R3 - A Related Resource Recommender

Thomas Kurz, Tobias Bürger and Rolf Sint

Salzburg Research Forschungsgesellschaft
Jakob Haringer Str. 5/3, 5020 Salzburg, Austria
firstname.lastname@salzburgresearch.at

Abstract. Due to the ever growing amount of content in the Web of Data, the retrieval of relevant information is challenging. Currently, efficient resource recommendation methods are lacking, that could ease the exploration of data in the Web of Data. To alleviate this situation, this paper proposes the R3 resource recommendation framework for retrieval of data in the Linked Open Data (LOD) cloud. It analyses relevant search engines and interlinking frameworks and, based on that, proposes the R3 framework which is illustrated both in theoretical and practical details. The framework enables the recommendation of (RDF) resources from the LOD cloud based on textual, structural, or semantic similarity.

1 Introduction

The goal of Linking Open Data (LOD) community is to bootstrap the Semantic Web (the "Web of Data") by publishing and interconnecting datasets using RDF[1]. The outcome of this movement is the so called LOD cloud which grew to 13.1 billion triples and 142 million RDF links in the last two years and it is still growing [2].
As within the traditional, document-centric Web, search and retrieval of information is of utmost importance. Similarly, a big challenge for a specific end user or application, operating on the Web of Data, is to find relevant data that serves their specific needs. Despite the fact, that Linked Data browsers and search engines are available to explore content in the LOD cloud, means to issue complex queries by ordinary users or to recommend content in the cloud based on particular interests, are currently lacking. In case a user is searching for the city of Berlin using a LOD search engine, he is able to retrieve resources with many properties such as their names, descriptions, latitude, longitude, or density of population. If she now would like to retrieve related resources such as a ranked list of cities ordered by geographical distance and/or density of population or resources with similar structure (like countries or provinces) ranked on the semantic similarity of their textual description, she will fail with current search engines. Similarly the recommendation of related resources could allow the user to issue a "Query by Example" by defining some kind of a fake-resource and use it as query base, which would be a novel form for searching the Web of Data.

In order to alleviate this situation, this paper investigates the state of the art in LOD search engines and interlinking frameworks (Section 2) and, based on that, proposes the R3 resource recommendation framework that is capable of recommending data from the LOD cloud based on the semantic, structural, or textual similarity of given resources. The framework allows to query for related things in the LOD cloud based on a given resource and is illustrated including its requirements, conceptual architecture, and implementation aspects (Section 3). Finally, details are given on how to further advance and implement the framework (Section 4).

2 Resource Discovery and Interlinking in the LOD Cloud

There are some applications on the web, which allow the user to search or browse the web of data. Supplementary to that there are so called Interlinking Frameworks that can be used to check the resources of two or more different datasets pairwise for similarity. Because of the analogies to our approach these frameworks should also be considered in the following discussion.

2.1 Browsers and Search Engines

Sindice[1], as described in [3], is a scalable index of the Semantic Web. It crawls the Web for RDF Documents and Microformats and indexes resulting resource URIs, Inverse Functional Properties (IFPs) and keywords. A human user can access these documents through a simple user interface, based on indexes mentioned above.
Sigma[2] is rather a semantic information mashup enabled by Sindice than a self-contained semantic search service. Nevertheless it enriches a lot of its functionalities with some nice additional features. It works as Web of Data browser where the user can start from any entity (found by a fulltext search) and then browse to the resulting page. The resources index is build out of from sites which use RDF, RDFa or Microformats.
The Open Link Search[3] will list entities with a user-defined text pattern occurring in any literal property value or label. It also supports Entity URI lookup. The Search can be redefined by filtering type, property value, etc.
It is also possible to execute SPARQL queries by using the SPARQL endpoint. Some demo queries are predefined and can easily be altered via text input fields.
Falcons[4] is described in [5] as a service for searching and browsing entities on the Semantic Web. It is a keyword-based search engine for the Semantic Web URIs and provides different query types for object, concept and document search.
Falcons also gives the facility of facetting over types by dynamically recommending ontologies. The recommendation is based on a combination of the TF-IDF technique and the popularity of ontologies.

