Ontology Understanding without Tears: The Summarization Approach

Editor(s): Fabien Gandon, INRIA, France; Marta Sabou, Technische Universität Wien, Austria; Harald Sack, Hasso Plattner Institute, Germany
Solicited review(s): Szymon Klarman, Grakn Labs, UK; Silvio Peroni, Università di Bologna, Italy; Vinu E. V, Indian Institute IT Madras, India; One anonymous reviewer

Georgia Troullinou, Haridimos Kondylakis*, Evangelia Daskalaki and Dimitris Plexousakis
Institute of Computer Science, FORTH, N. Plastira 100, Heraklion, Greece

Abstract. Given the explosive growth in both data size and schema complexity, data sources are becoming increasingly difficult to use and comprehend. Summarization aspires to produce an abridged version of the original data source highlighting its most representative concepts. In this paper, we present an advanced version of the *RDF Digest*, a novel platform that automatically produces and visualizes high quality summaries of RDF/S Knowledge Bases (KBs). A summary is a valid RDFS graph that includes the most representative concepts of the schema, adapted to the corresponding instances. To construct this graph we designed and implemented two algorithms that exploit both the structure of the corresponding graph and the semantics of the KB. Initially we identify the most important nodes using the notion of relevance. Then we explore how to select the edges connecting these nodes by maximizing either locally or globally the importance of the selected edges. The extensive evaluation performed compares our system with two other systems and shows the benefits of our approach and the considerable advantages gained.

Keywords: Semantic Summaries, RDF/S documents/graphs, Schema Summary

1. Introduction

The vision of Semantic Web is the creation of a common framework that allows data to be shared and reused across application, enterprise, and community boundaries. Ontologies are playing an important role in the development and deployment of the Semantic Web since they model the structure of knowledge and try to organize information for enhancing the understanding of the contextual meaning of data. Ontologies have been used in database integration [1], obtaining promising results, for example in the fields of biomedicine and bioinformatics [2], but also as means for publishing large volumes of interlinked data from which we can retrieve abundant knowledge. The Linked Open Data cloud for example contains more than 62 billion triples (as of January 2014) [3].

Given these sizes, in nowadays, data sources are becoming increasingly difficult to understand and use. They often have extremely complex schemas which are difficult to comprehend, limiting the exploration and the exploitation potential of the information they contain. Moreover, regarding ontology engineering, ontology understanding is a key element for further development and reuse. For example, a user/ontology engineer, in order to formulate queries [4], has to examine carefully the entire schema in order to identify the interesting elements. Besides schema, the data contained in sources should also help to identify the most important or relevant items. Currently, an efficient and effective way to understand the content of each source without examining all data is still a blind spot.

As a result, there is now, more than ever, an increasing need to develop methods and tools in order to facilitate the understanding and exploration of various data sources. Approaches for ontology modularization [6] and partitioning [7] try to minimize and

* Corresponding author. E-mail: kondylak@ics.forth.gr

partition ontologies for better understanding but without preserving the important information. Other works focus on providing overviews on the aforementioned ontologies [7, 8, 9, 10, 11] maintaining however the more important ontology elements. Such an overview can also be provided by means of an ontology summary. Ontology summarization [10] 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 to enable a decent understanding of the original schema. Moreover, the summarization should be coherent and 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 both the structure and the semantics provided by the schemas and the data instances is still missing.

In this paper, we focus on RDF/S ontologies and demonstrate an efficient and effective method to automatically create high-quality summaries. We view an RDF/S KB as two distinct and interconnected graphs, i.e. the schema and the instance graph. As such, a summary constitutes a "valid" sub-schema graph providing an overview of the original schema considering also the available data. Specifically our contributions are the following:

- A novel platform that automatically produces RDF schema summaries highlighting the most representative concepts of the schema adapted to the corresponding data instances.
- In order to construct these graph summaries our system exploits the *structure* and the *semantics* of the KB. It differentiates schema and instance nodes and assigns different weights according to the types of properties (user-defined and standard RDF/S properties) in order to identify and select the most important and relevant elements of the ontology.
- To identify the most important nodes we define the notion of *relevance* based on the *relative cardinality* and the *in/out degree centrality* of a node.
- Since the summary we would like to construct is a sub-graph out of the original schema graph containing the most important (relevant) nodes we try next to *identify the proper paths connecting those nodes*. We achieve this by implementing two diverse algorithms trying to maximize in essence *locally* or *globally* the *importance* of the selected edges.
- We present the corresponding algorithms and elaborate on their implementation details and their complexity.
- Our detailed experimental evaluation shows the benefits of our approach. Initially, we compare our algorithms with other works that select only the most important nodes as a summary showing the added value of our system. Next, we identify that sub-graph selection through global importance maximization has better results in almost all cases.

To our knowledge, this is a unique approach that, in the context of ontology, combines both schema and data instance information to enable KBs exploration through high-quality summary schema graphs.

An initial version of our work has already been presented [13] and demonstrated [14]. This paper extends our previous work in several ways. Our previous work could not handle blank nodes. However, as identified during our evaluation, blank nodes are apparent in many ontologies and KBs and we cannot keep ignoring them. Besides the variation handling blank nodes of the first algorithm, in this paper we present a new algorithm for selecting the edges to be included in the constructed summary, moving out from local maximization to global maximization of the importance of the selected edges. The implementation details and the complexities are presented, whereas the updated system provides more meta-data to enhance ontology understanding. Our expectations for improvement on the results are confirmed and presented in a new section. Besides benchmarking, the two algorithms and comparing them with two existing solutions that only select the most important classes as a summary, we conduct a completely new user evaluation study with ontologies with instances, and we evaluate the quality of the entire summary graph. In addition, we measure the execution times of our algorithms and we compare them with one of the existing solutions we could get access to.

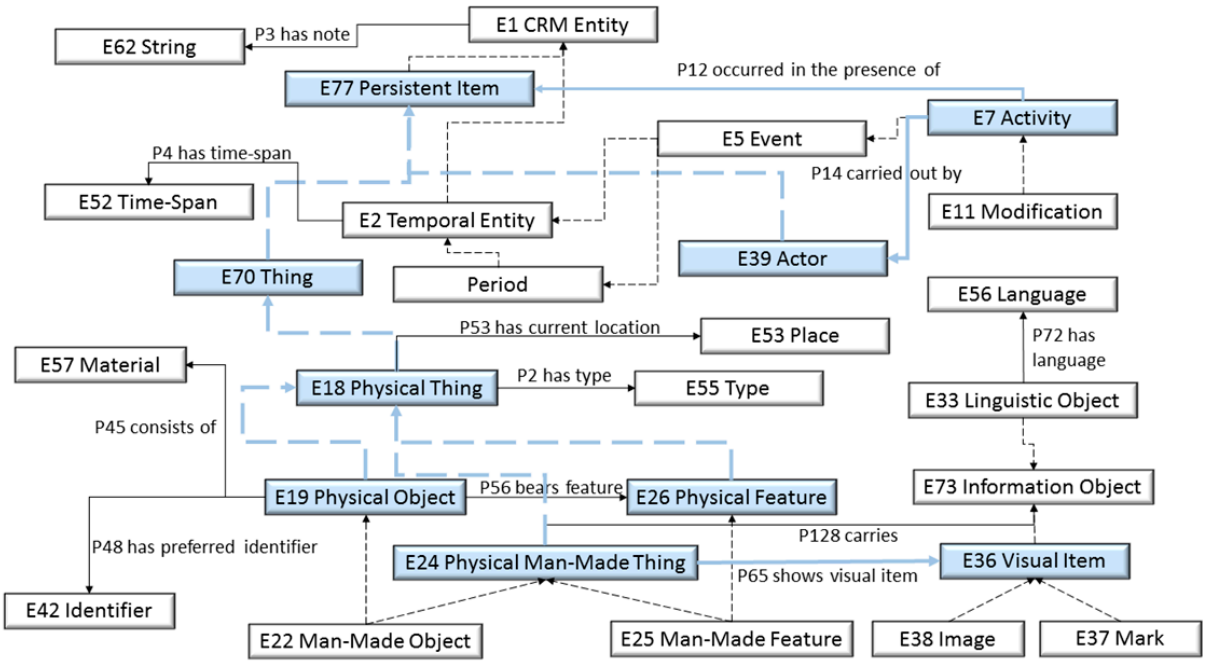

Fig. 1. An example schema graph and the corresponding schema summary (in blue)

The rest of the paper is organized as follows. Section 2 introduces the formal framework of our solution and Section 3 describes the metrics used in our algorithms to determine the nodes and paths to be included in the summary. Section 4 presents the two algorithms for selecting edges in order to construct the summary graph and Section 5 our implemented system. Section 6 describes the evaluation conducted whereas Section 7 presents related work. Finally, Section 8 concludes the paper and presents directions for future work.

2. Preliminaries

Schema summarization aims to highlight the most representative concepts of a schema, preserving "important" information and reducing the size and the complexity of the schema [11]. Despite the significance of the problem, there is still no universally accepted measurement on the importance of nodes in an RDF/S graph. In our approach, we try to elicit this information from the *structure* of the graph and the *semantics* of the KB. Our goal is to produce a simple and expressive graph that presents an overview of the schema and also provides an intuition about the corresponding stored data. Specifically, in this paper we focus on RDF/S KBs, as RDF/S is among the widely used standards for publishing and representing data on the web [3]. We have to note that our approach handles OWL ontologies as well, considering however only the RDF/S fragment of these ontologies.

The representation of knowledge in RDF is based on triples of the form of *(subject predicate object)*. RDF datasets have attached semantics through RDFS [15], a vocabulary description language. Representation of RDF data is based on three disjoint and infinite sets of resources, namely: URIs (U), literals (L) and blank nodes (B). We impose typing on resources, so we consider three disjoint sets of resources: classes ($C \subseteq U \cup B$), properties ($P \subseteq U$), and individuals ($I \subseteq U \cup B$). The set C includes all classes, including RDFS classes and XML datatypes (e.g. *xsd:string*, *xsd:integer*). The set P includes all properties, except *rdf:type* which connects individuals with the classes they are instantiated under. The set I includes all individuals (but not literals). In addition, we should note that our approach adopts the unique name assumption, i.e. that resources that are identified by different URIs are different.

