Hybrid Graph based Keyword Query Interpretation on RDF

Kaifeng Xu[1], Junquan Chen[1], Haofen Wang[1], and Yong Yu[1]

Apex Data and Knowledge Management Lab
Shanghai Jiao Tong University, Shanghai, 200240, China
{kaifengxu,jqchen,whfcarter,yyu}@apex.sjtu.edu.cn

Abstract. Adopting keyword query interface to semantic search on RDF data can help users keep away from learning the SPARQL query syntax and understanding the complex and fast evolving data schema. The existing approaches are divided into two categories: instance-based approaches and schema-based approaches. The instance-based approaches relying on the original RDF graph can generate precise answers but take a long processing time. In contrast, the schema-based approaches relying on the reduced summary graph require much less processing time but cannot always generate correct answers. In this paper, we propose a novel approach based on a hybrid graph which can achieve significant improvements on processing time with a limited accuracy drop compared with instance-based approaches, and meanwhile, can achieve promising accuracy gains at an affordable time cost compared with schema-based approaches.

1 Introduction

On the way to Semantic Web, Resource Description Framework (RDF) is a language for representing information about resources in the World Wide Web. The ever growing semantic data in RDF format provides fertile soil for semantic search, and formal query languages (e.g. SPARQL) are adopted by most current semantic search systems[1, 2] to accurately express complex information needs. However, the disadvantages of formal queries are: (1) *Complex Syntax*: It is hard to learn and remember complex syntax of formal queries for ordinary users. (2) *Priori Knowledge*: Users have to know the schema of the underlying semantic data beforehand. In contrast, keyword queries cater to user habits since keywords (or known as keyword phrases) are easier to be understood and convenient to use. An approach that can leverage the advantages of both query types is to provide a keyword user interface and then translate keyword queries into formal queries.

In XML and database communities, bridging the gap between keyword queries and formal queries has been widely studied. However, there exists a limited amount of work on how to answer keyword queries on semantic data in RDF format. As an early attempt to build a semantic search system, SemSearch [3] employed a template-based approach to capture the restricted interpretations

of given keywords. Later, improved approaches [4, 5] have been proposed to address the problem of finding all possible interpretations. In particular, Thanh et al. [5] employed the RDF graph (*instance-based approaches*) to discover the connections between nodes matching the input keywords, through which the interpretation accuracy can be ensured, but at the cost of a longer processing time. This problem has been recently tackled by [6, 7], where keyword queries are translated using a summary graph extracted from the RDF data (*schema-based approaches*). Although schema-based approaches significantly speed up the processing, the schema-graph loses too much connectivity information of the corresponding RDF graph to guarantee the interpretation accuracy.

In this paper, we propose a novel effective and efficient keyword query interpretation approach based on a hybrid graph carefully constructed from the original RDF graph. A hybrid graph is much smaller than the original RDF graph, and meanwhile it can preserve as much connectivity information as possible. In this way, we construct the hybrid graph under the guidance of a graph score which reflects the best tradeoff between effectiveness and efficiency of keyword query interpretation.

2 The Hybrid Graph based Approach

Figure 1 describes the entire process (both offline and online stages) of keyword query interpretation. In the offline stage, a hybrid graph is constructed from the origin RDF graph. After that, a keyword index is built for the mapping of keywords to corresponding nodes in the hybrid graph. During the online process, keywords input by end users are first mapped to the nodes in the hybrid graph using the keyword index, and then we search on the hybrid graph to construct top-k potential tree-shaped conjunctive queries (i.e., formal queries). While our focus is the construction of the hybrid graph, we implement a similar keyword mapping, query construction and ranking as mentioned in Q2Semantic [6]. Due to space limitations, we refer you to [6] for details.

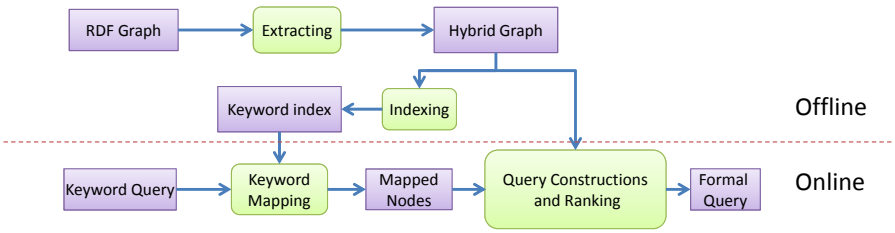

Fig. 1. The flow chart of keyword query interpretation

The construction of hybrid graph is an iterative process which extracts and refines a qualified subgraph from the original RDF graph by means of a graph

score. The graph score is used to define the overall interpretation performance, which plays an important role from the starting point to the ending point of each interaction in the whole construction process. More precisely, the graph score is the linear combination of the size of the hybrid graph and the amount of the connectivity information contained in the graph. The workflow of a hybrid graph construction is illustrated in Figure 2(a) which takes the original RDF graph as input. A *construction unit* (CU) is employed to carry out the refinement on the given RDF graph to generate a qualified hybrid subgraph. The CUs are additionally used several times for further refinement on the remaining unqualified RDF subgraphs. Finally, several qualified hybrid subgraphs are returned, and combined together to form an overall hybrid graph.

