Imprecise SPARQL: Towards a Unified Framework for Similarity-Based Semantic Web Tasks

Christoph Kiefer

Department of Informatics, University of Zurich,
Binzmuehlestrasse 14, CH-8050 Zurich, Switzerland
kiefer@ifi.unizh.ch

Abstract. This proposal explores a unified framework to solve Semantic Web tasks that often require similarity measures, such as RDF retrieval, ontology alignment, and semantic service matchmaking. Our aim is to see how far it is possible to integrate user-defined similarity functions (UDSF) into SPARQL to achieve good results for these tasks. We present some research questions, summarize the experimental work conducted so far, and present our research plan that focuses on the various challenges of similarity querying within the Semantic Web.

1 Motivation

Semantic Web tasks such as ontology alignment, semantic service matchmaking, and similarity-based retrieval depend on some *notion of similarity* (at least if they are not solely based on logic). Therefore, researchers still try to find sound *user-defined similarity functions* (UDSF) to achieve good results for these tasks. Finding good similarity functions is, however, data- and context-dependent, and needs to be reconsidered every time new data is inspected. Nonetheless, good UDSFs are crucial for the success of the above-mentioned Semantic Web tasks.

Furthermore, in recent years, query languages for the Semantic Web such as RDQL and SPARQL have gained increasing popularity. The current W3C candidate recommendation of SPARQL, however, does not support UDSF to analyze the data during query processing. The goal of this project is to overcome this limitation and to develop a *unified framework* based on SPARQL to solve similarity-dependent Semantic Web tasks. The proposed iSPARQL framework should be easy to use and easily extendable to allow for user-defined, task-specific similarity functions. The "i" stands for *imprecise* indicating that two or more resources are compared by using similarity measures.

We strive for a robust implementation of similarity querying for the Semantic Web and its integration into SPARQL. The proposed iSPARQL approach should have a high degree of flexibility in terms of customization to the actual Semantic Web task.

2 Related Work

RDF Retrieval. Siberski *et al.* [19] propose SPARQL extensions allowing the user to query the Semantic Web with preferences. New keywords (`PREFERRING`, `CASCADE`) are added to the official SPARQL grammar in order to favor query answers which match user-defined preference criteria. Finally, the answers which are not dominated by any other answers (optimal according to the defined preference dimensions) are returned to the user.

Ontology Alignment. The task of ontology alignment (aka *ontology mapping/matching*) is a heavily researched field within the Semantic Web. Noy and Musen [15] present the PROMPT framework – a suite of tools including iPROMPT and ANCHORPROMPT – simplifying the comparing, aligning, and merging of ontologies of different origins. Furthermore, Doan *et al.* [5] propose the GLUE system that assists the user in finding mappings between ontologies using techniques from machine learning. A different methodology is proposed by Ehrig and Staab in [6]: based on QOM, ontologies can be aligned on different layers focusing on different (modeling) aspects of ontologies. Euzenat and Valtchev [7] propose an approach that is based on a specialized similarity measure to compare OWL-lite ontologies. Last, in a more recent paper, Tous and Delgado [20] map nodes of ontologies to matrices which capture the relationships of the mapped nodes among each other. Finally, a graph matching algorithm is applied to find mappings between the ontologies under comparison.

Matchmaking/Discovery. Klusch *et al.* [14] propose OWLS-MX to perform service matchmaking which adopts both, pure logic-based and Information Retrieval (IR) based techniques for the needs of hybrid semantic service matchmaking. Furthermore, Hau *et al.* [10] propose a similarity measure to compare Semantic Web services expressed in the OWL-S language. In addition, Jaeger *et al.* [11] present an approach for matching service inputs, service outputs, a service category, and user-defined service matching criteria. The four individual matching scores are aggregated to result in an overall matchmaking score.

Query Optimization. Query optimization strategies have been developed to reduce the complexity of Semantic Web queries to boost their runtime performance. Ruckhaus *et al.* [16] propose to estimate the cost and cardinality of individual query predicates based on selectivity estimations taken from [18].

Similarity Joins (Data Integration). To perform data integration, Cohen [4] presents WHIRL and the notion of *similarity joins* by which data is joined on *similarity* rather than on *equality*. In WHIRL, the TF-IDF weighting schema from IR [1] is applied together with the cosine similarity measure to determine the affinity of text. Similar approaches are proposed by Gravano *et al.* employing *string joins* [8] and *text joins* [9] in order to correlate information from different databases and web sources respectively.