Here, we will follow an approach similar to [16], which imposes a convenient graph-theoretic view of RDF data that is closer to the way the users perceive their datasets. As such, in this work, we separate between the schema and the instances of an RDF/S KB, represented in separate graphs (G_S, G_I respectively). The schema graph contains all classes and the properties they are associated with (via the properties' domain/range specification); note that multiple

domains/ranges per property are allowed, by having the property URI be a label on the edge (via a labelling function λ) rather than the edge itself. The instance graph contains all individuals, and the instantiations of schema properties; the labelling function λ applies here as well for the same reasons. Finally, the two graphs are related via the τ_c function, which determines which class(es) each individual is instantiated under. Formally:

Definition 1 (RDF/S KB). An RDF/S KB is a tuple $V = (G_s, G_I, \lambda, \tau_c)$ such that:

- G_S is a labelled directed graph $G_S = (V_S, E_S)$ such that V_S, E_S are the nodes and edges of G_S, respectively, and $V_S \subseteq C \cup L$.
- G_I is a labelled directed graph $G_I = (V_I, E_I)$ such that V_I, E_I are the nodes and edges of G_I respectively, and $V_I \subseteq I \cup L$.
- A labelling function λ: $E_S \cup E_I \rightarrow \mathbb{P}(P)$ that determines the property URI that each edge corresponds to (properties with multiple domains/ranges may appear in more than one edge). $\mathbb{P}(P)$ is the power set of P.
- A function τ_c: $I \rightarrow 2^C$ associating each individual with the classes that it is instantiated under.

For simplicity, we forego extra requirements related to RDFS inference (subsumption, instantiation), because these are not relevant for our results below and would significantly complicate our definitions. In the following, we will write $p(v_1, v_2)$ to denote an edge e in G_S (where $v_1, v_2 \in V_S$) or G_I (where $v_1, v_2 \in V_I$) from node v_1 to node v_2 such that $\lambda(e) = p$. In addition for brevity we will call *schema node* a node $s \in V_S$, *class node* a node $c \in C \cap V_S$ and *instance node* a node $i \in I \cap V_I$. In addition, a *path* from a schema node v_s *to* v_i, denoted by $path(v_s \rightarrow v_i)$, is the finite sequence of edges, which connect a sequence of nodes, starting from the node v_s and ending in the node v_i. The *length of a path*, denoted by $d_{path(vs \rightarrow vi)}$, is the number of the edges that exist in that path. Finally, having a schema graph G_s, the *closure* of G_s, denoted by $Cl(G_s)$, contains all triples that can be inferred from G_s using inference.

Now, as an example, consider the CRM_{dig}[1] ontology part shown in Fig. 1 used to encode metadata about the steps and methods of production ("provenance") of digitization products and synthetic digital representations. Although this is only a short example, we have 27 classes and many properties that need to be examined in order to understand the schema. In blue color, we can see the summarized graph as it is produced by our method. Obviously, it is easier to understand schema content using only the summary graph. This is due to the fact that only a subset of the nodes is presented to the user, making it easier to comprehend that these nodes are the ones that are the most important out of the initial graph according to the way in which our algorithms assess importance.

[1] http://www.ics.forth.gr/isl/index_main.php?l=e&c=656

3. Assessment Measures

In this section, we present the properties that a sub-graph of a schema is required to have in order to be considered a high quality summary. Specifically, we are interested in important/relevant 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 notion of *relevance* trying to identify the most important nodes.

3.1. Assessing Schema Nodes Importance

Importance has a broad range of meanings and this has led to many different algorithms that try to identify it. Originating from the analysis of social graphs, in the domain of Semantic Web, algorithms adapting the well-known PageRank [8] have been proposed to determine the importance of elements in an XML document. For RDF/S, other approaches use measures such as the *degree centrality*, the *betweenness* and the *eigenvector* centrality *(weighted Page Rank and HITS)* [10], adjusting them to the specific features of RDF/S or they try to adapt *the degree* centrality and *the closeness* [11] to calculate the relevance of a node.

In our case, we believe that the importance of a node should be estimated by the nodes that are directly connected to it and also by the reachability of this node, i.e. the connection of this node with the entire graph, being able to represent effectively its neighbors. Intuitively, nodes with many connections in a schema graph will have a high importance. However, since RDF/S KBs might contain huge amounts of data, that data should also be involved when trying to estimate the importance of the nodes.

Consider for example the node *"E37 Mark"* and the node *"E38 Image"* in the schema graph of Fig. 1. The two nodes have the same number of connections

and they are connected to the same node *"E36 Visual Item"*. Now assume that the node *"E38 Image"* has the double number of instances. Due to the same number of connections, the two nodes may be considered equal but essentially the *"E38 Image"* is more important for the specific RDF/S KB, due to the higher number of instances it contains. Obviously, the number of instances of the class - that a node corresponds to - is a valuable piece of information for identifying its importance.

In our approach, initially, we determine how central/important a node is, judging from the instances it contains (*relative cardinality*). After that, we estimate the centrality of a node in the entire KB (*in/out centrality*), combining the relative cardinality with the number and type of the incoming and outgoing edges in the schema. Finally, the *relevance* of a schema node is defined by comparing its centrality with the centrality of its neighbors.

3.1.1. Relative Cardinality

The cardinality of a schema node is the number of instances it contains in the current RDF/S KB. If there are many instances of a specific class, then that class is more likely to be more important than another with very few instances. Similarly, the cardinality of an edge between two nodes in a schema graph is the number of the corresponding instances of the nodes connected with that specific edge. Using these ideas, we can formally define the relative cardinality of an edge.

Definition 2 (Relative Cardinality of an edge). Let $V = (G_s, G_I, \lambda, \tau_c)$ be an RDF/S KB. The *relative cardinality* of an edge $p(v_i, v_j) \in E_S$ (assuming $E_S \neq \{\}$), denoted by $RC(p(v_i, v_j))$ is the following:

$$RC(p(v_i,v_j)) = \begin{cases} \frac{1}{|\{p(v_a,v_b)\in E_s\}|} + \frac{|\{p(n_i,n_j)\in E_I\}|}{|\{p(n_i,n_a)\in E_I\}|+|\{p(n_b,n_j)\in E_I\}|}, \\ \exists p(n_i,n_j)\in E_I : \lambda(p(n_i,n_j)) = \lambda(p(v_i,v_j)), v_i\in\tau_c(n_i), v_j\in\tau_c(n_j) \\ \frac{1}{|\{p(v_a,v_b)\in E_s\}|}, \\ \nexists p(n_i,n_j)\in E_I : \lambda(p(n_i,n_j)) = \lambda(p(v_i,v_j)), v_i\in\tau_c(n_i), v_j\in\tau_c(n_j) \end{cases}$$

Obviously $|\{p(v_a, v_b)\}|$ is the number of available edges in E_s. In addition, the relative cardinality of a path is the sum of relative cardinalities of the individual edges. Our algorithm is flexible enough to focus on the available instances when they exist, and if they are not available, it only exploits the remaining semantics and the structure of the schema.

3.1.2. In/Out Centrality

In order to combine the notion of centrality in the schema and the distribution of the corresponding dataset, we define the *in/out centrality*, exploiting also the relative cardinality of the various nodes and edges. The *in/out* centrality is an adaptation of *the degree centrality* [10]. In an undirected graph, the *degree centrality* is defined as the number of links incident upon a node. In a directed graph however, as in our case, the degree centrality is distinguished to the *in-degree centrality* and the *out-degree centrality*.

Definition 3 (Node Centrality). Assume a node $c \in C \cap V_S$ in a dataset $V = (G_s, G_I, \lambda, \tau_c)$. The *in-centrality* $C_{in}(c)$ (respectively, the *out-centrality* $C_{out}(c)$) of c is defined as the sum of the weighted relative cardinality of the incoming $p(c_i, c) \in E_s$ (respectively, outgoing $p(c, c_i) \in E_S$) edges:

$$C_{out}(c) = \sum_{p(c,c_i)\in E_s} RC(p(c,c_i)) * w_p$$

$$C_{in}(c) = \sum_{p(c_i,c)\in E_s} RC(p(c_i,c) * w_p$$

The weights, that are used, are experimentally defined and depend on the type of the properties. We differentiate in our algorithm among two types of properties, the *standard RDF types* (for example "*rdfs:subClassOf"*, "*rdfs:label"*, "*rdfs:comment"*) and *the user-defined properties* (for example the *"P45 consists of"*, *"P128 carries"* shown in Fig. 1). We consider the user-defined ones as more important since they have been explicitly defined by users instead of reusing already existing ones commonly used. In our experiments we used w_p=0.8 for user-defined properties and w_p=0.2 for RDF/S ones.

Consider now the *"E38 Image"* class shown in Fig. 1. Assume also that there are no instances in the corresponding dataset. According to Def. 3, C_{in}(*E38 Image*) = 0.03 since there are 33 edges in the corresponding graph and there are no incoming edges. In addition, C_{out}(*E38 Image*)= 0.03+$RC_{(rdf:type)}$*$w_{rdf:type}$ = 0.03 +0.03*0.2=0.036.

3.1.3. Relevance

The notion of centrality, as defined previously, is a measure that can provide an intuition about how central a schema node in an RDF/S KB is. However, its importance should be determined considering also the centrality of the other nodes as well. Consider for example, the nodes *"E55 Type"* and *"E56 Language"* shown in Fig 1. They have the same number of incoming and outgoing edges. Assume now that they have the same number of instances as well. The *"E55 Type"* is connected to more important elements compared to the *"E56 Language"*. For example, the node *"E18 Physical Thing"* is directly connected to

the *"E55 Type"* and has many other connections and instances. Since the *"E18 Physical Thing"* is obviously a very important node, the *"E55 Type"* is a less appropriate node to represent this area in a summary. On the other hand, the *"E56 Language"* is more relevant than the *"E55 Type"* to represent the specific part of the graph since its neighbors do not have such a high relevance.

To select the most important nodes, we define the notion of the *relevance* of a node, affected by its surrounding neighbors and more specifically by the number and the connections of its adjacent nodes. To be more precise, the formula estimates the (number of) connections of a node and this number is compared to the connections of its neighbors.

