QUESTION ANSWERING in LINKED DATA for SCIENTIFIC EXPLORATION

Samir Tartir
Department of Software Engineering
Philadelphia University
Amman, 19392 Jordan
startir@philadelphia.edu.jo

I. Budak Arpinar
Department. of Computer Science
University of Georgia
Athens, GA 30602-7407
budak@cs.uga.edu

Mustafa Nural
Department. of Computer Science
University of Georgia
Athens, GA 30602-7407
nural@cs.uga.edu

ABSTRACT

As more data is being semantically annotated, it is getting more common that researchers in multiple disciplines rely on semantic repositories that contain large amounts of data in the form of ontologies as a source of information. One of the main issues currently facing these researchers is the lack of easy-to-use interfaces for data retrieval, due to the need to use special query languages or applications. In addition, the knowledge in these repositories might not be comprehensive or up-to-date due to several reasons, such as the discovery of new knowledge in the field after the repositories were created. In this paper, we present SemanticQA system that allows users to query semantic data repositories using natural language questions. If a user question cannot be answered solely from the populated ontology, SemanticQA detects the failing parts and attempts to answer these parts from web documents and glues the partial answers to reply to the whole questions, which might involve a repetition of the same process if other parts fail.

1. INTRODUCTION

Large amounts of data in many disciplines are continuously being added to semantic or non-semantic web repositories as a result of continuing research in different scientific fields. As a result, it is becoming an increasing challenge for researchers to use these repositories efficiently and at the same time cope with this fast pace of the introduction of new knowledge [5]. It is critical to provide an easy to use querying capability for experts of different fields, especially who like to pose cross-discipline queries on these repositories. In this paper, we present a hybrid natural language question answering system (SemanticQA) on scientific Linked Data Sets (i.e. populated ontologies) as well as scientific literature in the form of publications. SemanticQA processes a scientist's information need, expressed in a Natural Language (NL) query. Initially, SemanticQA assists users in building their questions as they type, by presenting them with relevant suggestions extracted from the ontology based on their previous input. Then, it retrieves relevant answers from well-established Linked Data Sets (LDS). If the answer is not found in LDS, SemanticQA gathers all the relevant cues and conducts a semantic search on relevant publications for answer extraction. Final steps include qualifying and prioritizing available answers from these sources, and, presenting and explaining answers in an effective manner.

Web Science Conf. 2010, April 26-27, 2010, Raleigh, NC, USA.

1.1 Current approaches

Systems that answer user questions are being introduced in a fast pace. However, most of the current approaches for question answering have some limitations. In some approaches like EBIMed [8] and TextPresso [7], any entered input is processed as keywords, without consideration of the semantics of the domain questions need to be answered in. In addition, many approaches are answering questions from locally stored resources, which can be a limited resource, especially in domains that are constantly evolving. Powerset (www.powerset.com) is a commercial question answering system, which shows promising results. PANTO [11] also processes natural language questions using an ontology. As with the previous non-semantic approaches, they are all single-sourced, either answering a question from the ontology alone, or from a set of web documents without allowing answers from different sources to be integrated together to answer the whole question.

1.2 Example Scenario

Consider a scientist who is interested in finding "genes which are involved in Carbon Fixation", which is a bio-chemical process. Some answers can be directly found in a single LDS or scientific article. Yet some others can be located by tracing the complex associations spanning multiple intermediate bio-chemical entities and processes. Connecting-the-dots for finding the answers can be difficult even for an expert eye in this case. SemanticQA has the ability to assemble semantic associations first from LDS and then scientific literature next to facilitate scientific exploration on cross LDS and scientific publications. Once the user's NL query is processed, SemanticQA forms multiple SPARQL queries internally using linguistic and semantic techniques to be processed on LDS. This scheme allows the system to divide the question into smaller parts that can be processed independently from each other. If some queries fail, this indicates that some parts of the question could not be answered only from the LDS. In this case, multiple collections of keywords are generated to search the scientific literature. The answers are extracted from these documents and then ranked using a novel measure, the Semantic Answer Score [9], which we devised that extracts the best answer from relevant documents and returns it to the system so it can be used to answer the

