Querying Large Linked Data Resources

Zareen Syed[1], Lushan Han[1], Muhammad Rahman[1],
Tim Finin[1], James Kukla[2], Jeehye Yun[2]

[1]University of Maryland, Baltimore County
1000 Hilltop Circle, Baltimore, MD, USA 21250
{zsyed, lushan1, mrahman1, finin}@umbc.edu

[2]RedShred, 5520 Research Park Drive, Baltimore, MD 21228
jkukla@redshred.net, jyun@redshred.net

Abstract. Exploring large complex linked data resources is challenging as it requires not only mastering SPARQL syntax and semantics but also understanding the RDF data model and large ontology vocabularies comprising of thousands of classes, hundreds of properties and millions of URIs for instances of interest. Natural language question answering systems solve the problem, but these are still subjects of research. We describe a compromise in which non-experts specify a graphical query 'skeleton' and annotate it with freely chosen words, phrases and entity names. Our system automatically generates a SPARQL query based on the input query skeleton.

Keywords: Information Storage and Retrieval, User Interfaces, Semantic Web

1 Introduction

We describe a new schema-free query (SFQ) interface, in which the user explicitly specifies the relational structure of the query as a graphical "skeleton" and annotates it with freely chosen words, phrases and entity names. Our framework makes three main contributions. It uses robust methods that combine statistical association and semantic similarity to map user terms to the most appropriate classes and properties used in the underlying ontology. Second, it uses a novel type inference approach based on concept linking for predicting classes for subjects and objects in the query. Third, it implements a general property mapping algorithm based on concept linking and semantic text similarity. We briefly describe an evaluation in the Schema-agnostic Queries over Large-schema Databases challenge [9], and directions for future work.

2 Approach

2.1 Semantic Similarity

We need to compute semantic similarity between user entered query terms and terms in the target ontology. Our approach [3,4] for computing semantic similarity com-

bines part of speech tagging, LSA word similarity and WordNet knowledge along with custom term alignment algorithms. Our system was ranked as the top performing system in 2013 and 2014 SemEval Conference challenge tasks [1].

2.2 Type Inference

Our main SFQ system [2] requires users to provide types or classes for subjects and objects in the query triples, however, this information is not available in many challenge queries. For the input query, we infer concept types using concept linking approach based on Wikitology [6,8] and Wikipedia Miner [5]. After linking the subject and object to concepts in Wikipedia [7] we retrieve the associated DBpedia ontology classes to represent concept types.

2.3 Concept level Association Knowledge Model (CAK Model)

We employ a computational semantic similarity measure for the purpose of locating candidate ontology terms for user input terms. Semantic similarity measures enable our system to have a broader linguistic coverage than that offered by synonym expansion. We know birds can fly but trees cannot and that a database table is not kitchen table. Such knowledge is essential for human language understanding. We refer to this as Concept level Association Knowledge (CAK). Domain and range definitions for properties in ontologies, argument constraint definitions of predicates in logic systems and schemata in databases all belong to this knowledge. Manually defining this knowledge is tedious, we therefore, learn Concept-level Association Knowledge statistically from instance data (the "ABOX" of RDF triples) and compute degree of associations between terms in the ontology based on co-occurrences. We count co-occurrences between schema terms indirectly from co-occurrences between entities because entities are associated with types. We then apply a statistical measure, Pointwise Mutual Information (PMI), to compute degree of associations between classes and properties and between two classes. The detailed approach is available in [2]. We employ the learned CAK and semantic similarity measures for mapping a user query to a corresponding SPARQL query which we discuss in the next sections.

2.4 Query Interpretation

For each SFQ concept or relation, we generate a list of the k most semantically similar candidate ontology classes or properties. In the example in Figure 1, candidate lists are generated for the five user terms in the SFQ, which asks "Which author wrote the book Tom Sawyer and where was he born?". Candidate terms are ranked by their similarity scores, which are displayed to the right of the terms. Each combination of ontology terms, with one term coming from each candidate list, is a potential query interpretation, but some are reasonable and others not. We use a linear combination of three pairwise associations to rank interpretations. The three are (i) the directed association from subject class to property, (ii) the directed association from property to object class and (iii) the undirected association between subject class and object class, all weighted by semantic similarities between ontology terms and their corresponding user terms. After user terms are disambiguated and mapped to appropriate ontology

terms, translating a SFQ to SPARQL is straightforward. Classes are used to type the instances, properties used to connect instances. Our system generates a ranked list of SPARQL queries.

