COLINA: A Method for Ranking SPARQL Query Results through Content and Link Analysis

Azam Feyznia, Mohsen Kahani, Fattane Zarrinkalam

Computer Engineering Department, Ferdowsi University of Mashhad, Mashhad, Iran

azam.feyznia@stu-mail.um.ac.ir
kahani@um.ac.ir
fattane.zarrinkalam@stu-mail.um.ac.ir

Abstract. The growing amount of Linked Data increases the importance of semantic search engines for retrieving information. Users often examine the first few results among all returned results. Therefore, using an appropriate ranking algorithm has a great effect on user satisfaction. To the best of our knowledge, all previous methods for ranking SPARQL query results are based on popularity calculation and currently there isn't any method for calculating the relevance of results with SPARQL query. However, the proposed ranking method of this paper calculates both relevancy and popularity ranks for SPARQL query results through content and link analysis respectively. It calculates the popularity rank by generalizing PageRank method on a graph with two layers, data sources and semantic documents. It also assigns weights automatically to different semantic links. Further, the relevancy rank is based on the relevance of semantic documents with SPARQL query.

Keywords: Ranking, SPARQL, Semantic Search Engine, Link Analysis, Content Analysis, Semantic Web

1 Introduction

Structured data has enabled users to search semantic web, based on SPARQL queries. The increasing amount of structured data on the web has led to many results to be returned by a SPARQL query [1]. Further, since in most cases, all returned results equally satisfy query conditions, checking all of them and finding the best answers takes too much time. Therefore, the semantic web search engines whose have provided a SPARQL endpoint for processing and running SPARQL queries on their indexed data, require some mechanisms for ranking SPARQL query results besides the ranking methods applied to keyword queries, to help users find their desired answers in less time.

In search engines, ranking are usually done by content and link analysis and the final rank for each result is calculated by combining scores obtained from each analysis algorithm [2-3]. The content analysis ranking algorithms calculate the relevancy between each result with the user query in online mode. In the link analysis ranking algorithms,

popularity calculation is done in offline mode, before the user query is received, by constructing data graph and analyzing the existing links in it.

To the best of our knowledge, all previous methods for ranking SPARQL query results are based on popularity calculation and currently there is no method for calculating the relevance of sub-graph results with SPARQL query. The ranking methods which are based on link analysis, compute rank for entities of result graphs by utilizing entity-centric data models. It is worth noting that, the results of a *SPARQL* query in addition to the entities, may be made up of predicates and constant values. As a result, the proposed algorithms by [4] and [5] which are only based on entity ranking, cannot rank all results of *SPARQL* queries. One of the cornerstones in ranking SPARQL query results are language model based ranking methods [6]. Providing an approach for analyzing content of structured queries such as SPARQL queries, is a significant advance which is obtained by these methods.

Therefore, by studying the limitations presented in existing researches and considering specific features of SPARQL queries and results, this paper proposed a ranking method which calculates relevancy and popularity scores through content and link analysis respectively.

2 Proposed Method: COLINA

We are interested in measuring how valuable the result graph is, for a given query. Our method ranks SPARQL query results by combining content and link analysis scores of semantic documents which results are retrieved from. In the next subsections, we briefly describe two key components of our method.

2.1 Offline Ranker

The offline ranker calculates data popularity by applying weighted PageRank algorithm on data graph. We first explain our data model and then reveal our scheme for weighting semantic links.

Data Model. In order to consider the provenance of data in our link analysis ranking, we choose a two-layer graph including data source and semantic document layers. Data source layer is made up of a collection of inter-connected data sources. A data source is the source which has authority to assign URI identifier and is defined as a pay-level domain similar to [3]. The semantic document layer is composed of independent graphs of semantic documents. Each graph contains a set of internal nodes and edges.

Our explanation for using document-centric data model instead of entity-centric data model is that in response to a SPARQL query, the sub-graphs that meet query conditions are returned as results. Depending on the number of triple patterns in query, each sub-graph constitutes several triples. Hence, we can estimate the rank score of triples by the rank score of documents which are appeared in them. The document graph was constructed by extracting explicit and implicit links between semantic documents according to [7].

Weighting mechanism. We categorized links in two classes based on their labels, but not their frequency: specific and general links. In semantic web, links are semantically different and so they have different importance. Our method for measuring the importance of link labels goes beyond just measuring the frequency of labels by also taking these categories into account. We first determine which category the link label belongs to, then we use different frequency based measurements. The intuition behind this idea is that general and common link labels such as owl:sameAs, which convey high importance, get high weight. On the other hand, specific link labels, that hold much information based on information-theory, get high weight too. This way we can consider the importance of common link labels and also maintain the importance of specific link labels. In this paper, we exploit a hierarchical approach to separate the link labels that are between data sources. From this point of view, the link label that is defined for a particular class is considered general for all of its subclasses. Hence each data source is a subclass of owl:thing, we can derive general labels through extracting link labels which rdfs:domain of them is defined owl:thing by Virtuoso[1].

