Web Semantics: Science, Services and Agents on the World Wide Web 30 (2015) 39–51

Contents lists available at ScienceDirect

Web Semantics: Science, Services and Agents on the World Wide Web

journal homepage: www.elsevier.com/locate/websem

SINA: Semantic interpretation of user queries for question answering on interlinked data

Saeedeh Shekarpour [a,b,*], Edgard Marx [b], Axel-Cyrille Ngonga Ngomo [b], Sören Auer [a,b,c]

[a] Department of Computer Science, IFI/AKSW, Universität Leipzig, Germany
[b] Department of Enterprise Information Systems (EIS), Institute for Applied Computer Science at University of Bonn, Germany
[c] Fraunhofer Institute for Intelligent Analysis and Information Systems, Bonn, Germany

ARTICLE INFO

Article history:
Received 16 July 2013
Received in revised form
23 May 2014
Accepted 18 June 2014
Available online 26 June 2014

Keywords:
Keyword search
Question answering
Hidden Markov model
SPARQL
RDF
Disambiguation

ABSTRACT

The architectural choices underlying Linked Data have led to a compendium of data sources which contain both duplicated and fragmented information on a large number of domains. One way to enable non-experts users to access this data compendium is to provide keyword search frameworks that can capitalize on the inherent characteristics of Linked Data. Developing such systems is challenging for three main reasons. First, resources across different datasets or even within the same dataset can be homonyms. Second, different datasets employ heterogeneous schemas and each one may only contain a part of the answer for a certain user query. Finally, constructing a federated formal query from keywords across different datasets requires exploiting links between the different datasets on both the schema and instance levels. We present SINA, a scalable keyword search system that can answer user queries by transforming user-supplied keywords or natural-languages queries into conjunctive SPARQL queries over a set of interlinked data sources. SINA uses a hidden Markov model to determine the most suitable resources for a user-supplied query from different datasets. Moreover, our framework is able to construct federated queries by using the disambiguated resources and leveraging the link structure underlying the datasets to query. We evaluate SINA over three different datasets. We can answer 25 queries from the QALD-1 correctly. Moreover, we perform as well as the best question answering system from the QALD-3 competition by answering 32 questions correctly while also being able to answer queries on distributed sources. We study the runtime of SINA in its mono-core and parallel implementations and draw preliminary conclusions on the scalability of keyword search on Linked Data.

1. Introduction

The principles underlying Linked Data have been applied worldwide to engender the Linked Open Data Cloud, a compendium of more than 300 datasets and more than 31 billions triples.[1] Within this compendium, millions of resources are described, partly over several datasets [1]. The current standard for accessing this wealth of data is the SPARQL query language. Yet, SPARQL is too complex to be used by non-expert users. Consequently, several search approaches have been developed over the last years to enable non-experts to access these datasets (e.g., [2–7]). While these approaches differ in their details (see Section 7), they can all be positioned on the following spectrum: on one end of the spectrum are simple keyword search systems that rely on traditional information retrieval approaches to retrieve resources that bear a label similar to the input of the user. We dub such approaches *data-semantics-unaware keyword search* as they do not take the semantics explicated by the data into consideration. The main advantage of such approaches is that they scale well as they can make use of the results of decades of research carried out in the field of information retrieval. On the other end of the spectrum, we find *question answering systems*, which assume a natural-language query as input and convert this query into a full-fledged SPARQL query. These systems rely on natural-language processing tools such as POS tagging and dependency parsers to detect the relations between the elements of the query. The detected relations are then mapped to SPARQL constructs.

The basic idea behind this work is to devise a *data-semantics-aware keyword search* approach, which stands in the middle of

* Corresponding author at: Department of Computer Science, IFI/AKSW, Universität Leipzig, Germany. Tel.: +49 17684419554.
E-mail address: sa.shekarpour@gmail.com (S. Shekarpour).

[1] See http://lod-cloud.net/state/ for more details.

