RDF Digest: Ontology Exploration Using Summaries

Georgia Troullinou, Haridimos Kondylakis, Evangelia Daskalaki, Dimitris Plexousakis

Institute of Computer Science, FORTH, N. Plastira 100, Heraklion, Greece

{troulin, kondylak, eva, dp}@ics.forth.gr

Abstract. Ontology summarization aspires to produce an abridged version of the original ontology that highlights its most representative concepts. In this paper, we present *RDF Digest*, a novel platform that automatically produces and visualizes summaries of RDF/S Knowledge Bases (KBs). A summary is a valid RDFS document/graph that includes the most representative concepts of the schema, adapted to the corresponding instances. To construct this graph our algorithm exploits the semantics and the structure of the schema and the distribution of the corresponding data/instances. A novel feature of our platform is that it allows summary exploration through extensible summaries. The aim of this demonstration is to dive in the exploration of the sources using summaries and to enhance the understanding of the various algorithms used.

1 Introduction

Given the explosive growth in both data size and schema complexity, data sources are becoming increasingly difficult to understand and use. Ontologies often have extremely complex schemas which are difficult to comprehend, limiting the exploration and the exploitation potential of the information they contain. Besides schema, the large amount of data in those sources increase the effort required for exploring them.

Over the latest years, various techniques have been provided on constructing overviews on ontologies [1-4], maintaining however the more important ontology elements. These overviews are provided by means of an ontology summary. Ontology summarization [4] is defined as the process of distilling knowledge from an ontology in order to produce an abridged version. While summaries are useful, creating a "good" summary is a non-trivial task. A summary should be concise, yet it needs to convey enough information in order to enable a decent understanding of the original schema. Moreover, the summarization should be coherent and should provide an extensive coverage of the entire ontology. So far, although a reasonable number of research works tried to address the problem of summarization from different angles, a solution that simultaneously exploits the semantics of the schemas and the data instances is still missing.

In this demonstration, we focus on RDF/S KBs and demonstrate for the first time the implementation of the algorithms introduced in [5]. Our system constructs summaries that constitute "valid" sub-ontologies and provide an overview of the ontology schema

considering a) the *semantics* of the schema, b) the *structure* of the graph and c) the *distribution* of the corresponding *data*/*instances*. Extending our previous work [5] we demonstrate also an efficient and effective method to explore these KBs using schema summaries that can be extended according to user selections. In addition, we provide more meta-data to enhance ontology understanding. To the best of our knowledge, our approach is the first, in the context of ontology, combining both schema and data to allow ontology exploration though a high-quality graph summary.

2 Approach

In this section we present the properties that a sub-graph of our schema is required to have in order to be considered a high-quality summary of an RDF/S KB. Specifically, we are interested in important schema nodes that can describe efficiently the whole schema and reflect the distribution of the data instances at the same time. To capture these properties, we use the notions of *relevance* and *coverage*. *Relevance* is used for identifying the most important nodes and *coverage* is used for extracting paths, which cover the whole spectrum of the RDF/S document.

In our approach, initially, we determine the importance of a node/edge, judging from the instances it contains by calculating its *relative cardinality*. The *Relative Cardinality* ($RC(e(v_i, v_j))$ of an edge $e(v_i, v_j)$ is the number of the specific instance connections divided by the total number of the connections of the instances of these two nodes v_i, v_j. After that, in order to combine the notion of centrality in the schema and the distribution of the corresponding dataset, we define a variation of the *degree centrality*, called *in*/*out centrality* (C_{in}/C_{out}) as the sum of the weighted relative cardinalities of the incoming/outgoing edges. The weights are experimentally defined and depend on the types of the properties, giving priority to user-defined properties. The algorithm is flexible enough to focus on the available instances when they exist, and if they are not available, it only exploits the semantics and the structure of the schema.

The notion of centrality, as defined previously, is a measure that can give an intuition about how central a schema node is in an RDF/S KB. However, its importance should be determined considering also the centrality of the other nodes as well. To achieve this goal, the *relevance* of a node is affected by its surrounding neighbors and more specifically by the number and the connections of its adjacent nodes.

Definition 2.1 (Relevance of a node). Let np_{in} be the number of incoming nodes v_i connected to v with $e_a(v_i, v)$ and np_{out} be the number of outgoing nodes v_j connected to v with $e_b(v, v_j)$. The *relevance* of v, i.e. the *Relevance(v)*, is the sum of *in* and *out centrality* of v multiplied by the corresponding number of nodes, divided by the sum of out-centrality of the incoming nodes v_i and the in-centrality of the outgoing nodes v_j.

$$Relevance(v) = \frac{C_{in}(v) * np_{in} + C_{out}(v) * np_{out}}{\sum_{1}^{np_{in}} (C_{out}(v_i)) + \sum_{1}^{np_{out}} (C_{in}(v_j))}$$

Obviously, the relevance of a schema node in an RDF/S KB is determined by both its connectivity in the schema and the cardinality of the instances. In addition, the produced summary should be a valid schema graph. So the chosen paths should be selected

having in mind to collect the more relevant nodes by minimizing the overlaps. As a consequence, the main criteria to estimate the level of coverage of a specific path are: a) the relevance of each node in the path, b) its relevant instances in the dataset and c) the length of the path. As a result, similar to [3], we define the notion of coverage.