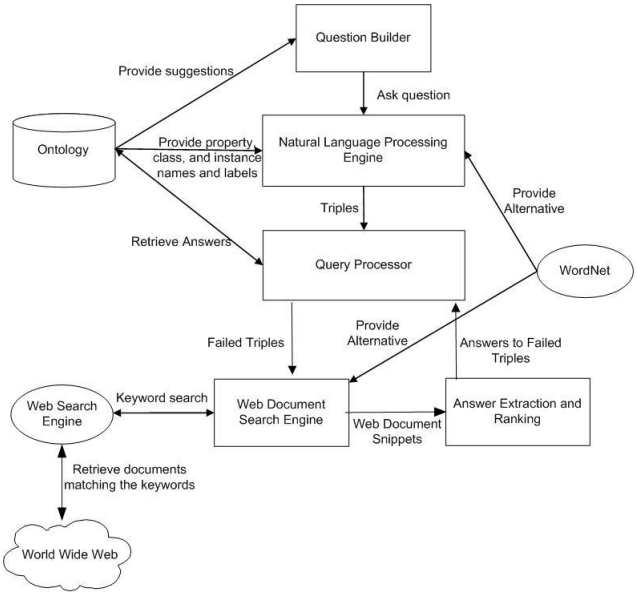

Figure 1: Architecture of SemanticQA

whole query. Figure 1 illustrates an overall architectural diagram of SemanticQA.

2. SEMANTICQA COMPONENTS

2.1 Question Builder

This component helps users to build questions in NL that use terms of the LDS. These terms are presented to the user as the query is being formed depending on what the user has been previously entered and what he/she started typing. Suggestions can be question words, "stop" words, LDS classes, relationships, or instances. For example, if the user has entered "professor" while using a university ontology, they would be presented with suggestions based on the properties of the class "professor, such as "teaching", or "advisor".

2.2 NLP Engine (Question Processor)

The main task of question processor is to map the contents of the question to LDS entities prior to answering it by exhaustively matching all word combinations in the question, starting from the largest, to ontology entities. For example, if the user asks: "Who is the advisor of John Smith?" some of the combinations that will be generated are as follows:

- "The advisor of John Smith"
- "The advisor of"
- "John Smith"

SemanticQA's matching performance is enhanced by "indexing" the contents of the LDS in advance by building three separate vectors for properties, classes and instances, and stored in a vector-space database constructed using Lucene [3]. The matching of question word combinations is performed in three phases. In the first phase, we map the combinations to properties of the LDS. Then the combinations are mapped to LDS classes. Finally combinations are matched to LDS instances. If a match is not found then we find alternatives to the word combination from WordNet [2]. For each alternative we do the same process we did with the original combination until we find a match. So, for the previous question the following triples are generated:

```
?prof rdf:type uni:Professor
<null> univ-bench:advisor ?prof
<univ-bench:JohnSmith univ-bench:advisor null>
```

2.3 Query Processor

The query processor's task is to provide user with the best answer to the question from the populated ontology(i.e. LDS) and web documents, if needed. The query processor first combines the triples generated in the NLP engine into a single SPARQL query. For example, if the question above was changed to: "Where did the advisor of John Smith get his degree from?" the following SPARQL query will be generated:

```
SELECT ?placeLabel
WHERE {
univ-bench:JohnSmith univ-bench:advisor ?person .
?person univ-bench:degreeFrom ?place .
?place rdfs:label ?placeLabel .
}
```

This query is issued against the ontology to attempt to retrieve the answer directly from ontology. If an answer is found, it is presented to the user and the execution halts. If the whole query fails, indicating that some of the triples don't have answers in the ontology, the query processor identifies the triple(s) that caused the failure. It attempts to answer this triple from the web by invoking the document search engine.