http://dx.doi.org/10.1016/j.websem.2014.06.002
1570-8268/

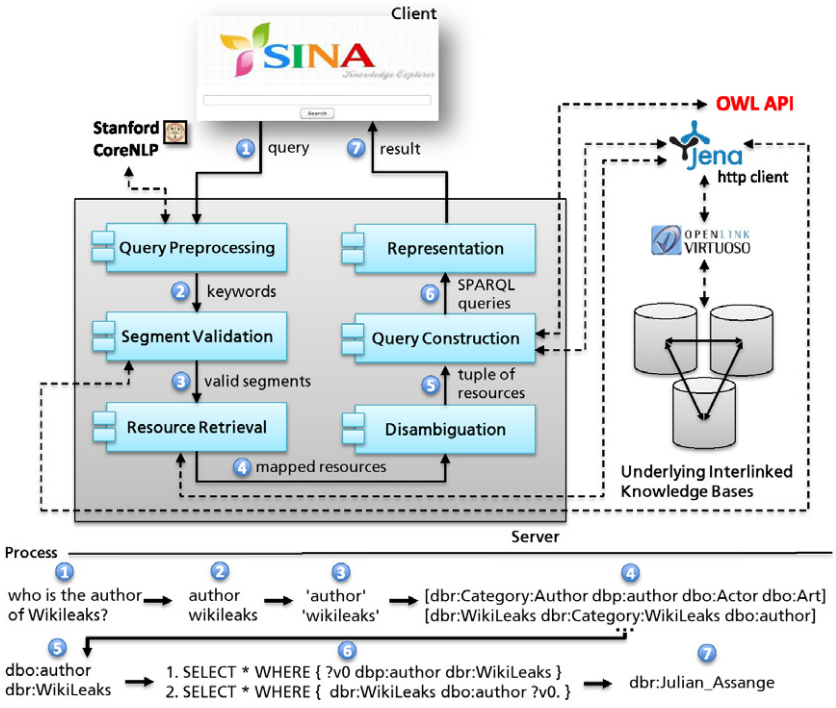

Fig. 1. Architecture of SINA search engine.

the spectrum. Our approach aims to achieve maximal flexibility by being able to generate SPARQL queries from both natural-language queries and keyword queries. This goal is achieved by limiting the type of formal queries (i.e. SPARQL queries) that our approach can generate to conjunctive SPARQL queries. Several challenges need to be addressed to devise such an approach: First, a query segmentation and disambiguation approach for mapping input query to resources has to be devised. To do so, statistical information of the retrieved resources has to be retrieved. Then, a method for generating conjunctive federated SPARQL queries, which can be sent to SPARQL endpoint to retrieve relevant data must be developed.

In this paper, we show how our framework, SINA, implements these different steps. In contrast to previous approaches, SINA can make use of the topology of Linked Data by exploiting links between resources to devise federated SPARQL queries. Thus, it can deal with both retrieving data from either a single dataset or several interlinked datasets. Consequently, it can be used over the whole of the Linked Open Data Cloud. We present a thorough evaluation of our approach on three different datasets: we use the QALD-1 to detect optimal parameters for SINA and present a first evaluation of the approach on this benchmark dataset. We then run SINA on the QALD-3 benchmark and show that we can generate the correct SPARQL queries for 32 of these queries, thus achieving the same results as the best system tested in the benchmark. Finally, we evaluate SINA in a federated scenario against the queries from the life science domain used in [1]. In an effort to make SINA easily portable, we refrained from using dataset-specific indexes and rely fully on the SPARQL endpoint when constructing SPARQL queries. To ensure that our approach still achieves acceptable runtimes, we implemented both a parallel and a sequential version of SINA. In the second part of the evaluation, we thus present a study of SINA's runtime on the QALD-3 benchmark.

This paper is organized as follows: In the subsequent section, we introduce the architecture of the proposed search engine. In Section 3, we present the problem at hand in more detail and some of the notations and concepts used in this work. Section 4 presents the proposed disambiguation method in detail along with the evaluation of the bootstrapping. In Section 5, we then present the key steps of our algorithm for constructing a conjunctive query. Our evaluation results are presented in Section 6 while related work is reviewed in Section 7. We close with a discussion and future work.

2. Overview

In this section, we describe the high-level architecture and implementation of the SINA search engine. Fig. 1 illustrates the architecture of SINA, which comprises six main components. Each component consumes the output of the previous component:

1. The **query preprocessing** component receives the textual input query and applies three functions: (1) Tokenization: extraction of individual keywords, removing punctuation and capitalization. (2) Stop word removal: removal of common words such as articles and prepositions. Since in this version, we do not recognize type of answers, thus wh-questions are removed as stop words. (3) Word lemmatization: determining the lemma of the remaining keywords.
2. The **segment validation** component groups keywords to form segments. This component validates the grouped segments with respect to the available resources in the underlying knowledge base(s).
3. The **resource retrieval** component obtains relevant resources from the underlying knowledge bases. The retrieval is based on the string matching between valid segments and the `rdfs:label` of resources. Furthermore, more resources are inferred from lightweight `owl:sameAs` reasoning.
4. The **disambiguation** component determines the best subset of resources for the given input query.
5. The **query construction** component results in formal queries (i.e. SPARQL query) using the graph-structure of data.
6. The **representation** component shows the retrieved results after evaluating the generated SPARQL queries.