Definition 4 (Relevance of a node). Assume a node $c \in C \cap V_S$ in a dataset $V = (G_s, G_I, \lambda, \tau_c)$. Assume also that $p(c_i, c) \in E_S$, $1 \le i \le n$ are the incoming edges of c and $p(c, c_j) \in E_S$, $1 \le j \le k$ are the outgoing edges of c. Then the relevance of c, denoted by $Rel(c)$, is the following:

$$Rel(c) = \frac{C_{in}(c) * n + C_{out}(c) * k}{\sum_{j=1}^{k} \left(C_{out}(c_j)\right) + \sum_{i=1}^{n} \left(C_{in}(c_i)\right)}$$

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. Thus, the number of instances of a node is of vital importance in the assessment procedure. When the data distribution significantly changes, the focus of the entire data source is shifted as well, and as a result, the relevance of the nodes changes. In addition, the importance of each node is compared to the other nodes in the specific area/neighborhood in order to identify the most relevant nodes that can represent all the concepts of a graph. As such, the notion of relevance depicts in essence the capability of a node to represent other nodes.

However, we are not interested only in extracting and presenting the nodes with the highest relevance to the users, but our target is to produce a valid subgraph out of the original one. Next, we focus on selecting the proper edges between the nodes.

4. Construction of RDF/S Summary Schema Graph

Having selected the most important schema nodes, it is now time to focus on the paths that exist in a schema graph. The idea behind this is that we are not interested in extracting isolated nodes, but most importantly, we want to produce valid sub-schema graphs. The chosen paths should be selected having in mind to collect the more relevant nodes by minimizing the overlaps.

Two different algorithms have been created to this direction with different targets: one trying to optimize locally and one globally the importance of the selected paths. Both algorithms exploit blank nodes by allowing them to participate in the calculations, used to establish connections between the nodes. As such, useful information and connections are now maintained and exploited for the construction of the final schema graph summary. Nevertheless, many researchers argue [17] that blank nodes do not offer useful information for understanding an RDF/S graph.

4.1. Sub-graph selection through coverage maximization

In our running example of Fig. 1, the nodes "*E53 Place*" and "*E55 Type*" are directly connected to the node "*E18 Physical Thing*" and have similar connectivity in the graph. The node "*E18 Physical Thing*" has a high relevance in the graph and as a consequence a great probability to be included in the summary. However, although the "*E18 Physical Thing*" can be located only in one place, it might have many types. As a consequence, the relative cardinality of the path from the "*E18 Physical Thing*" to the "*E55 Type*" will be higher than the relative cardinality of the path form "*E18 Physical Thing*" to "*E53 Place*". This means that the path from "*E18 Physical Thing*" to "*E55 Type*" is more appropriate to be included in the summary than the path from "*E18 Physical Thing*" to "*E53 Place*". This is because the "*E18 Physical Thing*" already covers the "*E53 Place*" - a physical thing is located only in one place.

In the above example, we dealt with paths of length one. However, the paths included in the summary should contain the most relevant schema nodes that represent the remaining nodes, achieving the digest of the entire content of the RDF/S KB. Therefore, the main criteria to estimate the level of coverage of a specific path are: a) the relevance of each node contained in the path, b) its relevant instances in the dataset and c) the length of the path. As a result, similar to the approach of Yu et al. [8], we define the notion of c*overage* as follows:

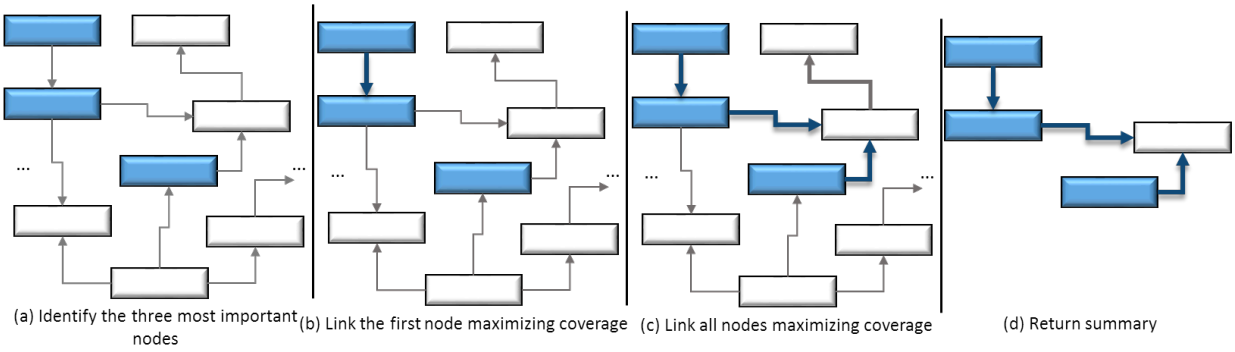

Fig. 2. An example execution of the algorithm for computing the RDF Schema Summary based on coverage

Definition 5 (Coverage of a path). The coverage of a path $path(v_s \rightarrow v_i)$, where v_s, $v_i \in V_S$, denoted by $Cov(path(v_s \rightarrow v_i))$, is the following:

$$Cov(path(v_s \rightarrow v_i)) = \frac{1}{d_{v_s \rightarrow v_i}} * \sum_{j=2}^{d_{v_s \rightarrow v_i}} \left(Rel(v_j) * RC\left(p\left(v_{j-1}, v_j\right)\right)\right)$$

where v_{j-1}, $v_j \in V_S$ *and* $p(v_{j-1}, v_j) \in path(v_s \rightarrow v_i)$.

We can see that we divide by the length of the path in order to penalize the longer paths. The above formula assesses a path and provides a metric to identify the degree of the contained relevant nodes and how this path can represent (a part of) the original graph without overlapping issues. Our goal is 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 relevant this path is considered in representing the original graph or part of it.

Definition 6 (CM Summary Schema Graph of size n): Let $V = (G_s, G_I, \lambda, \tau_c)$ be an RDF/S KB. Let also *TOP* be the n nodes with highest relevance in G_s. *A coverage maximization (CM) summary schema graph* of size n, of V, is a schema graph G_s' having the following properties:

1. $TOP \subseteq G_s' \subseteq Cl(G_s)$
2. $\forall v_i, v_j \in TOP, i \neq j, \exists path(v_i \rightarrow v_j) \in G_s'$
3. $\forall v_i, v_j \in TOP, \nexists path'(v_i \rightarrow v_j) \in Cl(G_s)$ such that $Cov(path'(v_i \rightarrow v_j)) > Cov(path(v_i \rightarrow v_j))$

Now that we have explained all formulas required in order to calculate the relevance and the coverage of the elements of an RDF/S KB, we can describe an algorithm for constructing an RDF/S schema summary that is based on coverage. The algorithm is shown in Fig. 3 whereas the main steps of the execution are shown in Fig. 2. Below we explain in more detail the steps of the corresponding algorithm.

In the beginning, (line 1) we calculate all inferred triples for the schema part of our KB and we construct the corresponding schema graph. As such the summary of an RDF/S KB $V = (G_s, G_I, \lambda, \tau_c)$ would be guaranteed to be always the same as the summary of any KB V' which is obtained from V by applying any number of inferences on G_s. Then the relevance of each schema node is assessed according to Def. 4 (lines 2-3). Having calculated the relevance of each node, we would like to get the n most important ones to be further elaborated (line 4). Usually n is defined by the user. However, if it is left blank this function automatically retrieves a specific percentage of the nodes in the schema (usually 20%). In our example, shown in Fig. 2, we assume that the user asked for a percentage that should return a summary with the three most important nodes. As such, the relevance of all schema nodes is calculated and the three most relevant ones are selected – in blue.

Now we would like to identify the paths that maximize coverage (line 8-14). In other words, we select the paths that contain the most relevant nodes according to the coverage measure as described in the previous section. As such, for each node in TOP we calculate the coverage (Def 5) of the paths connecting that node with the other nodes in TOP selecting the one with the highest value. Note that the selection of the nodes to complete the subgraph is done out of the initial RDF/S schema graph, since the summary should be coherent with the original schema. Moreover, in this selection, other nodes might be also included in the summary in order to connect the most important ones. If there are multiple paths with the same coverage value then the one minimizing the additional nodes introduces is selected. If all paths have the same number of nodes to be introduced then the first returned by the *path_with_max_cov* function is used. Note that in our example of Fig. 2 one addi-

tional schema node is selected to formulate the final summary.

Algorithm 1: *ComputeSummaryCM*(V, n)
Input: An RDF/S KB $V = (G_s, G_I, \lambda, \tau_c)$, n the number of the requested nodes
Output: An RDF/S Schema Summary G_s' *(Nodes,Edges)*
1. B=$Cl(G_s)$
2. *for each* node $v_i \in B$
3. $r_i := calculate\_relevance(B, v_i)$
4. $TOP := select\_top\_nodes(B, r, n)$
5. $Nodes := \{\}$
6. $Edges:=\{\}$
7. $RemNodes := TOP$
8. *while* $RemNodes \neq \{\}$
9. $RemNodes := RemNodes \setminus v_i$
10. $S := path\_with\_max\_cov(B, TOP, v_i)$
11. *for each* node $v_j \in S$
12. $Nodes := Nodes \cup v_j$
13. *for each* edge $p(v_m, v_n) \in S$
14. $Edges := Edges \cup p(v_m, v_n)$
15. Return $G_s'(Nodes, Edges)$

Fig. 3. The algorithm for computing the RDF/S Schema Summary based on coverage

When the algorithm finishes its execution, the selected sub-graph, according to the previous steps, will be a CM Summary Schema Graph. If the data distribution changes, the summary is also changed in order to provide an updated view on the corresponding schema and the updated data instances. The correctness of the algorithm can be easily proved by construction.

Theorem: The Algorithm *ComputeSimilarityCM* produces a CM Summary Schema Graph.

Proof. In order to prove that the result of the execution of the Algorithm 1 is a CM Summary Schema graph we should prove that the three properties from Def. 6 are satisfied. Since we first calculate the relevance of each node (line 1-3), we select the n most important nodes (line 4) and include them in G_s' (line 8) obviously $TOP \subseteq G_s' \subseteq Cl(G_s)$. In addition, since for each two nodes within the TOP we are looking the path maximizing coverage (lines 7-13), they are connected (property 2) with the path with the maximum coverage (property 3). As such, the result of the Algorithm *ComputeSimilarityCM* is a CM Summary Schema Graph.

