DC Proposal: Automatically transforming keyword queries to SPARQL on large-scale knowledge bases

Saeedeh Shekarpour
Supervisor: Dr. Sören Auer

Universität Leipzig, Institut für Informatik, AKSW,
Postfach 100920, D-04009 Leipzig, Germany,
{lastname}@informatik.uni-leipzig.de
http://aksw.org

Abstract. Most Web of Data applications focus mainly on using SPARQL for issuing queries. This leads to the Web of Data being difficult to access for non-experts. Another problem that will intensify this challenge is when applying the algorithms on large-scale and decentralized knowledge bases. In the current thesis, firstly we focus on the methods for transforming keyword-based queries into SPARQL automatically. Secondly, we will work on improving those methods in order to apply them on (a large subset of) the Linked Data Web. In an early phase, a heuristic method was proposed for generating SPARQL queries out of arbitrary number of keywords. Its preliminary evaluation showed promising results. So, we are working on the possible improvements for applying that on the large-scale knowledge bases.

1 Introduction

Web of Data is growing at an astounding rate (currently amounting to 28 Billion triples[1]) and contains a wealth of information on a large number of domains which enables automated agents and other applications to access the information on the web more intelligently. The vendor-independent standard SPARQL has been specified and allows to query this knowledge easily. Yet, while SPARQL is very intuitive for SQL-affine users, it is very difficult to use for lay users who are not familiar with the concepts behind the Semantic Web. Because a naive user needs to acquire both knowledge about the underlying ontology and proficiency in formulating SPARQL queries to query the endpoint. Also, with the huge amount of background knowledge present in Linked Data, it is difficult for naive as well as proficient users to formulate SPARQL queries. Consequently, there is a blatant need for another mean to query the Semantic Web that is appealing for novice users. In other word, the naive user prefers to hold the traditional paradigm of interaction with search engines which is based on natural language or keyword-based queries.

[1] http://www4.wiwiss.fu-berlin.de/lodcloud/state/ (June 19th, 2011)

Nowadays, keyword-based search is the most popular and convenient way for finding information on the Web. The successful experience of keyword-based search in document retrieval and the satisfactory research results about the usability of this paradigm [23] are convincing reasons for using the keyword search paradigm to the Semantic Web.

Document retrieval approaches use probabilistic foundations combining with ranking algorithms for selecting documents containing those keywords which practically works well. Since the nature of RDF data is different and potentially, information need is more complex, we require novel and efficient algorithms for querying Data Web. RDF data has a directed and multiple edges graph structure with typed vertices and labeled edges. The fundamental base of that is entity which has three main features as type, attributes and relations with other entities. Therefore, objective of search on this data is retrieval of entities or maybe along with some attributes and relations.

Although it is around one decade that semantic search has been targeted by researchers, to the best of our knowledge, still there is not any service which practically obviate this need. Services such as *Sindice* [25], *Sig.ma* [24], *Swoogle* [6] or *Watson* [5] offer simple search services[2], but are either restricted to the retrieval of single RDF documents or in the case of Sig.ma to the retrieval of information about a single entity from different sources.

In essence, the problem which we are addressing in the proposal can be stated as follows:

How can we interpret a user query in order to locate and exploit relevant information for answering the user's query using large-scale knowledge bases from the Linked Data Web?

Figure 1 shows a birds-eye-view of the envisioned research. Based on a set of user-supplied keywords, first, candidate IRIs (Internationalized Resource Identifier) for each of the keywords issued by the user is computed. Then, by using an inference mechanism, a subgraph based on the identified IRIs is extracted and represented to the user as the answer.

2 Research Challenges

To query Web of data by naive user, the NL (Natural Language) queries should be interpreted in accordance to user's intention. Because natural language inherently includes ambiguity, precise interpretation is difficult. For instance, relations of terms in an NL query are prevalently either unknown or implicit. In addition, the retrieval of knowledge from a large-scale domain such as Linked Data Web needs efficient and scalable algorithms. Some algorithms and applications can be robust in a small or medium scale, while they may not be suitable when applied to large scale knowledge bases present in the entire Linked Data Web. Therefore, a suitable approach for querying the Linked Data Web is essential for retrieving information both efficiently and precisely in a way that can be easily utilized

[2] These systems are available at: `http://sindice.com`, `http://sig.ma`, `http://swoogle.umbc.edu`, `http://kmi-web05.open.ac.uk/WatsonWUI`