[1] `http://sindice.com/`
[2] `http://sig.ma/`
[3] `http://lod.openlinksw.com/`
[4] `http://iws.seu.edu.cn/services/falcons/objectsearch/index.jsp`

Watson[5] offers keyword based querying to obtain a URI-list of semantic documents in which the keywords appear as identifiers or in literals of classes, properties, and individual. Search options make it possible to restrict the search space to particular types of entities (classes, properties or individuals) and to particular elements within the entities (e.g. local name, label, comment).
SWSE[6] is a search engine for the RDF Web. Similar search engines currently provided for the HTML Web it looks like a ordinary fulltext search. But the information retrieval capabilities of SWSE are much more powerful because of the inherent semantics of RDF and other Semantic Web languages.
Swoogle[7] allows a user to search through ontologies, instance data, and terms of the Semantic Web. Furthermore it supports browsing the Web of Data. This search engine also uses an archive functionality to identify and provide different versions of Semantic Web documents.

Like described above, each considered semantic search service provides a certain amount of functionalities. Some of them are part of two or more services, others are exclusive to one certain engine. Though it is possible to search for appearance of a given resource in some of them, neither it is possible to find related resources for a resource and its RDF triples nor to define on which triples the relationship should be calculated on. Also the search engines do not consider a semantic similarity of queries and content, which definitely could increase the quality of result. But there are applications in the area of Semantic Web which match some of these requirements in certain ways - the interlinking frameworks.

2.2 Interlinking frameworks

Interlinking frameworks for semantic web data try to detect related and link resources in different datasets. In [8] several frameworks are compared to each other concerning their functionalities, which brings us to the decision that the Silk[8] approach is rather related to our goals.
Silk[7] is a framework for detecting explicit RDF links between data items within different data sources. Using the declarative Silk - Link Specification Language (Silk-LSL), developers can specify which types of RDF links should be discovered between data sources and, based on arbitrary metrics and aggregation functions, which resources should be declared as related. Silk accesses the interlinking candidates via the SPARQL protocol.
The usage of different metrics and aggregation functions for different types of properties can be adopted to our resource recommender. In addition we can remodel Silk-LSL in some ways (e.g. alternative metrics) and use it as query syntax. This language makes it also possible to define the appropriated datasources by query.

[5] http://kmi-web05.open.ac.uk/WatsonWUI/

[6] http://swse.deri.org/

[7] http://swoogle.umbc.edu/

[8] http://www4.wiwiss.fu-berlin.de/bizer/silk/spec/

3 R3 - A Conceptual Overview

Our intent is to build a recommender service, which allows to query for related resources from various (predefined) datasources based on a given resource. But what is relatedness, what factors have an impact on it and how can we implement such a recommender service? This is discussed in the following sections.

3.1 Requirements

In case of RDF resources there are various factors which define relatedness. On the one hand the RDF structure itself (predicates and non-literal objects) reveals something about how similar two resources are. On the other hand the literal properties can be compared according to their types towards different metrics. That can be simple ones like euclidean metric for numbers, or more complex like semantic similarity of texts. A user should be able to specify the factors that are used to find relevant related resources, and also its impact on the result. In addition to that the whole recommendation process should be calculated in an adequate time. So we can specify requirements below:

1. Recommend related resources from the LOD cloud based on a given RDF resource.
2. Consider semantic similarity of texts and structural similarity of resources.
3. Offer a comparison mechanism for literals with adjustable metrics.
4. Allow user defined feature boost; that means a certain feature (e.g. property x or structure) has a higher relevance on relatedness than others.
5. Return related resources ordered by relevance.

3.2 Conceptual Architecture

The concept to fulfill these requirements is illustrated in Figure 1. The data must be fetched from the LOD cloud, combined and indexed; it should be queryable via a specific search syntax. This process is described more precisely in this section.

Data Consolidation
The service gets recommendable resources out of the Linked Data Cloud. Since it should possible, to build a multi-source index, there must be a kind of ontology alignment. Thus preprocessed data is stored directly into the index. The single datasources must be reindexed in given time intervals.