2.4 Document Search Engine (DSE)

The task of the DSE is to generate multiple keyword sets using different pieces of knowledge to find web documents that may contain the answer(s) to the question using a web search engine. We use Google (www.google.com) as the web search engine in this work. The DSE generates multiple keyword sets by using the instances and the labels of properties and classes included in the triple. This is done in addition to the labels of the classes (types) of the question instances and the label of the expected class of the answer we are looking for as extracted from the triple. Additional keyword sets are generated by using alternatives to the labels using WordNet. Some of the keyword sets are generated for the question above are as follows:

```
"John Smith", Advisor, Student, Professor
"John Smith", Adviser, Student, Prof
```

This component also allows for the user to restrict the documents to be retrieved from a single domain instead of documents from any domain that can be irrelevant to the field. For example, a user looking for bio-chemical process information in the introductory example might want to limit the search to PubMed, to guarantee more relevant results.

2.5 Semantic Answer Extraction and Ranking (SAER)

SAER's task is to extract possible answers to the unanswered triples of the question using the documents the DSE retrieve from the web, and then rank these answers. The SAER utilizes the snippets of web documents that are generated by web search engines to indicate where the search terms are located in the document. In SAER, noun phrases within these snippets are identified by the Stanford Parser and are considered candidate answers to the triple that we are answering. Each noun phrase (NP) is given a score that we call the Semantic Answer Score to determine their relevance to the triple using the following formula.

$$\begin{aligned} Score = (W_{AnswerType} * Distance_{AnswerType}) \\ +(W_{Property} * Distance_{Property}) \\ +(W_{Others} * Distance_{Others}) \end{aligned}$$

This score utilizes knowledge from the ontology to capture the most-likely answer to a question when extracted from web documents. The score is a weighted sum of three different groups of measurements that are explained below. The measurement weights are calibrated based on empirical trials. Please note that when referring to the name of class or a property, we also refer to any of its alternatives as determined by WordNet.

1. $Distance_{\text{AnswerType}}$: During our experiments, we found that if a NP is very close to the expected type (LDS class) of the answer that was a very good indication the NP is a candidate for being an answer for the unanswered triple.

2. $Distance_{\text{Property}}$: Similarly, the distance that separates a NP and the property that was used in the triple also determines the relevance of that NP to the triple.

3. $Distance_{\text{Others}}$: Finally, the distance that separates the NP from all other terms in the keyword set such as the named entities that were mentioned in the question or their types.

3. RESULTS

Effectiveness of SemanticQA in answering factoid questions is initially tested using several ontologies. We here show the results of testing SemanticQA on two of these: SwetoDblp (a large-scale populated ontology on DBLP) [1] and Lehigh University Benchmark(LUBM) [4]. Combined statistics from these test cases results in global instance recall score (IR) of 90% and a global instance precision score (IP) of 69%, showing that our system can deliver promising results [10]. Below is a summary of each of these datasets and an overview of the results obtained using the definitions of precision and recall from [6].

3.1 LUBM

LUBM ontology is widely-used benchmark that describes terms relating a university domain. The ontology was fitted with a small factual dataset that represents several professors, students, and institutions. Eight factoid questions using various properties and classes in that ontology were asked to the system and the answers retrieved were compared to the known correct answers. LUBM has the following statistics:

1. 42 classes

2. 24 relationships

Performing a magnitude of questions using LUBM similar to the ones in Table 1 resulted in a precision of 63% and a recall of 100%.

3.2 SwetoDblp

SwetoDblp is a widelyused and publicly available large scale populated ontology that models on computer science publications such as: proceedings, book chapters, conferences, author, affiliations, and co-editorships. SwetoDblp has the following statistics:

1. 21 classes

2. 28 relationships

3. 2,395,467 instances

4. 11,014,618 triples

We asked the system six sample questions that covered different classes and properties in the ontology. Of the six questions asked using SwetoDblp, the system was able to find the correct answers (in the top five results) five times (IR = 83%) and the correct answer was the first answer in all those times it found the answer five times (IP = 83%)(see Table 2 for sample questions and answers).