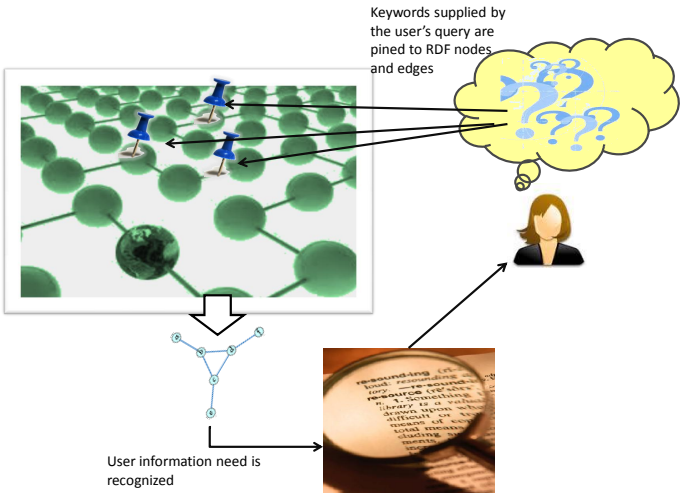

Fig. 1. Birds-eye-view of the envisioned research.

by a naive user. Based on this observation, we will investigate 3 fundamental questions:

- Can we (semi-)automatically interpret user queries using background knowledge from the Linked Data Web possibly involving the user in a feedback loop?
- How can we efficiently retrieve meaningful answers for the user query from large scale knowledge bases on the Linked Data Web?
- How can we cluster and represent the retrieved answers to the user in a meaningful way?

With regard to the first problem, we will use the keyword-based query that a user enters in a traditional interface, as a model. This model involves some difficulties due to the ambiguity, because in natural language, a query may stand for different meanings and be interpreted in different ways. This ambiguity in the case of keyword-based query is even intensified. Therefore, the main task is to interpret the user query using background knowledge. In the case of the second question, the user query needs to be transformed to the formal SPARQL query syntax and sent to different knowledge bases for retrieving relevant instances.

For tackling the third question, we will investigate clustering methods and extend an approach which can be applied on the graph nature of the answers. Therefore, the instances will be delivered in different categories.

Another problem that will intensify the aforementioned challenges is when applying the algorithms on large-scale and decentralized datasets, which will pose several difficulties. For example, mapping of user query terms to various ontologies causes severe ambiguity due to redundancy in entities and terminologies.

3 Methodology

Figure 2 indicates the main components of this thesis. The first component consists of processing and analyzing the keyword-based query employing a linguistic component. We will use the GATE infrastructure in order to pre-process the query. The output of this component will be a series of terms associated with the input query. Afterwards, extracted terms of the user query will be mapped to entities which exist in the underlying ontology and knowledge base such as DBpedia. The mapping process will be performed by applying some similarity metrics with regarding the context and using synonyms provided by lexical resources such as WordNet.

In the second component, the relation of the mapped entities will be recognized by using a heuristics approach over a graph traversal method. We aim to make a comparison between the existing approaches and our proposed method. We will design an interface that enables a user to interact with the system without any special knowledge of vocabularies or structure of the knowledge base. We need to mention that various interpretations of the user query are created since different options for mapping user terms to entities and connecting these mapped entities exist. With regard to tools, we use Jena and Virtuoso. The results of both components can be evaluated based on a comparison with a gold standard (i.e. a hand-crafted collection of natural language queries and their equivalent SPARQL queries with automatically constructed SPARQL queries obtained from the inference engine). Some accuracy and performance metrics such as precision, recall, fscore, MMR and runtime will be chosen. Another component is related to clustering of the instances. We will investigate and develop a clustering method based on context and background knowledge for organizing instances.

At the last phase, we will work on improving the previous components and develop them in order to apply them on (a large subset of) the Linked Data Web. For this, we will use a benchmark method for development. For this, a set of datasets which are large in terms of both the number of triples and variety of the background knowledge will be chosen. In an iterative phase, each of the earlier components will be individually applied on a selected set of datasets (the size of the selected set of datasets will increase in each phase). By comparing the performance metrics in each phase, appropriate revisions will be done on the algorithms and methods.