Resource Recommender Index
A core index can provide lot of metrics like euclidean distance, date similarity, string equality, etc. Semantic similarity which can be used to evaluate the semantic distance of texts and RDF structures is more complex, therefore we need a supplemental semantic index. Semantic textual indices (one for each defined property) as well as the semantic structure index (one for the whole dataset) are

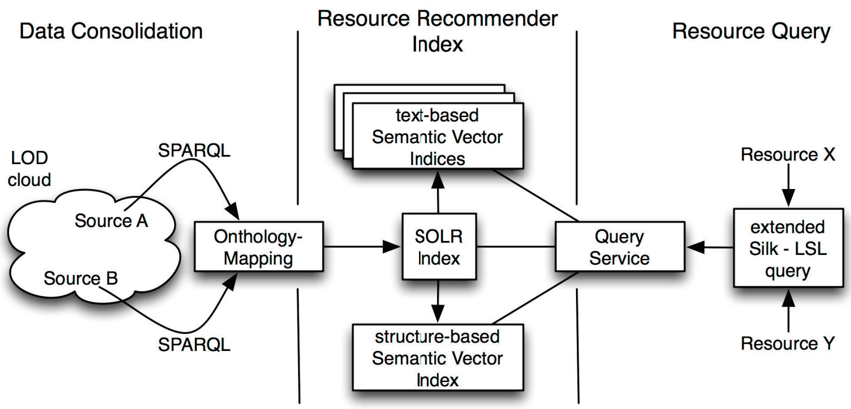


Fig. 1: Design and workflow of R3

build out of the core index.

Resource Query
To get recommended resources based on a given one, the recommender provides a query language, whereby the user can specify, which features should be included in the calculation according to which metric. Furthermore the factor how intensive a specific feature impacts the result and how the diverse values are combined is configurable by query. To restrict the set of base resources the user can define the included datasets. The searchresult is list of resources ranked by relevance.

3.3 Implementation

Datasets, which build our resources base is taken out from the LOD cloud via SPARQL. To map different resources from sources we use a simple mapping table. Complex ontology matching strategies like in [9] are also possible.
Because of its high scalability, its fast query processing and the possibility to use integrated functions and numerical as well as token-based comparison, we decided to use SOLR[9] as our index base. A lot of metrics like euclidean distance, date similarity, string equality, etc. are provided by or can be directly integrated into SOLR index. As described, for more complex metrics we need supplemental semantic indices build out of the SOLR index.

Text-based Semantic Index
A potential semantic index can be a Semantic Vector Index. This approach bases upon the Vector Space Model wherein every document is represented as a vector in an n-dimensional term space according to appearing terms. The Semantic

[9] `http://lucene.apache.org/solr/`

Vector Package[10] is able to build such an Index (which can be queried for semantic related documents) out of the basic Lucene Index.

Structure-based Semantic Index
The semantic vector index can also be used to index the semantic similarity of RDF structures. Therefore not every word or text module is integrated in the term model but the URI, RDF predicates and non-literal objects of a resource. Figure 2 shows the semantic similarity of a subset of dbpedia resources. To illustrate this semantic space we build a structure distance matrix of this resources and scaled it to two dimensions using classical multidimensional scaling (MDS) offered by the R statistics software[11]. We highlighted resources of different types which shows that related resources have a similar RDF structure.