2.2 Online Ranker

Unlike keyword-based queries which are collection of words that are specified by users, each triple pattern in SPARQL queries has two arguments: the bound arguments which are labeled by users and the unbound arguments which are variable. We can measure the relevancy of document, based on bound and unbound query arguments as follows:

$$S_q(doc) = \beta\, r_q(doc) + (1-\beta)\, r_r(doc) \tag{1}$$

where $r_q(doc)$ and $r_r(doc)$ denotes the relevancy score of a document with respect to unlabeled arguments in query and produced answer, respectively. Parameter β set empirically to a calibrated global value.

For example, assuming that "*Bob a Physicist"* is an answer for "*?x a Physicist"*. If this triple appears in a document which is exclusively about physicist or Bob, it is more relevant than when it is included in a document which is about anything else. This example highlights our justification for using both bound and unbound arguments in the relevance calculation for documents.

Since the computing value for $r_q(doc)$ and $r_r(doc)$ depends on query formulation, we need to deal with possible forms of triple patterns. For this, we define ACDT and QCDT functions for estimating $r_q(doc)$ and $r_r(doc)$ respectively.

The ACDT is Answer Container Document's Triples. In short, it computes frequency of a result in semantic documents with respect to the position of unbound arguments in intended triple pattern. The QCDT is Query Container Document's Triples. Similarly, it computes frequency of a query in semantic documents with respect to the position of bound arguments in intended triple pattern. The basic idea for ACDT and QCDT is derived from TF Scheme in information retrieval.

---

[1] http://lod.openlinksw.com/sparql

3 Combine Content and Link Analysis Ranks

We combine relevancy score S_r and popularity score S_p in order to compute final score s_f for document. Since the foundation of our ranking algorithms is similar to algorithms presented in [2], we use his method for combining scores of our algorithms.

4 Conclusion

In this paper we presented a method for ranking SPARQL query results based on content and link analysis, which can be used as ranking component in semantic web search engines. In our method, the rank of triples that constitute the result graphs are approximated by the rank score of semantic documents which expressed them. We introduced a two-layer data model and proposed a novel link weighting mechanism based on separation of link labels incorporating the notion frequency of labels in a convenient manner. Our content analysis ranking algorithm provides an approach to compute the relevancy of results with respect to the bound and unbound arguments in intended SPARQL query. We believe that using content analysis ranking in combination with link analysis ranking which is powered by our data model and weighting mechanism, can improve accuracy of ranking algorithm for SPARQL query results.

References

1. J. Hees, M. Khamis, R. Biedert, S. Abdennadher, and A. Dengel. Collecting links between entities ranked by human association strengths. *In Proceedings of ESWC-13*, pages 517-531, 2013.
2. R. Delbru. Searching Web Data: an Entity Retrieval Model. *Ph.D. Thesis*, *National University of Ireland*, Ireland, 2010.
3. A. Hogan, A. Harth, J. Umbrich, S. Kinsella, A. Polleres, S. Decker. Searching and Browsing Linked Data with SWSE: the Semantic Web Search Engine. *Journal of web semantics,* pages 365-401, 2011.
4. K. Mulay, P.S. Kumar. SPRING: Ranking the results of SPARQL queries on Linked Data. *17th International Conference on Management of Data COMAD*, Bangalore, India, 2011.
5. A. Buikstra, H. Neth, L. Schooler, A. ten Teije, F. van. Harmelen. Ranking query results from linked open data using a simple cognitive heuristic. *In Workshop on discovering meaning on the go in large heterogeneous data 2011 (LHD-11), Twenty-second International Joint Conference on Articial Intelligence (IJCAI-11)*, Barcelona, Spain, 2011.
6. G. Kasneci, F. M. Suchanek, G. Ifrim, M. Ramanath, G. Weikum. NAGA: Searching and Ranking Knowledge. *In 24th International Conference on Data Engineering (ICDE 2008). IEEE*, 2008.
7. A. Feyznia, M. Kahani, R. Ramezani. A Link Analysis Based Ranking Algorithm for Semantic Web Documents. *In 6th Conference on Information and Knowledge (IKT 2014),* Shahrood, Iran, 2014.