Definition 2.2 (Coverage of a path). The *coverage* of a path from v_s to v_i, i.e. the $Coverage(v_s \rightarrow v_i)$, is derived by the sum of the relevance of the sequential nodes v_j contained between the nodes v_s and v_i, multiplied by the relative cardinality of each edge $e(v_{j-1}, v_j)$ contained in the path. The result is divided by the length of the path in order to penalize the longer paths.

$$Coverage(v_s \rightarrow v_i) = \frac{1}{d_{n_s \rightarrow n_i}} * \sum_{j=2}^{d_{n_s \rightarrow n_i}} \left(Relevance(v_j) * RC\left(e\left(v_{j-1}, v_j\right)\right)\right)$$

The above formula aims to select the schema nodes that are more relevant while avoiding having nodes (or paths) in the summary which cover one another. The highest the coverage of a path, the more appropriate is considered in representing the original graph or part of it. For more information on the aforementioned formulas the interested reader is forwarded to the relevant publication [5].

According to the aforementioned formula, each selected node represents/covers a part of its neighborhood in the summary graph. In order to enable further exploration, we allow the extension of the summary on a node of interest. Our algorithm is trying to identify the neighbors that are not included in the current summary and until now they are represented/covered by the selected node. Having calculating the coverage of all paths starting from the selected node to all its neighbors, our algorithm includes in the summary those nodes contained in the paths that minimize the coverage compared to the paths (set of nodes) that have been already inserted in the existing summary.

3 Architecture & Demonstration Highlights

Based on the aforementioned metrics, the *RDF Digest* prototype has been implemented. The architecture of the system is shown in Fig. 1 and a beta version of the platform is currently available online (http://www.ics.forth.gr/isl/rdf-digest). The RDF Digest is composed of two major components, the *Summarizer* and the *Visualizer*.

Using the interface, a user can select or give the URL of an online RDF/S document, she would like to be summarized and is optionally able to define the expected length of the summary. The *Summarizer* gets the input RDF/S document and preprocesses it (using the *RDF Preprocessor* module) by computing the corresponding RDF/S KB. The result is stored in a Virtuoso instance to enable efficient data access. Then, the *RDF Accessor* module calculates the relevance of each node. The *RDF Summary Builder* generates the final summary of the schema, based on the rankings produced by the *RDF Assessor* and the requested size of the summary. The result and additional meta-data are returned to the *Visualizer* which enables effective visualization of the summary and exploration of the data source as shown in the right of Fig. 1.

In our demonstration, example ontologies will be used for generating summaries and their exploration through extensible summaries will be demonstrated. In the presented summary graph, the size of a node is depending on the node's *relevance*. In addition by

clicking on a node, additional meta-data (its relevance and centrality, the number of instances, the connected properties, instances etc.) are provided to enhance ontology understanding. Besides meta-data, further exploration of the data source is allowed by clicking further on the details (on the left) of the selected class and the properties. When clicked, its instances and connections appear in a pop-up window. Moreover, further exploration of the data source is allowed by double-clicking on a node to extend the summary on that specific node. Finally the user is able to download the summary as a valid RDFS document.

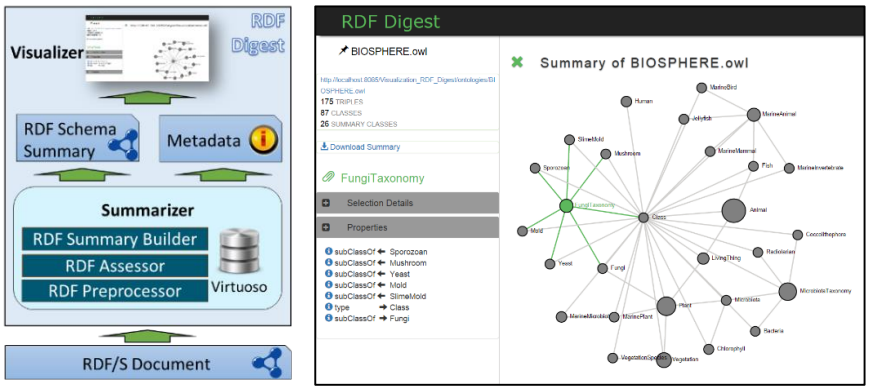

Fig. 1. The architecture of the RDF Digest (left) and a screenshot of the Visualizer (right).

Our immediate plans comprise the extension of RDF Digest to handle multi-ontology KBs, possibly by using external SPARQL endpoints, and to evaluate the summaries produced by checking if they can answer the most frequent queries issued to these KBs. As the size and the complexity of schemas and data increase, ontology summarization is becoming more and more important and several challenges remain to be investigated in the near future.

Acknowledgments: This work was partially supported by the EU projects iManageCancer (H2020-643529), MyHealthAvatar (FP7-600929) and EURECA (FP7-288048).

4 References

1. Peroni, S., Motta, E., and d' Aquin, M.: Identifying Key Concepts in an Ontology, Through the Integration of Cognitive Principles with Statistical and Topological Measures. In The Semantic Web Journal. 5367, pp. 242–256 (2008)
2. Queiroz-Sousa, P. O., Salgado, A. C., Pires, C. E.: A Method for Building Personalized Ontology Summaries. JIDM Journal, 4, 3, pp. 236 (2013)
3. Yu, C., Jagadish, H.V.: Schema Summarization, VLDB, pp. 319-330 (2006)
4. Zhang, X., Cheng, G., Qu, Y: Ontology Summarization Based on RDF Sentence Graph. WWW, pp. 707-716 (2007)
5. Troullinou, G., Kondylakis, H., Daskalaki, E., Plexousakis, D.: RDF Digest: Efficient Summarization of RDF/S KBs, ESWC, pp 119-134 (2015)