Semantic Search Engine with an Intuitive User Interface

Adam Styperek, Michal Ciesielczyk and Andrzej Szwabe
Poznan University of Technology
Maria Sklodowska-Curie Sq. 5
60-965 Poznan, Poland
{adam.styperek | michal.ciesielczyk | andrzej.szwabe}@put.poznan.pl

ABSTRACT

It is crucial to enable regular users to explore RDF-compliant data bases in an effective way, regardless their knowledge about the SPARQL or the underlying ontology. Natural language querying have been proposed to address this issue. However it has unavoidably lower accuracy, as compared to systems with graph-based querying interfaces, which, in turn, are usually still too difficult for regular users. This paper presents a search system of user interface that is more user-friendly than widely known graph-based solutions.

Categories and Subject Descriptors

H.5.2 [**Information Interfaces and Representation**]: User Interfaces—*Evaluation/methodology*;
H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*Query formulation, Search process*

Keywords

Semantic Web, Search, RDF, SPARQL, DBpedia

1. INTRODUCTION

The idea of Semantic Web has become popular partly thanks to the introduction of domain-generic knowledge bases like DBpedia and YAGO. As languages like SPARQL are difficult for regular users [1, 2], solutions enabling to query data more effectively were proposed [3, 4, 2].

The leading approaches to the problem are represented by systems allowing to formulate queries in the natural language, thus enabling the users to create queries without requiring any knowledge of the SPARQL syntax. As shown in this paper, despite the growing popularity of knowledge bases, the practical value of applications that enable to explore them is still rather limited. Such an observation motivates efforts at developing user interfaces for semantic data stores that are intuitive but, at the same time, do not compromise the expressiveness of queries.

To the best knowledge of the authors, there is no widely-agreed methodology allowing to evaluate semantic search systems with different user interfaces in an objective manner. Therefore, in order to address this issue, some simple measures have been proposed in this paper. The analysis is conducted in terms of correctness of answers to queries. The analysis of graph-based querying interfaces focuses on the average number of elements needed to create queries.

2. RELATED WORK

Interfaces based on the natural language processing (NLP) tend to be more user-friendly than graph-based query interfaces. However, despite their simplicity they are inherently inaccurate [2, 3] and, as shown herein, usually provide less relevant results than graph-based search engines.

The Facebook Graph Search [5] is an NLP component of a social networking service. However, because it basically works only on Facebook data we were not able to objectively compare it with other systems. In our research we were able to test PowerAqua [4] – an NLP-based query interface designed to work in any domain. Additionally, for the purposes of our investigation, we evaluated the Google search engine (both using keyword and natural language queries).

Using a graph-based interface, similarly as using a natural language queries, does not require any knowledge about ontologies or formal languages. Each time a user issues a graph-based query, he/she has to specify a set of triple/fact templates that may be seen as simple sentences containing a subject, predicate and object. Such an approach enables to build appropriately complex and precise queries.

Graph of Relations (GoR) [2] is a graph-based query interface allowing to directly type triple patterns. Although such patterns are simpler than SPARQL queries their syntax is still quite strict and unintuitive. The NAGA [3] is a search engine that operates on the YAGO knowledge base and provides users with a rather sophisticated query builder.

3. PROPOSED SOLUTION

The Semantic Search Engine (SSE) presented in this paper provides a user with a graph query builder allowing to construct query triples. The query triples are in fact RDF triple templates containing resources and literals along with variable nodes. Usually, the more complex a query is, the more query triples it has.

The user of SSE may additionally define the variables' types. This feature is useful when the user wants to create a query which contains more than one query triple in SPARQL, for example, when he/she wants to query for ev-

WWW'14 Companion, April 7–11, 2014, Seoul, Korea.
ACM 978-1-4503-2745-9/14/04.
http://dx.doi.org/10.1145/2567948.2577203 .

Table 1: Systems accuracy comparison results

System	Precision	Recall
`SSE`	**0.961**	**0.906**
`GoR`	0.693	0.672
`YAGO-NAGA`	0.661	0.639
`Google (Natural Language)`	0.620	0.632
`PowerAqua (DBPedia only)`	0.512	0.481
`Google (Keyword)`	0.422	0.425

ery football player that has an actress as a wife. Let X be the set of variables, matching any object or property in a query, $|X| = n$ and $x \in X$. While using SSE he/she may create just a single query triple, as shown in the Figure 1. Although this query contains only one query triple, it represents the following three RDF triples:
x_1 *isA FootballPlayer* . x_2 *isA Actress* . x_1 *hasWife* x_2 .

