created by IAS to honor Einstein, and that Gödel’s favorite movie was “Snow White” (trivia about the foremost mathematical logician of the twentieth century).

4 An Interesting Special Case: Entity Relatedness

When a keyword query K simply selects two nodes, N_1 and N_2 , of the RDF graph, an answer for K reduces to a path between N_1 and N_2 , called a *relationship path*. The *entity relatedness problem* is then defined as: “Given an RDF graph G_T and two entities, represented by two nodes N_1 and N_2 of G_T , compute the relationship paths that better describe the connectivity between the given entities”. For example, DBpedia has more than 10,000 paths between the entries for Einstein and Gödel, that is, the keyword query $\{Einstein, Gödel\}$ has, in this not infrequent case, the patently unwieldy total of more than 10,000 answers over DBpedia, and this is a problem.

There are two basic approaches to address this problem. First, one may try to abstract out the (large) set of relationship paths into a description meaningful to the users [7], or one may rank the relationship paths in an order that reflects their relevance [8], which raises additional questions. The relevance of a path π may have to do with its coherence, measured by how similar neighboring entities (nodes) in π are, or the relevance may be measured by how informative the labels of the edges are, similarly to information retrieval, or by a combination of both. The work in [8] reports an extensive comparison between different combinations of similarity and path ranking measures.

5 Final Remarks: What Else?

RDF Keyword search is tightly related to the exploration of knowledge bases, as a goal in itself or to complement traditional information retrieval. In this context, immediate challenges include to implement keyword search with sub-second response time for large RDF knowledge bases, and to fully incorporate such technology into mainstream search engines and question-and-answer tools to enhance the overall user experience.

Acknowledgments. This work was partly funded by grants CAPES/88881.134081/2016-01, CNPq/302303/2017-0, and FAPERJ/E-26-202.818/2017. The author gratefully acknowledges Altigran Silva, for his inspiring work, and the contributions to the research reported here of Bernardo Nunes, Luiz André Paes Leme, Antonio Furtado, Grettel García, Yenier Izquierdo, Elisa Menendez, José Herrera, Jerónimo Eichler, and Angelo Neves.

References

1. Beyer, K. et al.: On synopses for distinct-value estimation under multiset operations. In: Proceedings 2007 ACM SIGMOD, Beijing, China, pp. 199–210 (2007)
2. Brickley, D., Guha, R.V. (eds): RDF Schema 1.1. W3C Recommendation, 25 February 2014
3. Cyganiak, R., Wood, D., Lanthaler, M. (eds.): RDF 1.1 Concepts and Abstract Syntax. W3C Recommendation, 25 February 2014
4. Eichler, J.S.A., et al.: Searching linked data with a twist of serendipity. In: Dubois, E., Pohl, K. (eds.) CAiSE 2017. LNCS, vol. 10253, pp. 495–510. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-59536-8_31
5. García, G.M., Izquierdo, Y.T., Menendez, E., Dartayre, F., Casanova, M.A.: RDF keyword-based query technology meets a real-world dataset. In: Proceedings of 20th International Conference on Extending Database Technology, Venice, Italy (2017)
6. Harris, S., Seaborne, A.: SPARQL 1.1 Query Language. W3C Recommendation, 21 March 2013
7. Herrera, J.E.T., Casanova, M.A., Nunes, B.P., Lopes, G.R., Leme, L.A.P.P.: DBpedia profiler tool: profiling the connectivity of entity Pairs in DBpedia. In: Proceedings of Intelligent Exploration of Semantic Data - IESD, A Workshop at ISWC 2016, Kobe, Japan (2016)
8. Herrera, J.E.T., Casanova, M.A., Nunes, B.P., Leme, L.A.P.P., Lopes, G.R.: An entity relatedness test dataset. In: d’Amato, C., et al. (eds.) ISWC 2017. LNCS, vol. 10588, pp. 193–201. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-68204-4_20
9. Izquierdo, Y.T., García, G.M., Menendez, E.S., Casanova, M.A., Dartayre, F., Levy, C.H.: *QUIOW*: a keyword-based query processing tool for RDF datasets and relational databases. In: Hartmann, S., Ma, H., Hameurlain, A., Pernul, G., Wagner, R.R. (eds.) DEXA 2018. LNCS, vol. 11030, pp. 259–269. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-98812-2_22
10. Izquierdo, Y.T., et al.: Keyword Search over Schema-less RDF Datasets by SPARQL Query Compilation (Submitted for publication)
11. Menendez, E.S., Casanova, M.A., Paes Leme, L.A.P., Boughanem, M.: Novel Node Importance Measures to Improve Keyword Search over RDF Graphs. (to appear DEXA 2019)
12. Van Andel, P.: Anatomy of the unsought finding serendipity: origin, history, domains, traditions, appearances, patterns and programmability. *Br. J. Philos. Sci.* **45**(2), 631–648 (1994)