4 A Synopsis of the Current Situation

In an early phase, a novel method for generating SPARQL queries based on two user-supplied keywords was proposed [21]. Since this method is based on simple operations, it can generate SPARQL queries very efficiently. Also it is completely agnostic of the underlying knowledge base as well as its ontology schema. We currently use DBpedia as the underlying knowledge base, but this method is easily transferable to the whole Data Web. The implementation of this method

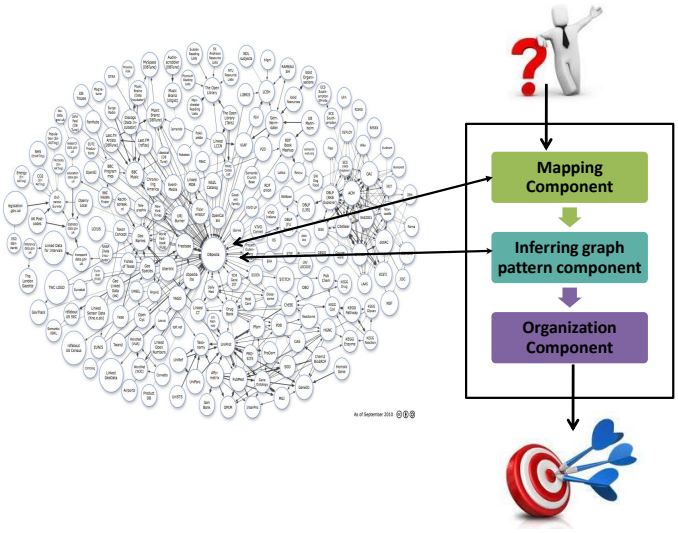

Fig. 2. Architecture of the envisioned semantic search algorithm.

is publicly available at: http://lod-query.aksw.org. Table 1 shows some samples of keywords for which the application is capable to retrieve suitable results. These keywords were categorized based on the type of queries which they can answer in three categories, i.e. similar instances, characteristics of an instance and associations between instances.

Keywords	**Answers**
Instance characteristics.	
Kidman spouse	`d:Kidman dp:spouse Keith Urban .`
Iran language	`d:Iran dp:Language d:Persian_language .`
Associations between instances.	
Obama Clinton	`d:Obama dp:predecessor d:Bush . d:Bush dp:predecessor d:Clinton .`
Volkswagen Porsche	`d:Volkswagen_Group dp:subsidiary d:Volkswagen .`
Similar instances.	

Germany Island	`1. d:Germany dp:Islands d:Rgen. d:Rgen a do:Island.`
	`2. d:Germany dp:Islands d:Fhr. d:Fhr a do:Island.`
	`3. d:Germany dp:Islands d:Sylt. d:Sylt a do:Island.`

Lost Episode	`1. d:Raised_by_Another dp:series dbp:Lost.`
	`d:Raised_by_Another a do:TVEpisode.`
	`2. d:Homecoming dp:series dbp:Lost.`
	`d:Homecoming a do:TVEpisode.`
	`3. d:Outlaws dp:series dbp:Lost.`
	`d:Outlaws a do:TVEpisode.`

Table 1. Samples of keywords and results.

In the next phase, we proposed a method for generating SPARQL queries for arbitrary number of keywords [Submitting Article to WSDM 2012]. Its base is an inference mechanism embedded in a graph traversal approach which both

accurately and efficiently builds SPARQL queries for the straightforward interpretations of the user's queries. Implementation of this method is also available at: http://sina.aksw.org. This application generats SPARQL query for those keyword-based queries which can be converted to a conjunctive query. A conjunctive query is a conjunction of triple patterns (based on the standard notions of the RDF and SPARQL specifications). For instance, consider the example shown below:

Example 1. Let suppose we have the keywords *people*, *birthplace*, *deathplace* and *Southampton*. A straightforward interpretation of these keywords is *people whose birthplace and deathplace are in southampton*. It can be expressed by a conjunctive query as $(?p\ a\ Person) \wedge (?p\ birthplace\ Southampton) \wedge (?p\ deathplace\ Southampton)$

Because of the promising results of the preliminary evaluation, we are currently working on this method for improving the overall functionality with regard to accuracy and efficiency so as to run that over more number of knowledge bases.

5 State of the Art

With the advent of the Semantic Web, information retrieval and question answering approaches were adapted for making use of ontologies. We can roughly divide related work into ontology-based information retrieval, ontology-based question answering and keyword search on structured data.