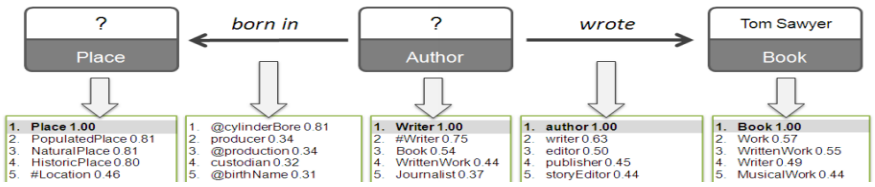

Fig. 1. A ranked list of candidate ontology terms

2.5 System II

Since our original SFQ system relies on CAK model which is based on DBpedia ontology classes and properties and does not take instance references into account, we created an independent parallel system to support instance references in SPARQL query. The system is based on concept linking and semantic similarity. For any concepts mentioned in the query, we try to link it to DBpedia using Wikitology and Wikipedia Miner and update the reference to the linked concept in DBpedia. For mapping properties, we retrieve all associated DBpedia properties for the linked concepts and compute semantic similarity with the property input by the user and select the property with the highest similarity with the user input property.

Table 1. Evaluation Results of independent and combined systems for SAQ-2015 challenge

	SFQ System	**System II**	**SFQ System + System II**
Avg. precision	0.27	0.22	0.33
Avg. recall	0.27	0.24	0.36
Avg. f1-measure	0.24	0.21	0.31
# of queries answered	34	30	45
% of queries answered	33%	29%	44%

3 Evaluation and Discussion

For evaluation we combined the output of both systems i.e. SFQ System and System II where System II addresses the queries related to instances for type constraints and SFQ System addresses the queries related to ontology classes for type constraints. Our combined system was awarded schema agnostic query challenge award in the Schema Agnostic Queries SAQ-2015 challenge competition. The evaluation dataset for the task had 103 queries in total. Table 1 presents the evaluation results for two systems independently and in combination. We analyzed the incorrect queries and found different sources of errors such as errors in type inference, concept linking and errors due to fewer or more number of triples generated compared to gold standard query.

The challenge queries were based on DBpedia 2014 whereas, our CAK model was trained on DBpedia 3.6, we believe that training the SFQ System on the newer DBpedia version may have improved the performance of the system. We also observed a number of cases in challenge queries which referenced instances for type constraints instead of ontology classes. The queries generated by our SFQ system only reference ontology classes for type constraints. System II addresses this issue by resolving links to instances. However, it cannot deal with cases where the number of relations or triples may vary between the user input query and the correct translated SPARQL query. In the future, we plan to improve our approach by developing a unified system that would incorporate the strengths of both systems.

4 Conclusions

The schema-free structured query approach allows people to query the DBpedia dataset without mastering SPARQL or acquiring detailed knowledge of the classes, properties and individuals in the underlying ontologies and the URIs that denote them. Our system uses statistical data about lexical semantics and RDF datasets to generate plausible SPARQL queries that are semantically close to schema-free queries. The key contributions of our approach are the robust methods that combine statistical association and semantic similarity to map user terms to the most appropriate classes and properties used in the underlying ontology and type inference for user input concepts based on concept linking.

5 References

1. Agirre, E., Cer, D., Diab, M., Gonzalez-Agirre, A., and Guo, W.: *SEM 2013 shared task: Semantic textual similarity, including a pilot on typed-similarity, 2nd Joint Conf. on Lexical and Computational Semantics, Association for Computational Linguistics. 2013.
2. Han, L., Finin, T. and Joshi, A.: Schema-free Structured Querying of DBpedia Data, In Proc. 21st ACM Int. Conf. on Information and Knowledge Management, pp. 2090-2093. ACM, 2012.
3. Han, L., Finin, T., McNamee, P., Joshi, A. and Yesha, Y.: Improving Word Similarity by Augmenting PMI with Estimates of Word Polysemy, IEEE Transactions on Knowledge and Data Engineering, IEEE Computer Society, v25n6, pp. 1307-1322, 2013.
4. Han, L.: Schema Free Querying of Semantic Data, Ph.D. Dissertation, Univ. of Maryland, Baltimore County, Aug. 2014.
5. Han, L., Finin, T., Joshi, A. and Cheng, D.: Querying RDF Data with Text Annotated Graphs, 27th Int. Conf. on Scientific and Statistical Database Management, June 2015.
6. Kashyap, A., Han, L., Yus, R., Sleeman, J., Satyapanich, T., Gandhi, S. and Finin, T.: Meerkat Mafia: Multilingual and Cross-Level Semantic Textual Similarity systems, Proc. 8th Int. Workshop on Semantic Evaluation, August 2014
7. Milne, D. and Witten, I. H.: Learning to Link with Wikipedia. In Proceedings of the 17th ACM conference on Information and knowledge management, pp. 509-518. ACM, 2008.
8. Syed, Z. and Finin, T.: Creating and Exploiting a Hybrid Knowledge Base for Linked Data, in Agents and Artificial Intelligence, Revised Selected Papers Series: Communications in Computer and Information Science, v129, Springer, April 2011.
9. https://sites.google.com/site/eswcsaq2015/