To identify the complexity of the algorithm we should first identify the complexity of its various components. Assume $|V|$ the number of nodes, $|E|$ the number of edges and $|I|$ the number of instances. Initially we calculate the $Cl(G_s)$ and we need $O(|V|^3)$ using Floyd–Warshall_algorithm [18]. For identifying the relative cardinality of the edges we should visit all instances and edges once. Then for calculating the node centralities we should visit each node once whereas for calculating the relevance of each node we should visit twice all nodes $O(|I|+|E|+2|V|)$. Then we have to sort all nodes according to their relevance and select the top n ones $O(|V|\log|V|)$. Next, we have to calculate the coverage of the paths connecting the selected nodes for each node. This can be done in $O(|E|^2)$. As such the time complexity of the algorithm is polynomial $O(|V|^3) + (O(|I|+|E|+2|V|) + O(|V|\log|V|) + O(|E|^2) \leq O(|V|^3)$.

4.2. Sub-graph selection through relevance maximization

Besides trying to locally optimize the importance of the selected nodes to be included in the summary using coverage, another idea would be to try to optimize the total importance of the edges of the summary graph. To do that we should first define the relevance of an edge as follows:

Definition 7 (Relevance of an edge). Let $p(v_i, v_j) \in E_S$ be the edge connecting the nodes v_i and v_j in a dataset $V = (G_s, G_I, \lambda, \tau_c)$. Then

$$Rel(p(v_i, v_j)) = Rel(v_i) + Rel(v_j)$$

Obviously, the relevance of a path is given by adding the relevance of all edges in the selected path.

$$Rel(path(v_s \rightarrow v_i)) = \sum_{j=s}^{i-1} Rel(p(v_j, v_{j+1}))$$

As such, we can now formally define what a summary schema graph would be, targeting the maximization of the total relevance of the selected schema summary.

Definition 8 (RM Summary Schema Graph of size n): Let $V = (G_s, G_I, \lambda, \tau_c)$ be an RDF/S KB. Let also TOP be the n nodes with highest relevance of B. *A relevance maximization (RM) summ*ary *schema graph,* of size *n,* of V, is a schema graph G_s' having the following properties:

1. $TOP \subseteq G_s' \subseteq Cl(G_s)$
2. $\forall v_i, v_j \in TOP, i \neq j, \exists path(v_i \rightarrow v_j) \in G_s'$
3. $\forall v_i, v_j \in TOP, \nexists path'(v_i \rightarrow v_j) \in Cl(G_s)$ such that $Rel(path'(v_i \rightarrow v_j)) > Rel(path(v_i \rightarrow v_j))$

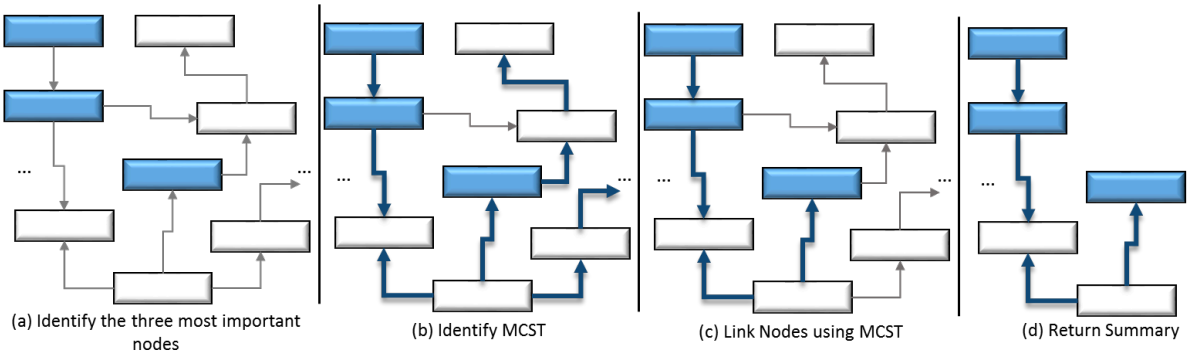

Fig. 4. An example execution of the algorithm for computing the RDF/S Schema Summary based on relevance maximization

Next, we present the algorithm for constructing an RM summary schema graph of a KB. The algorithm is shown in Fig. 4 and similarly to Algorithm 1 gets as input an RDF/S KB V and the number of requested nodes n and returns a corresponding RM summary schema graph. Below we explain in detail the steps of the algorithm execution whereas an example is shown in Fig. 5.

In the beginning (line 1) we calculate all inferred triples for the schema part of our KB and we construct the corresponding schema graph for ensuring that the result will be the same independent of the number of inferences applied to the schema graph G_s. Then, the relevance of each schema node is assessed (calculated using Def. 4) and then the n nodes with the highest relevance are identified. Similarly, to Algorithm 1, n is defined by the user and if left blank this function automatically retrieves a specific percentage of the nodes in the schema (line 4) (20%). In our example shown in Fig. 5, initially the three nodes with the maximum relevance are selected similarly to the previous example.

Algorithm 2: *ComputeSummaryRM*(V, n)
Input: An RDF/S Knowledge Base V, n the number of the requested nodes
Output: An RDF Schema Summary *S(Nodes, Edges)*
1. B=$Cl(G_s)$
2. *for each* node $v_i \in$ B
3. r_i := *calculate_relevance(B,* v_i*)*
4. *TOP := select_top_nodes(B, r, n)*
5. *MCST := max_cost_spanning_tree(B)*
6. *Nodes*:= v_1
7. *for each* node $v_i \in$ *TOP*, i>1
8. *Nodes*:= *Nodes* ∪ *identify_nodes*(*MCST*, v_1, v_i)
9. *Edges* := *Edges* ∪ *identify_edges*(*MCST*, v_1, v_i)
10. Return *S(Nodes, Edges)*

Fig. 5. The algorithm for computing the RDF/S Schema Summary

Then the algorithm tries to identify the paths connecting those nodes by maximizing the total relevance (line 5). In graph theory, a spanning tree T of an undirected weighted graph G is a subgraph that includes all of the vertices of G that is a tree. In general, a graph may have several spanning trees, but the maximum cost-spanning tree (MCST) would be one with the greater total weight. More precisely, a maximum spanning tree for a graph would be a tree connecting all nodes with the maximum total weighting for its edges (where the total weight of the edges in the tree is maximized). In our case, the weight of an edge is defined by its relevance (Def. 7) and as such, the maximum cost-spanning tree would include the more representative path(s). Several algorithms have been proposed for finding the MCST. Kruskal's greedy algorithm [19] is among the fastest ones and we are using it in our implementation. Note that there might be multiple MCSTs in a graph so we use the first one returned by Kruskal's greedy algorithm. In the second step of our example, shown in Fig. 5, the MCST is identified connecting all schema nodes.

Then the algorithm proceeds by isolating the nodes with the highest relevance and connecting them using the paths identified by the MCST thus maximizing the total relevance of the selected sub-graph (lines 6-9). In other words, after the initial identification of the nodes with the highest relevance we connect those nodes by selecting the paths which have the maximum relevance. Similarly, to the previous algorithm, other nodes might be also included in the summary in order to connect the most important ones. In our example of Fig. 5. two additional nodes are included in the schema summary and the final output is returned to the user.

Theorem: The Algorithm ComputeSummaryRM computes an RM Summary Schema Graph.

Proof. In order to prove that the result of the execution of the Algorithm 2 is an RM Summary Schema graph we should prove that the three properties from Def. 8 are satisfied. Since we first calculate the relevance of each node (line 1-3), we select the n most important nodes (line 4) and include them in G_s' (line 8) obviously $TOP \subseteq G_s' \subseteq Cl(G_s)$. Next we calculate the MCST. By definition, it is the tree connecting all nodes maximizing the total relevance of the selected subgraph. As such, both properties 2 and 3 of Def. 8 are satisfied.

The algorithm depends on the data distribution and if it is changed, the summary is also changed in order to provide an updated view on the corresponding schema and the updated data instances.

To identify the complexity of the algorithm, we analyze similarly to Algorithm 1 the complexity of its components. Again for calculating the closure we need $O(|V|^3)$. For the complexity of the lines 2 to 3 we have $O(|I|+|E|+2|V|)$ whereas for sorting all nodes according to their relevance and selecting the top n ones $O(|V|\log|V|)$. Next, we have to identify the MCST with complexity $O(|V|\log|V|)$ and finally to identify the paths between the nodes in TOP using the MCST which again requires to visit once the identified MCST per node. As such the time complexity of the algorithm is polynomial $(O|V|^3)+(O(|I|+|E|+2|V|) + O(|V|\log|V|) + O(|V|\log|V|) + O(n^*|V|))\leq (O|V|^3)$

5. Implementation

The algorithms described in the previous section were implemented in the advanced version of the *RDF Digest* prototype. The system is developed using JAVA and it is currently available online[2] allowing users to use only the second algorithm. Soon it will be updated to allow both the algorithms to be selected and used. The architecture of the system is shown in Fig. 6 and an example summary of the BIOSPHERE ontology using our system is shown in Fig. 7.

The *RDF Digest* is composed of two major components, the *Summarizer* and the *Visualizer*. Using the graphical use interface, a user can select or provide the URL of an online RDF/S document, she would like to be summarized. Optionally she is able to define also the percentage of the most important nodes that she would like to be included in the summary. The *Summarizer* gets the input RDF/S document and preprocesses it (using the *RDF Preprocessor* module) by computing the closure of the corresponding schema graph. The result is stored in a Virtuoso instance to enable efficient data access. Then, the *RDF Assessor* 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 visualizes the returned summary. Besides visualizing the summary schema graph, the user is able to identify several metrics for each node such as the relevance, the in-centrality, the out-centrality, the relative cardinality, the number of properties etc.