Ontology-based information retrieval Approaches falling into this category annotate and index documents using a background ontology. The retrieval process is subsequently carried out by mapping user query terms onto these semantic document annotations. The approaches described in [26,24,6,5,25,3,9,19] are examples of this paradigm. All these approaches use background knowledge to enhance the retrieval accuracy, however, they do not utilize the background knowledge for semantically answering user queries.

Ontology-based question answering Approaches falling into this category take a natural language question or a keyword-based query and return matching knowledge fragments drawn from the knowledge base as the answer. There are two different methods: (1) Using linguistic approaches for extracting complete triple-based patterns (including relations) from the user query and matching these triples to the underlying ontology (e.g. PowerAqua [16], OntoNL [12] and FREyA [4]). (2) Detecting just entities in the user query and discovering relations between these entities by analysing the knowledge base. Examples for this second group are KIM [18] and OntoLook [13] and [20,7,27,17]. In these two approaches the RDF data is considered to be a directed graph and relations among entities are found through sequences of links (e.g. using graph traversal). Sheth [22] introduced the term *semantic association* for describing meaningful and complex relations between entities. Our work differs from these approaches, since it

is completely independent of the underlying schema. Furthermore, schema information is in our approach just *implicitly* taken into account, so a complex induction procedure is not required.

Keyword search on relational and XML data With the beginning of the millennium, research on keyword search on relational and XML data attracted research interest. Meanwhile there exist many approaches such as [1], [11], [10], [14] for the relational domain and [8], [15], [2] for the XML domain. Especially the relational domain is relevant to our work due to the similarities to the RDF datamodel. All these approaches are based on *schema graphs* (i.e. a graph where tables and their primary-foreign key relations are represented as nodes and edges, respectively). In our work, we do not rely on an explicitly given schema, which is often missing for datasets on the Web of Data. However, achieving sufficient performance for instant query answering is more an issue in the RDF case, which is why our approach is currently limited to two keywords.

References

1. Sanjay Agrawal, Surajit Chaudhuri, and Gautam Das. Dbxplorer: A system for keyword-based search over relational databases. In *ICDE*, pages 5–16. IEEE Computer Society, 2002.
2. Liang Jeff Chen and Yannis Papakonstantinou. Supporting top-k keyword search in xml databases. In *ICDE*, pages 689–700. IEEE, 2010.
3. Fabio Crestani. Application of spreading activation techniques in information retrieval. *Artif. Intell. Rev.*, 11(6):453–482, 1997.
4. Danica Damljanovic, Milan Agatonovic, and Hamish Cunningham. FREyA: an Interactive Way of Querying Linked Data Natural Language. In *Proceedings of 1st Workshop on Question Answering over Linked Data (QALD-1), Collocated with the 8th Extended Semantic Web Conference (ESWC 2011)*, Heraklion, Greece, 2011.
5. M. D'aquin, E. Motta, M. Sabou, S. Angeletou, L. Gridinoc, V. Lopez, and D. Guidi. Toward a new generation of semantic web applications. *Intelligent Systems, IEEE*, 23(3):20–28, 2008.
6. Li Ding, Timothy W. Finin, Anupam Joshi, Rong Pan, R. Scott Cost, Yun Peng, Pavan Reddivari, Vishal Doshi, and Joel Sachs. Swoogle: a search and metadata engine for the semantic web. In David A. Grossman, Luis Gravano, ChengXiang Zhai, Otthein Herzog, and David A. Evans, editors, *CIKM*, pages 652–659. ACM, 2004.
7. Ramanathan V. Guha, Rob McCool, and Eric Miller. Semantic search. In *WWW*, pages 700–709, 2003.
8. Lin Guo, Feng Shao, Chavdar Botev, and Jayavel Shanmugasundaram. Xrank: Ranked keyword search over xml documents. In Alon Y. Halevy, Zachary G. Ives, and AnHai Doan, editors, *SIGMOD Conference*, pages 16–27. ACM, 2003.
9. Markus Holi and Eero Hyvönen. Fuzzy view-based semantic search. In *ASWC*, volume 4185 of *LNCS*, pages 351–365. Springer, 2006.
10. Vagelis Hristidis, Luis Gravano, and Yannis Papakonstantinou. Efficient ir-style keyword search over relational databases. In *VLDB*, pages 850–861, 2003.
11. Vagelis Hristidis and Yannis Papakonstantinou. Discover: Keyword search in relational databases. In *VLDB*, pages 670–681. Morgan Kaufmann, 2002.