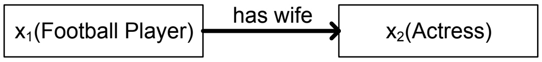

Figure 1: SSE query example.

SSE provides an additional support for variable naming, which is unique among the graph-based systems. Specifically, while looking for information about an unspecified actor, the user may select the suggested variable *?Actor* – with the type field being automatically complemented. As shown in our experiments, this feature reduces the number of element the user has to type in.

The presented system is the only one of the compared graph-based search systems that allows the user to use the functionality of full-text search on indexed literals.

4. EVALUATION

All of the evaluated systems featuring NLP-based and graph-based interfaces have been compared using the QALD-1 [6] queries (formulated in natural language, SPARQL, and in the form of keywords) and the DBPedia knowledge base. As a search result is a list containing results that are relevant or irrelevant to the query, it is possible to equivalently measure their accuracy. In our experiments we have used only these questions that may be represented in all of the evaluated search systems.

We considered a query result as relevant only when it was equal to one of the QALD answers. To resolve a problem of how many search results should be taken into consideration, we calculate the precision only for the first n results, where n is equal to the number of correct answers.

An interface may be considered more user-friendly when the user is forced to fill in less query elements. Therefore, to evaluate the user-friendliness of the user interfaces, we measured how many elements, such as resources, literals and variables, had to be typed in to construct the query.

The averaged results of the system accuracy evaluation are presented in the Table 1. The results confirm that a system with the natural language-based querying interface, in contrast to graph-based search systems, is likely to be unable to find correct answers for a significant number of questions. It should be also noted that one of the key factors affecting the search results is the use of different knowledge bases. However, the comparison should still be regarded as objective as all of the systems contain data necessary to generate all relevant answers for each of the selected queries.

Table 2: Average number of elements required to create queries

System	SSE	GoR	NAGA
Number of elements per query	**2.786**	5.714	3.571

The Table 2 presents the average number of elements per query that the user has to type to formulate the query in each system. As can be seen, the features of SSE such as intuitive variables naming or automatic type filling (unique among other semantic search systems [2, 3]) may reduce the amount of data needed to construct a query. It may be concluded that the proposed system provides more useful responses than the other compared systems do, and more user-friendly query formulation interface.

5. CONCLUSIONS

We proposed an easily repeatable semantic search system evaluation methodology. We experimentally confirmed that graph-based search provide more satisfying results than the NLP-based systems. We demonstrated that the proposed system provides simplified user interface that not compromises the system usability.

Acknowledgments.

This work is supported by project O ROB 0025 01 financed by NCBR (Contract 0025/R/D1/2011/01) and by NCN under grant DEC-2011/01/D/ST6/06788.

6. REFERENCES

[1] D. Damljanovic, M. Agatonovic, and H. Cunningham, “Natural language interfaces to ontologies: Combining syntactic analysis and ontology-based lookup through the user interaction.” in *ESWC (1)*, ser. Lecture Notes in Computer Science, L. Aroyo, G. Antoniou, E. HyvÃűnen, A. ten Teije, H. Stuckenschmidt, L. Cabral, and T. Tudorache, Eds., vol. 6088. Springer, 2010, pp. 106–120.

[2] L. Han, T. Finin, and A. Joshi, “GoRelations: An intuitive query system for DBpedia.” in *JIST*, ser. Lecture Notes in Computer Science, J. Z. Pan, H. Chen, H.-G. Kim, J. Li, Z. Wu, I. Horrocks, R. Mizoguchi, and Z. Wu, Eds., vol. 7185. Springer, 2011, pp. 334–341.

[3] G. Kasneci, F. M. Suchanek, G. Ifrim, M. Ramanath, and G. Weikum, “NAGA: Searching and ranking knowledge,” in *24th International Conference on Data Engineering (ICDE 2008)*. Mexico: IEEE Press, 2008.

[4] V. Lopez, M. Fernandez, E. Motta, and N. Stieler, “Poweraqua: Supporting users in querying and exploring the semantic web.” *Semantic Web*, vol. 3, no. 3, pp. 249–265, 2012.

[5] “Facebook Graph Search,” https://www.facebook.com/about/graphsearch, 2013.

[6] “Question Answering over Linked Data,” http://greententacle.techfak.uni-bielefeld.de/∼cunger/qald/, 2013.