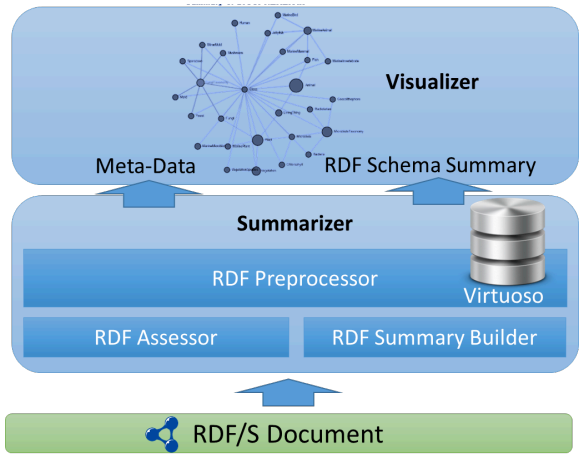

Fig. 6. The architecture of RDF Digest.

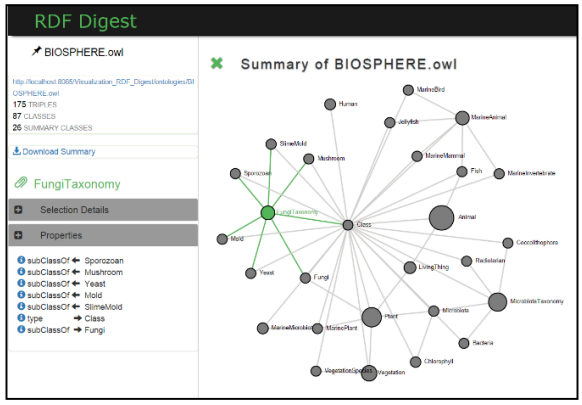

Fig. 7. A screenshot of the RDF Digest prototype.

[2] http://www.ics.forth.gr/isl/rdf-digest

6. Evaluation

To evaluate our system, we used in total six ontologies:

- **BIOSPHERE**[3]: The BIOSPHERE ontology is consisted of 87 classes and 3 properties and models information in the domain of bioinformatics.
- **Financial**[4]**:** The Financial ontology in consisted of 188 classes and 4 properties and describes information on the financial domain.
- **Aktors Portal**[5]**:** The Aktors Portal ontology describes an academic computer science community and is consisted of 247 classes and 167 properties.
- **CRM_{dig}**[6]**:** The CRM_{dig} is an ontology to encode metadata about the steps and methods of production ("provenance") and synthetic digital representations created by various technologies. It is consisted of 100 classes and 361 properties. In addition, for our experiments we used 966 real instances provided by the 3D-SYSTEK[7] project.
- **LUBM**[8]: The Lehigh University Benchmark[9] (LUBM) is a widely used benchmark for evaluating semantic web repositories. It contains 43 classes and 37 properties modeling information about universities and is accompanied by a synthetic data generator. For our tests, we used the default 1555 instances coming from a real dataset.
- **eTMO**[10]. This ontology has been defined in the context of MyHealthAvatarEU project [20] and is used to model various information within the e-health domain. It is consisted of 254 classes and 61 properties and it is published with 3861 real instances coming from the aforementioned project.

The accumulated characteristics of those ontologies are shown in Table 1. The variety on the size, the domain and the structure of these ontologies offers an interesting test case for our evaluation. We have to note that most of these ontologies are actually OWL ontologies (BIOSPHERE, Financial, Aktors Portal, LUBM, eTMO) however we consider only their RDF/S fragment – ignoring also the distinction between TBox and ABox blank nodes.

Table 1. Characteristics of the used ontologies

	Classes	Properties	Instances
BIOSPHERE	87	3	-
Financial	188	4	-
Aktors Portal	247	167	-
CRM_{dig}	100	361	966
LUBM	43	37	1555
eTMO	254	61	3861

We performed an extensive three-stage evaluation to assess the effectiveness of our algorithms:

- **Stage 1**: Initially the first three ontologies are used to compare our algorithms with the algorithms proposed by Peroni et al. [12] and by Queiroz-Sousa et al. [11]. To do that we are using the reference summaries and the results published in [11] and [12]. The Peroni et al. system, automatically defines the key concepts in an ontology, combining cognitive principles, lexical and topological measurements. Queiroz-Sousa et al. on the other hand propose an algorithm that produces an ontology summary in two manners: automatically using relevance measures and semi-automatically, using the users' opinion in addition. We have to note that both the Peroni et al. and the algorithm in Queiroz-Sousa et al. return as a summary only a set of nodes whereas in our case we return an entire graph. As a result, in the first stage of our evaluation we only compare the schema nodes selected by our algorithms with the nodes selected by these two aforementioned works. In addition, the three ontologies used do not contain instances.
- **Stage 2**: To experiment with ontologies containing instances we use the next three ontologies to compare our algorithms with the results from Peroni et al. We tried but could not get access to the system proposed by Queiroz-Sousa et al. [11] to perform the same experiments. In this stage, we conducted a new user-study to construct the reference summaries with and without instances, as we shall see in the sequel, which we used in our evaluation.

[3] http://www.aiai.ed.ac.uk/project/biosphere/downloads.html
[4] http://bit.ly/2e3W6Ct
[5] http://www.daml.org/ontologies/322
[6] http://www.ics.forth.gr/isl/index_main.php?l=e&c=656
[7] http://www.ics.forth.gr/isl/3D-SYSTEK/
[8] http://swat.cse.lehigh.edu/projects/lubm/
[9] http://swat.cse.lehigh.edu/projects/lubm/
[10] http://www.myhealthavatar.eu/

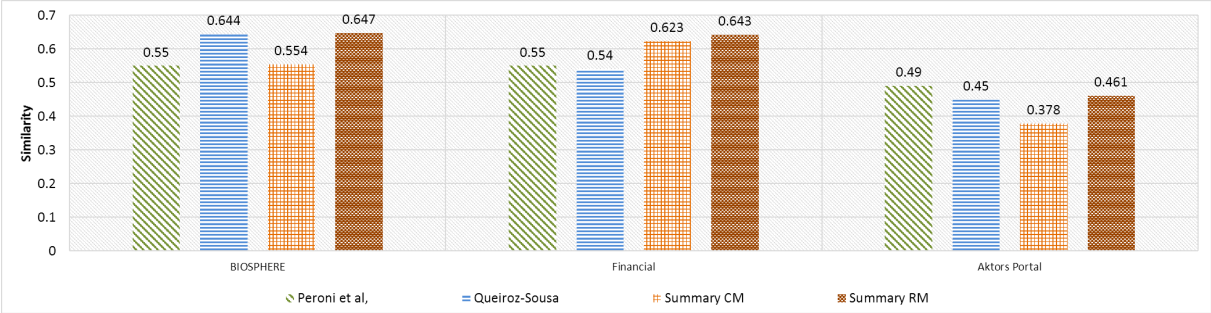

Fig. 8. Stage 1 similarity results.

- **Stage 3**: Since our system is the only system returning an entire graph as a summary, in the last stage of our evaluation we compared as a whole our returned summaries with the reference ones.

Finally, we evaluate the efficiency of our algorithms in terms of execution time, comparing them also with the system proposed by Peroni et al. Below we describe in detail the performed evaluation.

All ontologies used and the reference summaries created by the experts are available online[11].

6.1. Stage 1 Evaluation

6.1.1. Stage 1 Reference Summaries

The reference summaries used in this evaluation stage were generated by Peroni et al. [12] and were also used by Queiroz-Sousa et al. [11] in their evaluation. The reference summaries were generated by eight human experts. These human experts had a good experience in ontology engineering and were familiar with the aforementioned ontologies. The experts were requested to select up to 20 concepts, which were considered as the most representative of each ontology. The level of agreement among experts for the three ontologies had a mean value of 74% [12] meaning that the experts did not entirely agree on their selections.

6.1.2. Stage 1 Evaluation Measures

Measures like precision, recall and F-measure, used by the previous works [10, 11, 20, 21] are limited in exhibiting the added value of a summarization system because of the "disagreement due to synonymy" [23] meaning that they fail to identify closeness with the ideal result when the results are not exactly the same with the reference ones. On the other hand, content-based metrics compute the similarity between two summaries in a more reliable way [10]. In the same spirit, Maedche et al. [24] argue that ontologies can be compared at two different levels: lexical and conceptual. At the lexical level, the classes and the properties of the ontology are compared lexicographically, whereas at the conceptual level the taxonomic structures and the relations in the ontology are compared. To this direction, we use the following similarity measure, denoted by $Sim(G_S, G_R)$, in order to define the level of agreement between an automatically produced graph summary $G_S = (V_S, E_S)$ and a reference graph summary $G_R = (V_R, E_R)$. Assuming $\{c_{k,\dots}, c_p\}$ are the classes in V_R that are subclasses of the classes $\{c'_{k,\dots}, c'_p\}$ of V_S and that $\{c_{m,\dots}, c_n\}$ are the classes in V_R that are superclasses of the classes $\{c'_{m,\dots}, c'_n\}$ of V_S, $Sim(G_S, G_R)$ is defined as follows:

$$Sim(G_S, G_R) = \frac{|V_S \cap V_R| + a * \sum_{i=k}^{p} \frac{1}{d_{p(c_i \to c'_i)}} + \beta * \sum_{i=m}^{n} \frac{1}{d_{p(c_i \to c'_i)}}}{|V_R|}$$

In the above definition α and β are constants assessing the existence of sub-classes and super-classes of G_S in G_R with a different percentage. In our experiments presented below we used α=0.6 and β=0.3 giving more weight to the super-classes. The idea behind that is that the super-classes, since they generalize their sub-classes, are assessed to have a higher weight than the sub-classes, which limit the information that can be retrieved. We experimented with other values as well but although the similarity numbers were different, when comparing systems between each other the results were almost the same and as such they are not presented in this paper.

Consequently, the effectiveness of a summarization system is calculated by the average number of the similarity values between the summaries produced by the system and the set of the corresponding experts' summaries.

[11] http://www.ics.forth.gr/~kondylak/SWJ_2016.zip

Fig. 9. Stage 2 similarity results without (top) and with (bottom) instances.

6.1.3. Stage 1 Comparison

To evaluate the effectiveness of our system we compared the similarity – as defined previously- between the summaries produced by our algorithms and the corresponding reference summaries. The results are shown in Fig. 8.