3 General Problem Areas/Gaps

Numerous Semantic Web tasks rely on some *notion of similarity*, either to *compare ontologies* (for alignment and/or integration), or to *compare services* (for matchmaking and/or discovery), or to *compare resources* (for querying and similarity-based retrieval) among others. All of the approaches presented in Section 2 tackle one of these tasks *individually* (*i.e.*, in their own specific way). None of the approaches present a unified framework to solve them all. We made the following observations:

- To solve these tasks, Semantic Web researchers still try to find sound *user-defined similarity functions* (UDSF), which are crucial for the *success* of these tasks. However, good similarity functions are data- and context-dependent, and generally not easy to find.
- SPARQL in *combination* with UDSFs could be used to solve individual tasks. However, traditional SPARQL does not support querying ontologies with UDSF. It is not clear what the optimal solution would look like: an extension of the official SPARQL grammar or the exploitation of "magic properties" (aka *virtual triples* or *property functions*) as supported in ARQ.[1]
- The semantics and complexity of UDSF-extended SPARQL queries are unclear. Hence, they should be elaborated and formally studied.
- UDSF statements add an additional layer of *complexity* to SPARQL queries. Therefore, an approach for optimizing queries containing UDSFs should be provided. This is particularly important when executing *web-scale queries*. In other words: do UDSF-queries have the potential to scale to the web?

4 Research Plan

4.1 Choice of Datasets and Evaluation Strategy

So far we have experimented with the two matchmaking/retrieval test collections OWLS-TC[2] and the OWL MIT Process Handbook[3]. For our preliminary optimization experiments we used SwetoDblp[4], which focuses on bibliography information of computer science publications. Furthermore, we worked with EvoOnt[5] – a set of ontologies to model the domain of object-oriented software source code. We will use these datasets for the evaluations of our proposed unified framework.

4.2 Current State of Our Research

RDF Retrieval. iRDQL [2] is our extension of traditional RDQL with similarity joins to determine the similarity of Semantic Web resources.[6] A limitation of iRDQL is that it allows to utilize only one similarity measure per query and

[1] `http://jena.sourceforge.net/ARQ/`

[2] `http://projects.semwebcentral.org/projects/owls-tc/`

[3] `http://www.ifi.unizh.ch/ddis/mitph.html`

[4] `http://lsdis.cs.uga.edu/projects/semdis/swetodblp/`

[5] `http://www.ifi.unizh.ch/ddis/evoont.html`

[6] All similarity measures are implemented in SimPack, our generic library of similarity measures for the use in ontologies (`http://www.ifi.unizh.ch/ddis/simpack.html`).

it does not perform any query optimization. A demonstration of our current prototype implementation iSPARQL is available at `http://www.ifi.unizh.ch/ddis/isparql.html`. We will use this prototype as a starting point (and benchmark) for the new framework to be accomplished within this PhD thesis.
Matchmaking/Discovery. In [12], the applicability of our iSPARQL prototype is evaluated for the task of Semantic Web service discovery within the OWL MIT Process Handbook.
Query Optimization. Our first steps toward Semantic Web query optimization are presented in [3]. The proposed OptARQ approach investigates SPARQL query optimization by means of a rule-based query optimization engine. Optimization techniques for UDSF-queries, however, are not covered by OptARQ.
Analyzing Software Repositories. To highlight the benefits and applicability of the proposed unified framework to different, initially *non-Semantic Web tasks*, we realized the Coogle [17] and EvoOnt [13] projects for the tasks of software evolution analysis and visualization as well as design flaws detection.

4.3 Our Approach – Next Steps

The aim of this work is the design, specification, implementation, and evaluation of a unified framework for similarity-based Semantic Web tasks. There are several goals to achieve: the first goal consists of a detailed revision of our preliminary work. This will answer the question if the virtual triple approach taken so far is sufficient to solve the remaining challenges of such a unified framework. The second goal is the formal elaboration of the iSPARQL grammar, its semantics and complexity. As a third goal, we investigate query optimization techniques to boost the performance of UDSF-queries. Finally, the whole iSPARQL model and implementation will be evaluated for applicability to different application tasks (see Section 2).