12. Anastasia Karanastasi, Alexandros Zotos, and Stavros Christodoulakis. The OntoNL framework for natural language interface generation and a domain-specific application. In *Digital Libraries: Research and Development, First International DELOS Conference, Pisa, Italy*, pages 228–237. 2007.
13. Yufei Li, Yuan Wang, and Xiaotao Huang. A relation-based search engine in semantic web. *IEEE Trans. Knowl. Data Eng.*, 19(2):273–282, 2007.
14. Fang Liu, Clement T. Yu, Weiyi Meng, and Abdur Chowdhury. Effective keyword search in relational databases. In Surajit Chaudhuri, Vagelis Hristidis, and Neoklis Polyzotis, editors, *SIGMOD Conference*, pages 563–574. ACM, 2006.
15. Ziyang Liu and Yi Chen. Reasoning and identifying relevant matches for xml keyword search. *PVLDB*, 1(1):921–932, 2008.
16. Vanessa Lopez, Victoria S. Uren, Enrico Motta, and Michele Pasin. Aqualog: An ontology-driven question answering system for organizational semantic intranets. *J. Web Sem.*, 5(2):72–105, 2007.
17. Xiaomin Ning, Hai Jin, Weijia Jia, and Pingpeng Yuan. Practical and effective ir-style keyword search over semantic web. *Inf. Process. Manage.*, 45(2):263–271, 2009.
18. Borislav Popov, Atanas Kiryakov, Angel Kirilov, Dimitar Manov, Damyan Ognyanoff, and Miroslav Goranov. Kim - semantic annotation platform. *Journal of Natural Language Engineering*, 10(3-4):375–392, September 2004.
19. Cristiano Rocha, Daniel Schwabe, and Marcus Poggi de Aragão. A hybrid approach for searching in the semantic web. In *WWW*, pages 374–383. ACM, 2004.
20. Guus Schreiber, Alia Amin, Lora Aroyo, Mark van Assem, Victor de Boer, Lynda Hardman, Michiel Hildebrand, Borys Omelayenko, Jacco van Osenbruggen, Anna Tordai, Jan Wielemaker, and Bob Wielinga. Semantic annotation and search of cultural-heritage collections: The MultimediaN E-Culture demonstrator. *Journal of Web Semantics*, 6(4):243–249, 2008.
21. Saeedeh Shekarpour, Sren Auer, Axel-Cyrille Ngonga Ngomo, Daniel Gerber, Sebastian Hellmann, and Claus Stadler. Keyword-driven sparql query generation leveraging background knowledge. In *International Conference on Web Intelligence*, 2011.
22. Amit Sheth, Boanerges Aleman-Meza, I. Budak Arpinar, Christian Halaschek-Wiener, Cartic Ramakrishnan, Yashodhan Warke Clemens Bertram, David Avant, F. Sena Arpinar, Kemafor Anyanwu, and Krys Kochut. Semantic association identification and knowledge discovery for national security applications. *Journal of Database Management*, 16(1):33–53, 2005.
23. Thanh Tran, Tobias Math, and Peter Haase. Usability of keyword-driven schema-agnostic search. In Lora Aroyo, Grigoris Antoniou, Eero Hyvnen, Annette ten Teije, Heiner Stuckenschmidt, Liliana Cabral, and Tania Tudorache, editors, *ESWC (2)*, volume 6089 of *Lecture Notes in Computer Science*, pages 349–364. Springer, 2010.
24. Giovanni Tummarello, Richard Cyganiak, Michele Catasta, Szymon Danielczyk, Renaud Delbru, and Stefan Decker. Sig.ma: Live views on the web of data. *J. Web Sem.*, 8(4):355–364, 2010.
25. Giovanni Tummarello, Renaud Delbru, and Eyal Oren. Sindice.com: weaving the open linked data. pages 552–565, 2007.
26. Haofen Wang, Qiaoling Liu, Thomas Penin, Linyun Fu, Lei Zhang 0007, Thanh Tran, Yong Yu, and Yue Pan. Semplore: A scalable ir approach to search the web of data. *J. Web Sem.*, 7(3):177–188, 2009.
27. Gideon Zenz, Xuan Zhou, Enrico Minack, Wolf Siberski, and Wolfgang Nejdl. From keywords to semantic queries – incremental query construction on the semantic web. *Web Semantics*, 7(3):166–176, 2009.