As we can observe, the summaries generated by our algorithms appear to be quite similar to what experts have produced, in most of the cases, showing better results than other similar systems. Comparing further our two implemented algorithms, we can observe that Summary RM outperforms Summary CM in all cases exploiting the global maximization of the selected edges. The only case that our algorithms are worse than the other two algorithms is in the case of the Aktors Portal ontology. By trying to understand the reasons behind this, we identified that the Aktors Portal ontology contains a huge amount of blank nodes and this has a direct effect to the quality of our constructed summary, despite the fact that both our algorithms consider them when calculating the summary schema graphs. Since the Aktors Portal ontology is an OWL ontology and we only exploit the RDF/S part of the ontology, an interesting experiment would be to consider in addition, the various OWL constructs and to differentiate among ABox and TBox blank nodes. However, we leave this for future work.

6.2. Stage 2 Evaluation

6.2.1. Stage 2 Reference Summaries & Evaluation Measures

The reference summaries used in this evaluation stage were generated by a new user-study. For each one of the CRM_{dig}, the LUBM and the eTMO ontologies three external human experts from the ontology engineering group of Institute of Computer Science at FORTH provided the corresponding reference summaries. These human experts had an extensive experience in ontology engineering and were familiar with the aforementioned ontologies. We have to note that in the second stage of the evaluation the number of experts involved was lower (three) than those used in [12] (eight). The experts were requested to select the most representative *schema graph summary* containing the 10% of the classes for each ontology considering two cases: a) the case that only the schema graph of the ontology is available and b) the case that instances are available as well. As such, for each ontology a human expert had to provide two reference schema graph summaries. The level of agreement among experts for the nodes of the three ontol-

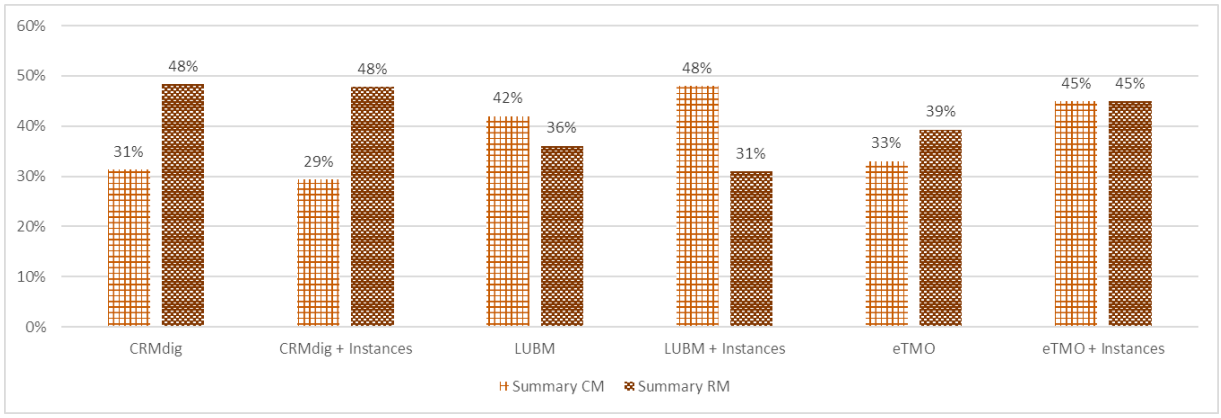

Fig. 10. The average percentage (AP) of triples that should be added to/deleted from our summary to reach the reference summaries.

ogies had a mean value of 73% for LUBM, 33% for eTMO and 34% for CRM_{dig} meaning that the experts did not completely agree on their selections. As such, we calculated the similarity for each expert separately and then we calculated the mean values.

In the second evaluation stage, we use the nodes selected by the experts in the reference graph summary and we compare our algorithms with the Peroni et al. using again the similarity measure.

6.2.2. Stage 2 Comparison

The results of comparing the similarities between the reference summaries and the summaries generated by Peroni et al. and our algorithms are shown in Fig. 9.

We can observe that in all cases both our algorithms outperform Peroni et al. In addition, in most of the cases the Summary RM algorithm outperforms the Summary CM algorithm exploiting the global maximization of the selected paths.

Furthermore, we can notice that when the ontologies contain instances the quality of the selected summaries for both our algorithms increases significantly showing that they effectively exploit instances to understand the important nodes of the schema graph. This is not the case for Peroni et al.

Additionally, our algorithms show better results for ontologies where experts have a better agreement on the generated reference summaries. This is the case for LUBM for example with a mean value of agreement between the experts of 73%.

6.3. Stage 3 Evaluation

Since our system is the only one generating a complete ontology with nodes and properties as a summary, in this section we evaluate as a whole the result of our two algorithms comparing them to the reference summaries generated by experts. We use the ontologies participating in the second stage of our evaluation since the ontology experts were requested to select as a summary an entire schema graph. Peroni et al. returns only nodes as a summary and as such, it is not included in this evaluation stage.

6.3.1. Stage 3 Evaluation Measures.

In [25] the authors argue that low-level deltas can be used to describe the set of triples which were added ($\delta^{+}_{V1,V2}$) or deleted ($\delta^{-}_{V1,V2}$) during the evolution from V_1 to V_2. Thus, the average percentage (AP) of the triples that should be added to or deleted from our generated summaries in order to get the reference summaries can be viewed as a good measure for quantifying the distance in the two summaries and we use it in this section:

$$AP = \frac{\left|V_{summary}\right| + \left|\delta^{+}{}_{summary,reference}\right| - \left|\delta^{-}{}_{summary,reference}\right|}{\left|V_{summary}\right| + \left|V_{reference}\right|}$$

6.3.2. Stage 3 Comparison

The results of our comparison are shown in Fig. 10. As we can observe for summaries generated by our algorithms, at most the 48% of the triples should be changed (this is the case for CRM_{dig} for Summary RM and LUMB with instances for Summary CM). In addition, the average percentage of changes that should be implemented is 37% for Summary CM and 40% for Summary RM showing that the two algorithms generate results of almost the same quality with respect to the triples that should be added or deleted on the generated summary in order to get the reference ones.

Although the generated summaries do not contain the same triples with the reference ones we have to keep in mind that the graphs the experts have in their

mind are really close to the ones that our summaries correspond to. This is shown also by the similarity measure in the previous section.

In addition, we have to keep in mind that even the experts do not agree on the selected references summaries. In fact, many of our experts declared that in many cases it was too difficult to select the paths connecting the most important nodes without being able to argue on which path should be preferred.

6.4. Efficiency

Finally, to test the efficiency of our system, we measured the average time of 50 executions in order to produce summaries of 10% of the nodes using the aforementioned ontologies. We evaluated Peroni et al. and our two algorithms. The experiments run on a 64 bit Windows 7 Enterprise system with 8GB of main memory and an Intel Core 2 Quad CPU running at 2.39 GHz.

The results are shown in Table 2. As we can observe Peroni et al. runs faster than our algorithms. However, this is reasonable since Peroni et al. returns only nodes as a summary whereas in our case the two implemented algorithms have to consider paths as well, returning an entire graph as a summary. In addition, Peroni et al. loads everything in memory whereas our system uses an external triple store to be able to handle mass amounts of data.

Table 2. Execution times for the three algorithms (sec)

	Peroni et al.	Summary CM	Summary RM
BIOSPHERE	1.00	3.07	1.01
Financial	1.34	3.21	2.28
Aktors Portal	1.93	8.68	7.66
CRMdig	1.07	57.34	50.60
CRMdig + Instances	2.00	79.60	73.30
LUBM	0.70	2.42	1.16
LUBM + Instances	1.14	1.35	1.17
eTMO	1.06	7.50	8.20
eTMO + Instances	2.66	8.05	23.19

All algorithms require more time as the size and the density of the ontology increase and in all cases we need even more time when instances are considered as well.

In addition, we can observe that in all but one cases Summary RM is faster than Summary CM. This is because Summary CM has to assess the coverage for each node independently of his neighbors whereas the Summary RM constructs only once the MCST. The only case that Summary RM is slower that Summary CM is the case of eTMO. By carefully examining the aforementioned ontology we can identify that there are properties where the domain and/or the range is not defined. Our algorithm tries to consider those nodes many times in order to construct an MCST. This leads to a significant overhead in execution time. On the other hand, Summary CM simply ignores them. The execution time is increased even more when instances are considered for the same reason (Summary RM requires 23.19 sec for eTMO + Instances whereas Summary CM requires 8.05 sec)

Finally, we can observe that dense ontologies with many properties such as CRM_{dig} require significantly more time than the ones with a small number of properties. This is reasonable since trying to calculate the selected paths is one of the most expensive functions in terms of execution time.

7. Related Work

As already stated, various techniques have been developed for the identification of summaries over different types of schemas and data. The first works on schema summarization focused on conceptual [26] and XML schemas [15]. Yu et al. [8] affirm that, while schema structure is of vital importance in summarization, data distribution often provides important knowledge that improves the summary quality. Another work [27] on XML Schemas derives a summary of the schema and then transforms the instances through summary functions. Other works focus on summarizing meta-data and large graphs. For example, Hasan [21] proposes a method to summarize the explanation of the related metadata over a set of Linked Data, based on user specified filtering criteria and producing rankings of explanation statements.

One of the latest approaches that deals with graph summaries [28] examines only the structure of an undirected graph, neglecting any additional information (such as semantics). The goal of this work is to generate a summary graph that minimizes the loss of information out of the original graph. Furthermore, a wide variety of research works have been focused on producing and visualizing summaries of the datasets, or in other words dataset statistics, without taking into consideration any semantic aspects of the schemata. To this direction Dudas et al. [29], Khatchadourian et al. [29, 30], and Palmonari et al. [32] produce node-link visualization graphs, showing combination of links that reportedly exist in the datasets. However, our system differs from the above in

terms of both goals and techniques. Other approaches try to create mainly instance summaries, by exploiting the instances' semantic associations, by proposing different algorithms that do not take into consideration the schemata of the graphs. To this end, Campinas et al. [33] present several different summary graphs with different instance equivalence criteria for each algorithm. Jiang et.al. [34], Navlakha et al. [35], and Tian et al. [36] propose to construct instance-focused graph summaries of unweighted graphs by grouping similar nodes and edges to super-nodes and super-edges. Although we reuse interesting ideas from these works, our approach is focused towards RDF/S KBs expressing richer semantics than conceptual schemas and XML and single instances.

