A Semantic Question Answering Framework

Harsh Shah, Mohneesh Khaneja, Yash Jain,
Anupam Jaiswal, Anujraaj Goyal

Instructor: Prof. U.S. Tiwary

1 Introduction

Accessing online resources often requires the support from advanced information retrieval technologies to produce expected information. This brings new challenges to the construction of information retrieval systems such as search engines and question answering (QA) systems. Given an input query expressed in a keyword-based mechanism, most search engines return a long list of title and short snippet pairs ranked by their relevance to the input query. Then the user has to scan the list to get the expected information, so this is a time consuming task. Unlike search engines, QA systems directly produce an exact answer to an input question. In addition, QA systems allow to specify the input question in natural language rather than as keywords.

Ontological Question Answering systems propose to attack the problem by means of an internal unambiguous knowledge representation. As any knowledge intensive application, ontological QA systems have as intrinsic limitation related to the small scale of the underlying syntactic-semantic models of natural language. We are investigating an approach to ontology-based QA in which users ask questions in natural language to knowledge bases of facts extracted from a BBC news dataset.

The question analysis component uses a knowledge base of grammar rules for analyzing input questions and the answer retrieval component is responsible for interpreting the input questions with respect to a target ontology. The association between the two components is an intermediate representation element which captures the semantic structure of any input question. This intermediate element contains properties of the input question including question structure, question category, keywords and semantic constraints between the keywords.

2 Motivation

Despite the IR-based approaches to QA, there is no easy way to perform a federated search over both structured databases and unstructured text documents, including articles, manuals, reports, emails, blogs, and others. There is no easy way to enable more intelligent applications over such diverse data sources without considerable time and effort spent in system and data model customization by experts.

With the recent emergence of commercial grade Resource Description Framework (RDF) [1] triple stores, it becomes possible to merge massive amounts of structured and unstructured data by defining a common ontology model for the DBMS schemas and representing the structured content as semantic triples. However, technology gaps exist. More specifically, there are no efficient and accurate algorithms and tools to transform unstructured document content into a rich and complete semantic representation that is compatible with the RDF standard. There are no methods for accessing information to enable intelligent applications while hiding the underlying complexity of the voluminous semantic data being searched.

3 Related Work

In a recent survey [2], Bouziane et al. divide QA systems into:

1. QA for web of documents and text that follow three main distinct subtasks: Question Analysis, Document Retrieval, and Answer Extraction [3] to process natural language questions and retrieve precise answers from textual documents.

2. QA for web of data that apply Named Entity Recognition, Syntactic Parsing, Question Classification, and SPARQL Generation on natural language questions and retrieve precise answers from Linked Data.

Development of QA for web of documents and text has been the center of research in the IR and NLP communities for several decades. Such QA systems have been developed, hardened, and evaluated under several government funded programs including National Institute of Standards and Technology (NIST) TREC QA competition from 1999 until 2007 [4]. These QA systems rely on shallow, named entity based indexing to retrieve a small set of candidate answer documents from large collections.

The candidate answer documents undergo deep semantic analysis in a post-processing phase to retrieve the final answers. The TREC QA competitions have revealed that the systems perform very well on processing and retrieving precise answers for factoid questions that mainly query for date, time, location, person, and organization named entities but do not perform well on list and definition questions [5][6]. The lack of semantic knowledge being indexed and queried in the document retrieval phase results in low coverage of answer candidate sentences/documents for further analysis and processing, and thus leads to non optimal performance on certain types of questions.

QA for web of data has lately drawn the attention of many researchers and has resulted in the development of several QA systems for Linked Data such as Aqualog [7], PowerAqua [8], NLP-Reduce [9] and FREyA [10]. The advent of several competitions in this arena including the Open Challenge on Question Answering over Linked Data [11] has helped in data sharing and development of robust systems. Most approaches to QA for web of data use dependency or syntactic parsing to extract and represent a question's semantics as a set of triples and then build a SPARQL query.