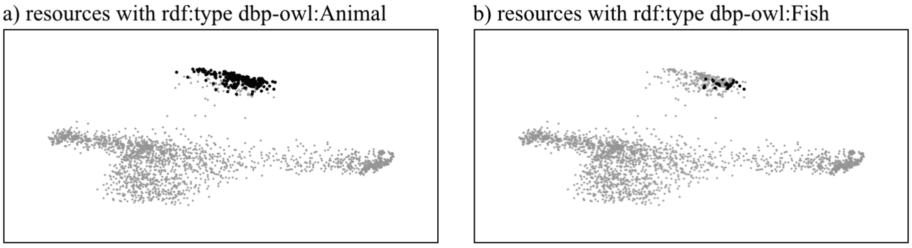


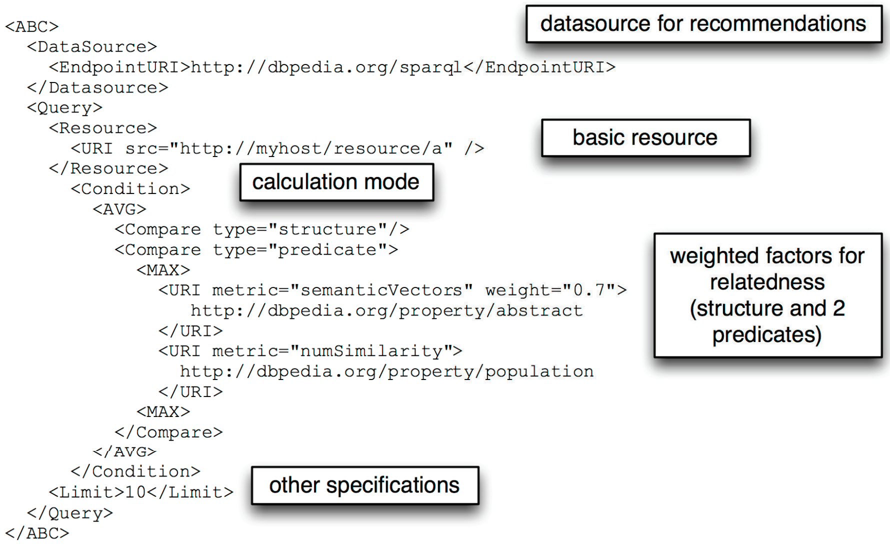
Fig. 2: Evaluation of Structure Index

Query Language
As mentioned, the SILK Link Specification Language[12] can be used as inspiration for a query format that fulfills our query requirements and allows to specify the basic resource (set of RDF triples or URI), the considered datasets (SPARQL endpoints used from data consolidator), relevant features and its impact and the applied metrics (taken from a fix set). Figure 3 shows an simple query example.

4 Further Work

In this paper we described the conceptual architecture of a resource recommendation framework for the Semantic Web. Our future work includes the implementation of this concept and a practical evaluation with real datasets. In a further step we plan to optimize the Semantic Vector package, which is used in one core

[10] http://code.google.com/p/semanticvectors/
[11] http://www.r-project.org/
[12] http://www4.wiwiss.fu-berlin.de/bizer/silk/spec/#specification



```
<ABC>
  <DataSource>
    <EndpointURI>http://dbpedia.org/sparql</EndpointURI>
  </Datasource>
  <Query>
    <Resource>
      <URI src="http://myhost/resource/a" />
    </Resource>
      <Condition>
        <AVG>
          <Compare type="structure"/>
          <Compare type="predicate">
            <MAX>
              <URI metric="semanticVectors" weight="0.7">
                 http://dbpedia.org/property/abstract
              </URI>
              <URI metric="numSimilarity">
                http://dbpedia.org/property/population
              </URI>
            <MAX>
          </Compare>
        </AVG>
      </Condition>
    <Limit>10</Limit>
  </Query>
</ABC>
```

datasource for recommendations
basic resource
calculation mode
weighted factors for relatedness (structure and 2 predicates)
other specifications

Fig. 3: Sample for a Recommander Query

component of the framework, to enhance its scalability and performance. The resulting recommender will be integrated into the KiWi[13] system.

References

1. C. Bizer et al. Linked Data - The Story So Far. International Journal on Semantic Web and Information Systems (IJSWIS), Vol. 5, Issue 3, 2009.
2. Linking Open Data: W3C SWEO Community Project. http://esw.w3.org/topic/SweoIG/TaskForces/CommunityProjects/LinkingOpenData, 2010.
3. E. Oren et al. Sindice.com: a document-oriented lookup index for open linked data. Int. J. Metadata, Semantics and Ontologies, Vol. 3, No. 1, 2008.
4. DERI Galway: Sindice API for Query Services. http://sindice.com/developers/api, 2008-2009.
5. G. Cheng and Y. Qu. Searching linked objects with Falcons: Approach, implementation and evaluation. International Journal on Semantic Web and Information Systems 5(3):49-70, September 2009
6. W.B. Frakes and R.A. Baeza-Yates. Information Retrieval: Data Structures and Algorithms. Prentice-Hall, New Jersey, 1992.
7. J. Volz et al. SILK - A Link discovery framework for the Web of Data. Linked Data on the Web (LDOW2009), Madrid, 2009.
8. F. Scharffe and J. Euzenat. Alignments for data interlinking. http://melinda.inrialpes.fr, 2009
9. C. A. Curino et al. X-SOM: A Flexible Ontology Mapper. 18th International Conference on Database and Expert Systems Applications (DEXA 2007), 2007.

[13] http://kiwi-project.eu/