To achieve the goals, the following steps are planned: a revision of the current prototype with special attention to its usability, flexibility, customizability, and scalability; the specification of the iSPARQL model, particularly the complexity and semantics of UDSF-queries; the implementation of the unified framework; the investigation of UDSF-query optimization techniques; and an evaluation of the applicability to different similarity-based Semantic Web tasks in terms of testing, usability, customization, and performance measurement.

5 Conclusions

In this proposal we outlined the need of a unified framework to solve similarity-based Semantic Web tasks, such as ontology alignment, service matchmaking, and RDF retrieval. Our approach extends traditional SPARQL with user-defined similarity functions (UDSF). The semantics and complexity of SPARQL-based similarity queries will be formally elaborated and query optimization techniques proposed. This systematical assessment will answer the questions of what is the *range of tasks* that can be solved with the iSPARQL system, what is the *performance* to solve these tasks, and what is its potential to *scale to the web*. It is important to realize that these tasks provide a kind of "stress test" for the usefulness of our unified framework.

References

1. R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval.* Addison Wesley, 1999.
2. A. Bernstein and C. Kiefer. Imprecise RDQL: Towards Generic Retrieval in Ontologies Using Similarity Joins. In *SAC 2006*, pages 1684–1689.
3. A. Bernstein, C. Kiefer, and M. Stocker. OptARQ: A SPARQL Optimization Approach based on Triple Pattern Selectivity Estimation. Technical Report ifi-2007.03, Department of Informatics, University of Zurich, 2007.
4. W. W. Cohen. Data Integration Using Similarity Joins and a Word-Based Information Representation Language. *TOIS*, 18(3):288–321, 2000.
5. A. Doan, J. Madhavan, R. Dhamankar, P. Domingos, and A. Halevy. Learning to match ontologies on the Semantic Web. *VLDB Journal*, 12(4):303–319, 2003.
6. M. Ehrig and S. Staab. QOM - Quick Ontology Mapping. In *GI Jahrestagung*, pages 356–361, 2004.
7. J. Euzénat, D. Loup, M. Touzani, and P. Valtchev. Ontology Alignment with OLA. In *ISWC 2004*, pages 333–337.
8. L. Gravano, P. G. Ipeirotis, H. V. Jagadish, N. Koudas, S. Muthukrishnan, and D. Srivastava. Approximate string joins in a database (almost) for free. In *VLDB Journal*, pages 491–500, 2001.
9. L. Gravano, P. G. Ipeirotis, N. Koudas, and D. Srivastava. Text Joins in an RDBMS for Web Data Integration. In *WWW 2003*, pages 90–101.
10. J. Hau, W. Lee, and J. Darlington. A Semantic Similarity Measure for Semantic Web Services. In *WWW 2005*.
11. M. C. Jaeger, G. Rojec-Goldmann, G. Mühl, C. Liebetruth, and K. Geihs. Ranked Matching for Service Descriptions using OWL-S. In *KiVS 2005*, Informatik Aktuell, pages 91–102.
12. C. Kiefer, A. Bernstein, H. J. Lee, M. Klein, and M. Stocker. Semantic Process Retrieval with iSPARQL. In *ESWC 2007*.
13. C. Kiefer, A. Bernstein, and J. Tappolet. Mining Software Repositories with iSPARQL and a Software Evolution Ontology. In *MSR 2007*.
14. M. Klusch, B. Fries, and K. Sycara. Automated Semantic Web Service Discovery with OWLS-MX. In *AAMAS 2006*, pages 915–922.
15. N. F. Noy and M. A. Musen. The PROMPT Suite: Interactive Tools for Ontology Merging and Mapping. *IJHCS*, 59(6):983–1024, 2003.
16. E. Ruckhaus, E. Ruiz, and M.-E. Vidal. Query Optimization in the Semantic Web. In *ALPSWS 2006*.
17. T. Sager, A. Bernstein, M. Pinzger, and C. Kiefer. Detecting Similar Java Classes Using Tree Algorithms. In *MSR 2006*.
18. P. G. Selinger, M. M. Astrahan, D. D. Chamberlin, R. A. Lorie, and T. G. Price. Access path selection in a relational database management system. In *SIGMOD 1979*, pages 23–34.
19. W. Siberski, J. Z. Pan, and U. Thaden. Querying the Semantic Web with Preferences. In *ISWC 2006*.
20. R. Tous and J. Delgado. A Vector Space Model for Semantic Similarity Calculation and OWL Ontology Alignment. In *DEXA 2006*.