The solutions differentiate themselves mainly on:

1. The linguistic processing tools and their performance.

2. Usage of knowledge bases, and

3. The ease of adaptation to newer domains.

Unger et al. [12] focused on applying aggregation and filter constructs to resolve SPARQL query generation related issues. The authors proposed a template-based approach to SPARQL query generation and handle constructs that are not captured using semantic triple representation. The SPARQL templates specify the query's select clause, its filter and aggregation functions, as well as the number and form of the semantic triples.

4 Aim

In this project, we aim to perform following tasks:

1. Transform unstructured data into a structured format.
2. Merge it with other ontologies and structured data into a consolidated RDF store, and
3. Offer a natural language QA interface for easy use.

In the next section, we describe an approach to perform above tasks.

5 Proposed Approach

Our semantic question answering (SQA) framework pinpoints the semantics of both the document collection as well as the user's input question. In the Document Indexing phase, the semantic information derived from document content is represented in an RDF format that facilitates its storage into a semantic triple store. At query time, the user's natural language question is parsed to identify its meaning. This is then automatically converted into a SPARQL query that will be used to retrieve precise answers from the already populated RDF store.

- In this section, we detail the novel steps of the document indexing phase of our proposed SQA framework. More specifically, we present the RDF representation of the input document content as well as the various types of entailment rules that can be used on an RDF store to generate additional triples. Having extracted various semantic knowledge from the input documents and therefore, having created a more structured dataset from the unstructured input text, we define a robust RDF representation, which when translated into triples, can be stored within an RDF semantic index.
- This store can then be accessed, visualized, queried, or integrated with already available structured data. We note that, in addition to the RDF representation of the input document collection, the RDF store may contain triples that define a domain ontology. For these knowledge resources, we store any concept information that they provide (part-of-speech, sense number, named entity class, if available) as well as any semantic relationships identified between the ontological concepts.

- The available knowledge extracted from the document content includes: (1) lexical knowledge (sentence boundaries, token information, including part-of-speech tag, start and end positions, and lemma), (2) syntactic information (head-of-parse-phrase flags for tokens, and syntactic phrase dependencies), as well as (3) semantic knowledge (named entity labels, WordNet word senses, coreference chains, and semantic relations).

- The semantic information is the most valuable to an end consumer of the knowledge conveyed by the input text. Therefore, we reduce the set of linguistic knowledge translated to RDF to include: (1) only concepts that participate in semantic relationships, and (2) the semantic relations linking these concepts. More specifically, for named entities, we store only their entity type, lemma, synset information (if available), and reference sentence.

- The question answering phase of the QA process must generate SPARQL queries semantically equivalent to input questions and use these queries to interrogate the RDF store. In order to ensure the system's robustness, several query relaxation procedures to be used when no answers are found in the RDF store will be implemented.

- For a given question, the answer type and answer type term information is used to decide which SPARQL variables are to be SELECT ed and returned by the query. Furthermore, the set of semantic relations identified within the question text describe the SPARQL query's WHERE constraints – the triple patterns that must be satisfied by the retrieved answers

- After the SPARQL query has been generated corresponding to a natural language question, it is possible that system will return more than one possible answer. Thus, an answer set ranking procedure has to be developed before the results are presented to the user. The ordering of the answers to be returned will be done at query time by the RDF store's internal retrieval engine by including an ORDER BY clause in generated SPARQL queries. The value used to sort the returned answers is the confidence of the semantic relation that links the answer concept to the rest of the question.

6 Activity Chart

Task was divided as follows :-

	Mid Sem		End Sem	
	Phase 1	Phase-II	Phase-III	Phase-IV
	5 Jan-5 Feb	6 Feb-28 Feb	15 Mar-5 Apr	6 Apr-28 Apr
Literature Survey	Done			
Problem Identification	Done			
Identify an Approach		Done		
RDF Generation		Done		
Storing RDF				
SPARQL Generation				
Creating Framework				
Evaluation				

7 Technologies Used

1. RDF(Resource description Framework) : The RDF data model is similar to classical conceptual modeling approaches (such as entity–relationship or class diagrams). It is based upon the idea of making statements about resources expressions, known as triples. Triples are so named because they follow a subject–predicate–object structure. The subject denotes the resource, and the predicate denotes traits or aspects of the resource, and expresses a relationship between the subject and the object.

