Random Indexing for Searching Large RDF Graphs

Danica Damljanovic, Johann Petrak, and Hamish Cunningham

Department of Computer Science, University of Sheffield
Regent Court, 211 Portobello Street, Sheffield, UK
d.damljanovic, j.petrak, h.cunningham@dcs.shef.ac.uk

1 Introduction

Querying large RDF spaces with traditional query languages such as SPARQL is challenging as it requires a familiarity with the structure of the RDF graph and the names (URIs) of its classes, properties and relevant individuals. In this paper, we propose a complementary approach based on Vector Space Models (VSM), more concretely Random Indexing (RI) [1] for building a *semantic index* for a large RDF graph. Traditionally, a *semantic index* captures the similarity of *terms* based on their contextual distribution in a large document collection, and the similarity between *documents* based on the similarities of the terms contained in them. By creating a semantic index for an RDF graph, we are able to determine contextual similarities between graph nodes (e.g., URIs and literals) and based on these, between arbitrary subgraphs. These similarities can be used for finding a ranked list of *similar* URIs/literals for given input term which can be used for exploring the repository or enriching SPARQL queries.

2 SPARQL Query Refinement

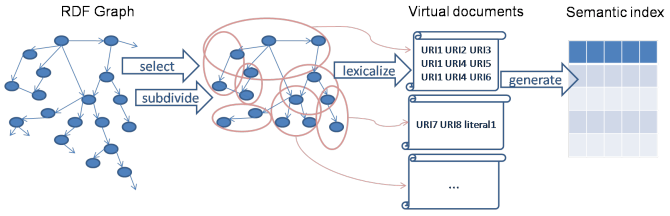

Fig. 1. Generating vectors from an RDF graph

In order to apply RI to RDF graphs, we first *select* the relevant part of the original graph and *subdivide* it into a set of potentially overlapping subgraphs. The next step is *lexicalization* in order to create *virtual documents* from these

subgraphs. Finally, we generate the semantic index from the virtual documents (see Figure 1). The details of how each of these steps is performed significantly influences the final vector space model. For example, in the *selection and subdivision* step, all or just a part of the TBOX could be selected; the subgraphs could be individual *triples*, or *RDF molecules* (the set of triples sharing a specific subject node), or more complex/bigger subgraphs. In the *lexicalization* step, the URIs, blank nodes, and literals from an RDF subgraph are converted to a sequence of terms. When generating the semantic index, different strategies for creating tokens and performing normalization have to be applied to typed literals, string literals with language tags, and URIs. Once the semantic index has been created, it can be used to find similarities between URIs, literals, and RDF subgraphs. We use the ranked list of similar terms for URIs/literals that occur in certain kinds of SPARQL queries to make the query more generic and also return results for entities that are semantically related to those used in the original query (see Figure 2).

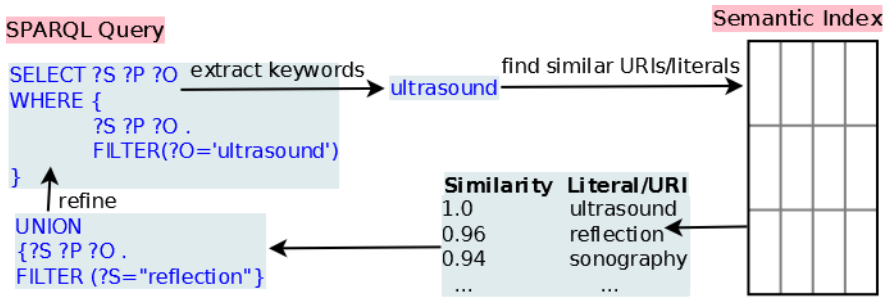

Fig. 2. Refinement of the SPARQL query using Random Indexing

We have applied our approach to parts of the Linked Life Data (LLD[1]) RDF graph and are in the process of adding a search facility based on RI to the LLD website. The code is available as a plugin of the LarKC platform from `http://sourceforge.net/projects/larkc/`.

Acknowledgments This research has been supported by the EU-funded LarKC[2] (FP7-215535) project.

References

1. Karlgren, J., Sahlgren, M.: From words to understanding. In: Uesaka, Y., Kanerva, P., Asoh, H. (eds.) Foundations of Real-World Intelligence, pp. 294–308. Stanford: CSLI Publications (2001)

[1] `http://linkedlifedata.com/`
[2] `http://www.larkc.eu/`