4. CONCLUSIONS AND FUTURE WORK

In this paper, SemanticQA is introduced to combine different techniques to provide an easy-to-use interface to answer questions from multiple sources. Although SemanticQA is shown to perform well through our preliminary results, we plan to conduct future tests on larger data sets (DBPedia) and real-world evaluation question sets (TREC).
To improve further SemanticQA, we consider processing web documents as a whole rather than the snippets of these web documents as produced by the web search engine. This can cause elimination of problems that are caused by truncation used by the search engine to produce snippets, and will allow sentences that are better processed by English language parsers to extract possible answers. Still, this will present a challenge of text processing as web documents are frequently filled with content that is irrelevant to its contents such as advertisements and navigational panels in different sections of the document. In addition, we are working on adding the capabilities to answer more complex questions that will require query execution planning and dividing the main query into subqueries in advance to allow faster retrieval of the answers.

5. REFERENCES

[1] B. Aleman-Meza, F. Hakimpour, I.B. Arpinar, and A.P. Sheth. Swetodblp Ontology of Computer Science Publications. *Web Semantics*, 5(3):151–155, 2007.

[2] C. Fellbaum et al. *WordNet: An electronic lexical database*. MIT press Cambridge, MA, 1998.

[3] O. Gospodnetic and E. Hatcher. *Lucene in action*. Manning, 2005.

Table 1: Sample questions and answers from LUBM

Question	Correct Answer (s)	Rank
Who is the advisor of Samir Tartir?	Dr. Budak Arpinar	1
Who is Budak Arpinar the advisor of?	Samir Tartir Bobby McKnight	4 Not Found
Who is head of the Computer Science Department at UGA?	Krys J. Kochut	1

Table 2: Sample questions and answers from SwetoDblp

Question	Correct Answer (s)	Rank
What is the volume of "A Tourists Guide through Treewidth in Acta Cybernetica"?	Volume 11	1
What is the journal name of "A Tourists Guide through Treewidth"?	Acta Cybernetica	1
What university is Amit Sheth at?	Wright State University*	Not Found
What is the ISBN of Database System the Complete Book?	ISBN-10: 0130319953	1

*Amit Sheth recently moved to this university, and Google's indices might have not captured this fact yet.

[4] Y. Guo, Z. Pan, and J. Heflin. LUBM: A benchmark for OWL knowledge base systems. *Web Semantics: Science, Services and Agents on the World Wide Web*, 3(2-3):158–182, 2005.

[5] W. Hersh and R.T. Bhupatiraju. TREC Genomics Track Overview. In *TREC 2003*, pages 14–23, 2003.

[6] M. Li, J.H. Badger, X. Chen, S. Kwong, P. Kearney, and H. Zhang. An information-based sequence distance and its application to whole mitochondrial genome phylogeny. *Bioinformatics*, 17(2):149, 2001.

[7] H.M. Muller, E.E. Kenny, and P.W. Sternberg. Textpresso: an ontology-based information retrieval and extraction system for biological literature. *PLoS Biol*, 2(11):e309, 2004.

[8] D. Rebholz-Schuhmann, H. Kirsch, M. Arregui, S. Gaudan, M. Riethoven, and P. Stoehr. EBIMed–text crunching to gather facts for proteins from Medline. *Bioinformatics*, 23(2):e237, 2007.

[9] S. Tartir. *Ontology-Driven Question Answering and Ontology Quality Evaluation*. PhD thesis, University of Georgia, 2009.

[10] S. Tartir, B. McKnight, and I.B. Arpinar. SemanticQA: web-based ontology-driven question answering. In *Proceedings of the 2009 ACM symposium on Applied Computing*, pages 1275–1276. ACM, 2009.

[11] C. Wang, M. Xiong, Q. Zhou, and Y. Yu. Panto: A portable natural language interface to ontologies. In *ESWC '07: Proceedings of the 4th European conference on The Semantic Web*, pages 473–487, Berlin, Heidelberg, 2007. Springer-Verlag.