   For example, one way to represent the notion ”The sky has the color blue” in RDF is as the triple: a subject denoting ”the sky”, a predicate denoting ”has the color”, and an object denoting ”blue”.

2. Ontology: A model for describing the world that consists of a set of types, properties, and relationship types.

3. SPARQL (Sparql Protocol and RDF Query Language) : SPARQL is a semantic query language for databases, able to retrieve and manipulate data stored in Resource Description Framework (RDF) format. It allows for a query to consist of triple patterns, conjunctions, disjunctions, and optional patterns.

4. NLTK (Natural Langauge Toolkit) : NLTK is a suite of libraries and programs for symbolic and statistical natural language processing (NLP) for English written in the Python programming language.

5. STANFORD CoreNLP Toolkit : Stanford CoreNLP toolkit is an extensible pipeline that provides core natural language analysis. This toolkit is quite widely used, both in the research NLP community and also among commercial and government users of open source NLP technology.

References

[1] RDF Working Group, “Resource Description Framework (RDF),” http://www.w3.org/RDF/, 2014.

[2] A. Bouziane, D. Bouchiha, N. Doumi, and M. Malki, “Question Answering Systems: Survey and Trends,” Procedia Computer Science, vol. 73, pp. 366 – 375, 2015.

[3] V. Lopez, V. Uren, M. Sabou, and E. Motta, “Is Question Answering Fit for the Semantic Web?: A Survey,” Semantic Web, vol. 2, no. 2, pp. 125–155, Apr. 2011.

[4] H. T. Dang, D. Kelly, and J. Lin, “Overview of the TREC 2007 Question Answering Track,” in Proceedings of The Sixteenth Text REtrieval Conference, 2008.

[5] D. Moldovan, C. Clark, and M. Bowden, “Lymba’s PowerAnswer 4 in TREC 2007,” in Proceedings of Text REtrieval Conference, 2007.

[6] D. Moldovan, M. Bowden, and M. Tatu, “A Temporally-Enhanced PowerAnswer in TREC 2006,” in Proceedings of Text REtrieval Conference, 2006.

[7] V. Lopez, V. Uren, E. Motta, and M. Pasin, “AquaLog: An Ontology-driven Question Answering System for Organizational Semantic Intranets,” Web Semantics: Science, Services and Agents on the World Wide Web, vol. 5, no. 2, 2007.

[8] V. Lopez, M. Fernndez, 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.

[9] E. Kaufmann, “Talking to the Semantic Web? Natural Language Query Interfaces for Casual End-users,” Ph.D. dissertation, University of Zurich, February 2009. [Online]. Available: http://www.ifi.uzh.ch/pax/web/uploads/pdf/publication/ 1202/Dissertation Esther Kaufmann.pdf

[10] D. Damljanovic, M. Agatonovic, and H. Cunningham, “FREyA: An Interactive Way of Querying Linked Data Using Natural Language,” in Proceedings of 8th Extended Semantic Web Conference, 2012, pp. 125–138.

[11] C. Unger, C. Forascu, V. Lopez, A.-C. N. Ngomo, E. Cabrio, P. Cimiano, and S. Walter, “Question Answering over Linked Data (QALD- 5),” in Cross-Language Evaluation Forum CLEF (Working Notes), ser. CEUR Workshop Proceedings, L. Cappellato, N. Ferro, G. J. F. Jones, and E. SanJuan, Eds., vol. 1391, 2015.

[12] C. Unger, L. Buhmann, J. Lehmann, A.-C. Ngonga Ngomo, D. Gerber, and P. Cimiano, “Template-based Question Answering over RDF Data,” in Proceedings of WWW ’12, 2012, pp. 639– 648.

8 Suggestion by Board members