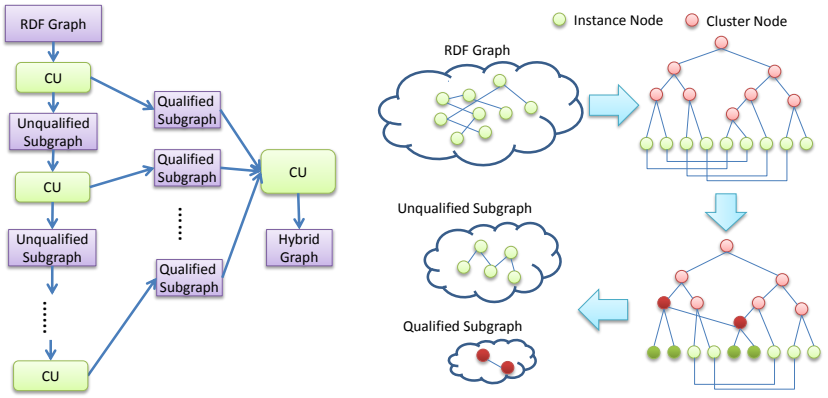

Fig. 2. (a) The work flow of hybrid graph construction. (b)The work flow of CU.

Figure 2(b) shows the detailed components inside a CU. Given a RDF graph G, CU will work through the following streamline: (1) *Instance Clustering*: This phase aims to generate the nodes of hybrid graph. Each instance in G is represented by the feature set of its relations and regarded as a trivial cluster initially. After that, the cluster pairs with the highest similarity are hierarchically clustered until a cluster tree T is derived. (2) *Relation Refinement*: This phase aims to generate the relations of hybrid graph. It tries to substitute the relations in the given RDF graph with those between the high level nodes in the cluster tree. The relation replacement will increase the score of the graph, and the goal is to get a hybrid graph with the highest score. (3) *Graph Detachment* The score of the whole graph generated after relation refinement might be too low. Thus, we extract a part of the graph (called a qualified subgraph) whose score is above the given score threshold, and feed the remaining part (called an unqualified subgraph) into the CU for next iterations.

3 Preliminary Results

We compare our approach with instance-based approches and schema-based counterparts on three different datasets (i.e., semanticweb.org, DBpedia, DBLP) in terms of processing time and interpretation accuracy. Table 1 lists The statistics of the three data sets. We manually construct 42 scenarios (17 from semanticweb.org, 10 from DBpedia, and 5 from DBLP) for the comparison.

Table 1. Statistics of semanticweb.org, DBLP and DBpedia.

Data set	♯Category	♯Instance	♯Relation	♯Inst.degree	♯Rel.kind	♯Rel/kind
semanticweb.org	5.06×10^2	7.483×10^3	1.628×10^4	2.18	4.77×10^2	3.413×10^1
DBLP	1.0×10^1	1.640×10^6	3.176×10^6	1.94	1.0×10^1	3.176×10^5
DBpedia	2.694×10^5	2.520×10^6	6.868×10^6	2.73	1.128×10^4	6.088×10^2

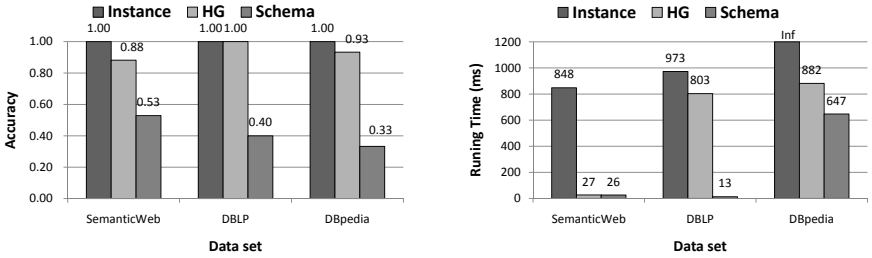

Fig. 3. The practical interpretation accuracy and efficiency on different data sets

Figure 3 shows that our approach has achieved a 61.66% efficiency improvement with a 6.17% accuracy drop over instance-based approaches on average. On the other hand, our approach achieves a 132.30% accuracy gain with a 20.08% time increase compared with schema-based approaches.

References

1. Broekstra, J., Kampman, A., Van Harmelen, F.: Sesame: A generic architecture for storing and querying rdf and rdf schema. In: ISWC. (2002) 54–68
2. Lu, J., Ma, L., Zhang, L., Brunner, J., Wang, C., Pan, Y., Yu, Y.: SOR: a practical system for ontology storage, reasoning and search. In: VLDB. (2007) 1402–1405
3. Lei, Y., Uren, V., Motta, E.: Semsearch: A search engine for the semantic web. In: EKAW. (2006) 238
4. Zhou, Q., Wang, C., Xiong, M., Wang, H., Yu, Y.: Spark: Adapting keyword query to semantic search. In: ISWC/ASWC. (2007) 694
5. Tran, T., Cimiano, P., Rudolph, S., Studer, R.: Ontology-based interpretation of keywords for semantic search. In: ISWC/ASWC. (2007) 523
6. Wang, H., Zhang, K., Liu, Q., Tran, T., Yu, Y.: Q2Semantic: A lightweight keyword interface to semantic search. In: ESWC. (2008) 584
7. Tran, T., Wang, H., Rudolph, S., Cimiano, P.: Top-k exploration of query candidates for efficient keyword search on graph-shaped (RDF) data. In: ICDE. (2009) 405–416