More closely related works to our data model and approach are [9, 10, 11, 36]. Zhang et al. [10] propose a method for ontology summarization based on the RDF Sentence Graph. The notion of RDF Sentence is the basic unit for the summarization and corresponds to a combination of a set of RDF statements. The creation of a sentence graph is customized by the domain experts who provide as input the size of the summary and their navigation preferences to create the RDF Sentence graph. The importance of each RDF sentence is assessed by determining its centrality in the graph. In addition, the authors compare five different centrality measures (degree, between-ness, PageRank, HITS), showing that weighted in-degree centrality and some eigenvector-based centralities are better. However, in this approach, the overall importance of the entire graph is not considered and many important nodes may be left out.

On the other hand, Peroni et al. [12] try to identify automatically the key concepts in an ontology, combining cognitive principles, lexical and topological measurements such as density and the coverage. The goal is to return a number of concepts that match as much as possible those produced by human experts. However, this work focuses only on returning the most important nodes and not on returning an entire graph summary. In the same direction, Queiroz-Sousa et al. [11] propose an algorithm which produces an ontology summary in two ways: automatically, using relevance measures and, semi-automatically, using additionally the users' opinion (user-defined parameters), producing a personalized ontology summary. However, this work produces summaries which include nodes that are already represented by other nodes.

Pires et al. [22], propose an automatic method to summarize ontologies that represent schemas of peers participating in a peer-to-peer system. In order to determine the relevance of a concept, a combination two measures, *centrality* and *frequency* is used. Wu et al. [37] on the other hand use similar algorithms, named Concept-And-Relation-Ranking, to identify the most important concepts and relations in an iterative manner however, without considering instances.

Although in most of these works the importance of each node is calculated considering each node in isolation, in our work, we assess its importance in comparison with its neighbors, producing a better result. Moreover, many of these works (such as [11] and [21]) do not try to identify how one node represents others and end up collecting nodes already represented by other nodes. In addition, some of these works (e.g. [12]) provide a list of the more important nodes, whereas others [9, 10, 21, 36] and our approach, create a valid summary schema. Our work is the only one that automatically produces a summary graph, exploiting the data instances and essentially provides an overview of the entire KB (both schema and instances).

8. Conclusions and Future Work

In this paper, we present a novel method that automatically produces summaries of RDF/S KBs. To achieve that, our method exploits the semantics of the KB and the structure of the corresponding graph. Based on the notion of relevance, first the most relevant nodes are selected. Then, two algorithms have been implemented trying to identify the edges connecting those nodes trying to maximize edges importance either locally or globally. The performed evaluation verifies the feasibility of our solution and demonstrates the advantages gained by producing high quality summaries. In addition, our approach outperforms in most of the cases other similar systems. Moreover, although most of the systems just select nodes or paths, our result is a valid RDF/S document out of the initial schema graph and can be used for query answering as well.

Currently we are experimenting with extensible summaries. In an ideal scenario, the user should not be limited only to exploring the most important nodes. A user should be able to further explore the components of the summary in order to get more detailed information for a particular part of the original graph. For example, if a user is interested in a specific node she should be able to selectively extend

that summary class getting more detailed information for that particular part of the graph, without being exposed to other unrelated details. This idea can be combined with zooming operations allowing users to request more details on a specific region showing gradually more neighbor connections.

A new direction we intend to explore is how our implementation can be extended in order to produce the schema summary of large schemas in the Linked Data Cloud. Instead of relying on reference summaries for the evaluation of the automatically produced summaries, an interesting idea is to check if these summaries are able to answer the most common queries formulated by the users. Finally, another interesting topic would be to extend our approach to handle more constructs from OWL ontologies such as class restrictions, disjointness and equivalence dropping also the unique name assumption.

As the size and the complexity of schemas and data increase, ontology summarization is becoming more and more important and several challenges arise.

9. Acknowledgments.

The authors would like to thank Maria Theodoridou, Christos Georgis, George Bruseker, Yannis Marketakis and Nikolaos Minadakis for providing the reference summaries, Yannis Roussakis for calculating the changes in Stage 3 evaluation and Ioannis G. Tollis for the useful discussions on graph algorithms.

This work was partially supported by the EU projects DIACHRON (FP7-601043), iManageCancer (H2020-643529) and MyHealthAvatar (FP7-600929).

References

[1] D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, A. Poggi, M. Rodriguez-Muro, R. Rosati. Ontologies and Databases: The DL-Lite Approach. In S. Tessaris, E. Franconi, T. Eiter, C. Gutierrez, S. Handschuh, M Rousset, R.A. Schmidt, eds.: *Reasoning Web. Semantic Technologies for Information Systems, 5th International Summer School 2009, Brixen-Bressanone, Italy, August 30 - September 4, 2009, Tutorial Lectures*, volume 5689 of *Lecture Notes in Computer Science*, pp. 255-356, Springer 2009. 10.1007/978-3-642-03754-2_7.

[2] H. Kondylakis, D. Plexousakis, V. Hrgovcic, R. Woitsch, M. Premm and M. Schüle. Agents, Models and Semantic Integration in support of Personal eHealth Knowledge Spaces. In B. Benatallah, A. Bestavros, Y. Manolopoulos, A. Vakali, Y. Zhang, eds.: *Web Information Systems Engineering - WISE 2014 - 15th International Conference, Thessaloniki, Greece, October 12-14, 2014, Proceedings, Part I*, volume 8786 of *Lecture Notes in Computer Science*, pp. 496-511, Springer, 2014. 10.1007/978-3-319-11749-2_37.

[3] M. Schmachtenberg, C. Bizer, H. Paulheim. *State of the LOD Cloud*. http://linkeddatacatalog.dws.informatik.uni-mannheim.de/state/, (last accessed May 2016).

[4] H. Kondylakis, D. Plexousakis. Ontology Evolution: Assisting Query Migration. In P. Atzeni, D.W. Cheung, S. Ram, eds.: *Conceptual Modeling - 31st International Conference ER 2012, Florence, Italy, October 15-18, 2012, Proceedings*, volume 7532 of *Lecture Notes in Computer Science*, pp. 331-344, Springer, 2012. 10.1007/978-3-642-34002-4_26.

[5] H. Kondylakis, L. Koumakis, M. Psaraki, G. Troullinou, M. Chatzimina, E. Kazantzaki, K. Marias, M. Tsiknakis. Semantically-enabled Personal Medical Information Recommender. In S. Villata, J.Z. Pan, M. Dragoni, eds.: *Proceedings of the ISWC 2015 Posters & Demonstrations Track co-located with the 14th International Semantic Web Conference (ISWC-2015), Bethlehem, PA, USA, October 11, 2015*, volume 1486 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2015. http://ceur-ws.org/Vol-1486/paper_49.pdf.

[6] H. Stuckenschmidt, C. Parent, S. Spaccapietra, eds.. *Modular Ontologies: Concepts, Theories and Techniques for Knowledge Modularization*, volume 5445 of *Lecture Notes in Computer Science*, Springer, 2009. 10.1007/978-3-642-01907-4.

[7] H. Stuckenschmidt, M. Klein. Structure-based Partitioning of Large Concept Hierarchies. In S.A. McIlraith, D. Plexousakis, F. van Harmelen, eds.: *The Semantic Web - ISWC 2004: Third International Semantic Web Conference, Hiroshima, Japan, November 7-11, 2004. Proceedings*, volume 3298 of *Lecture Notes in Computer Science*, pp. 289-303, Springer, 2004. 10.1007/978-3-540-30475-3_21.

[8] C. Yu, H.V. Jagadish. Schema Summarization. In U. Dayal, K. Whang, D.B. Lomet, G. Alonso, G.M. Lohman, M.L. Kersten, S.K. Cha, Y. Kim, eds.: *Proceedings of the 32nd International Conference on Very Large Data Bases, Seoul, Korea, September 12-15, 2006*, pp. 319-330, ACM, 2006. http://dl.acm.org/citation.cfm?id=1164156.

[9] A. Graves, S. Adali, J. Hendler. A Method to Rank Nodes in an RDF Graph. In C. Bizer, A. Joshi, eds.: *Proceedings of the Poster and Demonstration Session at the 7th International Semantic Web Conference (ISWC2008), Karlsruhe, Germany, October 28, 2008*, volume 401 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2008. http://ceur-ws.org/Vol-401/iswc2008pd_submission_66.pdf.

[10] X. Zhang, G. Cheng, Y. Qu. Ontology Summarization Based on RDF Sentence Graph. In C.L. Williamson, M.E. Zurko, P.F. Patel-Schneider, P.J. Shenoy, eds.: *Proceedings of the 16th International Conference on World Wide Web, WWW 2007, Banff, Alberta, Canada, May 8-12, 2007*, pp. 707-716, 2007. 10.1145/1242572.1242668.

[11] P.O. Queiroz-Sousa, A.C. Salgado, C.E. Pires. A Method for Building Personalized Ontology Summaries. *Journal of Information and Data Management*, 4(3):236-250, 2013. http://seer.lcc.ufmg.br/index.php/jidm/article/view/244.

[12] S. Peroni, E. Motta, M. Aquin. Identifying Key Concepts in an Ontology, through the Integration of Cognitive Principles with Statistical and Topological Measures. In J. Domingue, C. Anutariya, eds.: *The Semantic Web, 3rd Asian Semantic Web Conference, ASWC 2008, Bangkok, Thailand, December 8-11, 2008. Proceedings*, volume 5367 of *Lecture Notes in Computer Science*, pp. 242–256, Springer, 2008. 10.1007/978-3-540-89704-0_17.

[13] G. Troullinou, H. Kondylakis, E. Daskalaki, D. Plexousakis. RDF Digest: Efficient Summarization of RDF/S KBs. In

F.Gandon, M. Sabou, H. Sack, C. d'Amato, P. Cudré-Mauroux, A. Zimmermann, *The Semantic Web. Latest Advances and New Domains - 12th European Semantic Web Conference, ESWC 2015, Portoroz, Slovenia, May 31 - June 4, 2015. Proceedings*, volume 9088 of *Lecture Notes in Computer Science*, pp 119-134, Springer, 2015. 10.1007/978-3-319-18818-8_8.

