L3S Research Center at the Semsearch 2010 Evaluation for Entity Search Track

Gianluca Demartini Philipp Kärger George Papadakis
L3S Research Center, Hannover, Germany
{demartini,kaerger,papadakis}@L3S.de

Peter Fankhauser
Fraunhofer IPSI,
Darmstadt, Germany
fankhauser@ipsi.fraunhofer.de

ABSTRACT

In this paper we describe our submission to the Entity Search Track of the Semantic Search Workshop 2010. The goal is to find those Semantic Web URIs in an RDF graph that best match a given keyword query. The collection provided is an RDF graph of more than one billion triples—the collection of the Billion Triple Challenge (BTC) 2009 held at ISWC2009. Our submission tries to prominently improve effectiveness of such a search task and shows that the loss in efficiency is low enough to compete with other approaches. In order to achieve a short query time, we disregard everything in the collection except the URIs and follow the assumption that the most relevant URIs to a keyword query contain most of the query's keywords.

1. GENERAL IDEA

Our submitted run (termed "L3S" in the charts) exploits the assumption that URIs do carry semantic information. In short: our approach consists in searching a keyword query against an inverted index of URIs.

This approach follows the observation that a high percentage of the URIs currently used in the Semantic Web contain meaningful strings serving as human understandable labels for the underlying resource. For example. the URI

`http://dbpedia.org/resource/Air_Wisconsin`

falls into this category, whereas

`http://sw.cyc.com/concept/Mx4rv1hbBpwpEbGdrcN5Y29ycA`

does not. Of course, it is not always clear, whether the string in a URI is meaningful w.r.t. to the resource or even a query.

In addition to meaningful identifiers encoded in URIs, of course also the actual statements about the resources may provide valuable information for query matching. Overall, in the collection there are 182,424,110 unique URIs (appearing both as subjects and objects). Out of these URIs, 12.55% appear only as subjects and 30.49% appear only as objects. The rest (56.96%) appear both as subjects and objects. This indicates a strong connection between nodes in the graph and the presence of many relations between entity identifiers. Out of a total of 1,151,383,509 RDF statements there are 126,808,344 unique subjects. That is, each subject has an average of 9 statements describing it. One may consider such set of statements as an initial entity description (or entity profile) that can be further processed and matched against keyword queries.

However, we decided not to follow this approach and to focus on the URIs only. This decision on the one hand has the drawback that lots of potentially useful information (e.g., literals directly connected to a URI by means of a name property) is disregarded. On the other hand, the size of the collection is reduced tremendously which provides a gain in efficiency.

2. APPROACH DESCRIPTION

In the following, the preparation of the collection and the execution of the keyword queries is explained in detail.

First, we created an index over the 182 million unique URIs that appear either as subjects or as objects of a predicate in the BTC 2009 collection. In order to create such an index we considered each URI a document composed of URI tokens. Such tokens are obtained by splitting on special characters using a straightforward regular expression. For example, the URI `http://dbpedia.org/resource/Air_Wisconsin` is split into the following set of tokens:

{`http`, `dbpedia`, `org`, `resource`, `Air_Wisconsin`}.

The weighting function used for tokens in the index is TF-IDF. This way common tokens like `http` and `dbpedia` have lower importance for a given query because of the high document frequency.

Second, a keyword query is run against the inverted index and URIs are ranked according to their cosine similarity with the query: the more keywords are present in the URI the higher it is ranked.

Technical details. The inverted index has been build using the Lucene library v3.0.0[1]. The constructed index including all split and original URIs consists of 20GB. The system running on a single machine took an average time of 40ms to run the 92 queries included in the testset.

[1] `http://lucene.apache.org/`