[14] G. Troullinou, H. Kondylakis, E. Daskalaki, D. Plexousakis. RDF Digest: Ontology Exploration Using Summaries. In S. Villata, J.Z. Pan, M. Dragoni, eds.: *Proceedings of the ISWC 2015 Posters & Demonstrations Track co-located with the 14th International Semantic Web Conference (ISWC-2015), Bethlehem, PA, USA, October 11, 2015*, volume 1486 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2015. http://ceur-ws.org/Vol-1486/paper_79.pdf.

[15] D. Brickley, R.V. Guha, eds.. *RDF Schema 1.1*. W3C Recommendation 25 February 2014. http://www.w3.org/TR/rdf-schema/.

[16] G. Karvounarakis, S. Alexaki, V. Christophides, D. Plexousakis, M. Scholl. RQL: a declarative query language for RDF. In D.Lassner, D. De Roure, A. Iyengar, eds.: Proceedings of the Eleventh International World Wide Web Conference, WWW 2002, May 7-11, 2002, Honolulu, Hawaii, pp. 592-603, ACM, 2002. 10.1145/511446.511524.

[17] M. Arenas, M. Consens and A. Mallea. Revisiting Blank Nodes in RDF to avoid the Semantic Mismatch with SPARQL. In *W3C Workshop - RDF Next Steps*, 2010. http://www.w3.org/2009/12/rdf-ws/papers/ws23

[18] Floyd - Warshall Algorithm. Available Online: https://en.wikipedia.org/wiki/Floyd%E2%80%93Warshall_algorithm (last accessed August 2016).

[19] P. Sriram and S. Skiena. *Computational Discrete Mathematics: Combinatorics and Graph Theory with Mathematica®*. Cambridge University Press, 2003.

[20] H. Kondylakis, M. Spanakis, S. Sfakianakis, V. Sakkalis, M. Tsiknakis, K. Marias, Z. Xia, H.Q. Yu, F. Dong. Digital Patient: Personalized and Translational Data Management through the MyHealthAvatar EU Project. In *37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBC 2015, Milan, Italy, August 25-29, 2015*, pp. 1397-1400, IEEE, 2015. 10.1109/EMBC.2015.7318630.

[21] R. Hasan, Generating and Summarizing Explanations for Linked Data. In V. Presutti, C. d'Amato, F. Gandon, M. d'Aquin, S. Staab, A. Tordai, eds.: *The Semantic Web: Trends and Challenges - 11th International Conference, ESWC 2014, Anissaras, Crete, Greece, May 25-29, 2014. Proceedings*, volume 8465 of *Lecture Notes in Computer Science*, pp. 473-487, Springer, 2014. 10.1007/978-3-319-07443-6_32.

[22] C.E. Pires, P. Sousa, Z. Kedad, A.C. Salgado, Summarizing ontology-based schemas in PDMS. In *Workshops Proceedings of the 26th International Conference on Data Engineering, ICDE 2010, March 1-6, 2010, Long Beach, California, USA*, pp. 239-244, IEEE Computer Society, 2010. 10.1109/ICDEW.2010.5452706.

[23] R.L. Donaway, K.W. Drummey, L.A. Mather. A comparison of rankings produced by summarization evaluation measures. In U. Hahn, C. Lin, I. Mani, D. Radev, eds.: *NAACL-ANLP-AutoSum '00 Proceedings of the 2000 NAACL-ANLP Workshop on Automatic Summarization Seattle, Washington - April 30, 2000*, pp. 69-78, Association for Computational Linguistics, 2000. http://dl.acm.org/citation.cfm?id=1567572.

[24] A. Maedche, S. Staab. Measuring similarity between ontologies. In A. Gómez-Pérez, V.R. Benjamins, eds.: *Knowledge Engineering and Knowledge Management. Ontologies and the Semantic Web, 13th International Conference, EKAW 2002, Siguenza, Spain, October 1-4, 2002, Proceedings*, volume 2473 of *Lecture Notes in Computer Science*, pp. 251-263, 2002. 10.1007/3-540-45810-7_24.

[25] V. Papavassiliou, G. Flouris, I. Fundulaki, D. Kotzinos, V. Christophides. High-Level Change Detection in RDF(S) KBs. *ACM Transactions on Database Systems*, 38(1):1:1-1:42, 2013. 10.1145/2445583.2445584.

[26] S. Castano, V. De Antonellis, M.G. Fugini, B. Pernici, Conceptual schema analysis: techniques and applications. *ACM Transactions on Database Systems*, 23(3):286-333, 1998. 10.1145/293910.293150.

[27] J. Marciniak, XML Schema and Data Summarization. In L. Rutkowski, Rafal Scherer, R. Tadeusiewicz, L.A. Zadeh, J.M. Zurada, eds.: *Artifical Intelligence and Soft Computing, 10th International Conference, ICAISC 2010, Zakopane, Poland, June 13-17, 2010, Part II*, volume 6114 of *Lecture Notes in Computer Science*, pp. 556-565, Springer, 2010. 10.1007/978-3-642-13232-2_68.

[28] X. Liu, Y. Tian, Q. He, W.C. Lee, J. McPherson. Distributed Graph Summarization. In J. Li, X.S. Wang, M.N. Garofalakis, I. Soboroff, T. Suel, M. Wang, eds.: *Proceedings of the 23rd ACM International Conference on Information and Knowledge Management, CIKM 2014, Shanghai, China, November 3-7, 2014*, pp. 799-808, ACM, 2014. 10.1145/2661829.2661862.

[29] M. Dudas, S. Svátek, and J. Mynarz. Dataset Summary Visualization with LODSight. In F. Gandon, C. Guéret, S. Villata, J.G. Breslin, C. Faron-Zucker, A. Zimmermann, eds.: *The Semantic Web: ESWC 2015 Satellite Events – Portorož, Slovenia, May 31 – June 4, 2015, Revise Selected Papers*, volume 9341 of *Lecture Notes in Computer Science*, pp. 36-40, Springer, 2015. 10.1007/978-3-319-25639-9_7.

[30] S. Khatchadourian, M.P Consens. ExpLOD: Summary-Based Exploration of Interlinking and RDF Usage in the Linked Open Data Cloud. In L. Aroyo, G. Antoniou, E. Hyvönen, A. ten Teije, H. Stuckenschmidt, L. Cabral, T. Tudorache, eds.: *The Semantic Web: Research and Applications, 7th Extended Semantic Web Conference, ESWC 2010, Heraklion, Crete, Greece, May 30 – June 3, 2010, Proceedings, Part II*, volume 6089 of *Lecture Notes in Computer Science*, pp. 272-287, 2010. 10.1007/978-3-642-13489-0_19.

[31] S. Khatchadourian, M.P Consens. Exploring RDF Usage and Interlinking in the Linked Open Data Cloud using ExpLOD. In C. Bizer, T. Heath, T. Berners-Lee, M. Hausenblas, eds.: *Proceedings of the WWW2010 Workshop on Linked Data on the Web, LDOW 2010, Raleigh, USA, April 27, 2010*, volume 628 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2010. http://ceur-ws.org/Vol-628/ldow2010_paper05.pdf.

[32] M. Palmonari, A. Rula, R.Porrini, A. Maurino,B. Spahiu, and V. Ferme. ASBTAT: Linked Data Summaries with ABstraction and STATistics. In F. Gandon, C. Guéret, S. Villata, J.G. Breslin, C. Faron-Zucker, A. Zimmermann, eds.: *The Semantic Web: ESWC 2015 Satellite Events – Portorož, Slovenia, May 31 – June 4, 2015, Revise Selected Papers*, volume 9341 of *Lecture Notes in Computer Science*, pp. 128-132, Springer, 2015. 10.1007/978-3-319-25639-9_25.

[33] S. Campinas, R. Delbru, and G. Tummarello. Efficiency and precision trade-offs in graph summary algorithms. In B.C. Desai, J. Larriba-Pey, J. Bernadino, eds.: *17th International Database Engineering & Applications Symposium IDEAS '13, Barcelona, Spain - October 09 - 11, 2013*, pp. 38-47, ACM, 2013. 10.1145/2513591.2513654.

[34] X. Jiang, X. Zhang, F. Gao, C. Pu, and P. Wang. Graph Compression Strategies for Instance-focused Semantic Mining. In

G. Qi, J. Tang, J. Du, J.Z. Pan, Y. Yu, eds.: *Linked Data and Knowledge Graph - 7th Chinese Semantic Web Symposium and 2nd Chinese Web Science Conference, CSWS 2013, Shanghai, China, August 12-16, 2013. Revised Selected Papers*, volume 406 of *Communications in Computer and Information Science*, pp. 50-61, Springer, 2013. 10.1007/978-3-642-54025-7_5.

[35] S. Navlakha, R. Rastogi, and N. Shrivastava. Graph Summarization with Bounded Error. In J.T. Wang, *Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD 2008, Vancouver, BC, Canada, June 10-12, 2008*, pp. 419–432, ACM, 2008. 10.1145/1376616.1376661.

[36] Y. Tian, R. Hankins, and J. Patel. Efficient Aggregation for Graph Summarization. In J.T. Wang, *Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD 2008, Vancouver, BC, Canada, June 10-12, 2008*, pp. 567–580, ACM, 2008. 10.1145/1376616.1376675.

[37] G. Wu, J. Li, L. Feng, K. Wang. Identifying Potentially Important Concepts and Relations in an Ontology. In A.P. Sheth, S. Staab, M. Dean, M. Paolucci, D. Maynard, T.W. Finin, K. Thirunarayan, eds.: *The Semantic Web - ISWC 2008, 7th International Semantic Web Conference, ISWC 2008, Karlsruhe, Germany, October 26-30, 2008. Proceedings*, volume 5318 of *Lecture Notes in Computer Science*, pp. 33-49, Springer, 2008. 10.1007/978-3-540-88564-1_3.

[38] H. Seddiqui, R. P. D. Nath and M. Aono. An Efficient Metric of Automatic Weight Generation for Properties in Instance Matching Technique. Journal of Web and Semantic Technology, 6 (1), pp. 1-17, 2015.