# Exploiting Semantic Web Datasets: a Summarisation Based Approach

Honghan Wu[1], Boris Villazon-Terrazas[2], Jeff Z. Pan[1], and Jose Manuel Gomez-Perez[2]

[1] Department of Computing Science, University of Aberdeen, UK,
{honghan.wu,jeff.z.pan}@abdn.ac.uk
[2] iSOCO, Intelligent Software Components S.A., Spain,
bvillazon,jmgomez@isoco.com

**Abstract.** In the last years, we have witnessed vast increase of Linked Data datasets not only in the volume, but also in number of various domains and across different sectors. However, due to the nature and techniques used within Linked Data, it is non-trivial work for normal users to quickly understand what is within the datasets, and even for tech-users to efficiently exploit the datasets. In this paper, we propose a summarisation based approach to guide the exploitation of Linked Data.

## 1 Introduction

So far, Linked Data principles and practices are being adopted by an increasing number of data providers, getting as result a global data space on the Web containing hundreds of LOD datasets [1]. However the technical prerequisites of using Semantic Web dataset prevent efficient exploitations on these datasets. To tackle this problem, in this paper we present a summarisation based approach which can not only provide a quick understanding of the dataset in question, but also is able to guide users in exploiting it in various ways.

The rest of paper is organized as follows: in section 2, we briefly discuss the related work. Then, in section 3 we introduce the details of the summarisation definition and generation. In section 4, we demonstrate several typical data exploitation scenarios based on the information and properties of our summarisation. Conclusions and future work are briefly given in the final section.

## 2 Related Work

There are existing work such as (1) *LODStats*[3] that provides the information related to a dataset, and (2) *make-void* [4] that computes statistics about RDF files. However, LODStats is thought for the whole set of LOD datasets registered in The Data Hub [5], and it is based on declarative descriptions of those datasets; and *make-void* is thought for RDF files but not for RDF datasets.

[3] http://stats.lod2.eu/
[4] https://github.com/cygri/make-void
[5] http://thedatahub.com

Moreover, there are some existing efforts such as Zhang et al.[4] for summarising ontologies based on RDF sentence graphs, and Li et al. [2] for user-driven ontology summarisation. However, both help the understanding rather than the exploitation, which is usually task oriented.

## 3 RDF Summarisation: The EDP Graph

Given an RDF graph, the summarisation is to generate a condensed description which can facilitate data exploitations. Our summarisation method applies a bottom-up strategy to summarise a semantic web dataset. Specifically, we propose an atomic pattern concept in which only one node is involved. Based on this concept, we summarise the given RDF dataset as a new graph which describes the relations between atomic graph patterns.

**Entity Description Block** In an RDF graph, we call its non-literal nodes as entities. For an entity $e$ in an RDF graph $G$, we can get a data block for it by extracting triples in $G$ each of which has $e$ as its subject or object. We call such kind of data blocks as entity description blocks. Formally, each entity $e$ has an entity description block (EDB for short) as defined in Definition 1.

**Definition 1.** *(Entity Description Block)* $\forall e \in G$, *the description block of* $e$ *is defined as*

$$B_e = \{< e, p_i, o_i > | < e, p_i, o_i > \in G\} \cup \{< s_i, p_i, e > | < s_i, p_i, e > \in G\} \quad (1)$$

**Entity Description Pattern** For an entity description block, it can be summarised by a notion of entity description pattern (Definition 2). EDP, the short name for entity description pattern, is the atomic graph pattern in our summarisation model.

**Definition 2.** *(Entity Description Pattern) Given an entity description block* $B_e$, *its description pattern is a tuple* $P_e = (C_e, A_e, R_e, V_e)$, *where*

- $C_e = \{c_i | < e, rdf : type, c_i > \in G\}$ *is called as the class component;*
- $A_e = \{p_i | < e, p_i, l_i > \in G$ *and* $l_i$ *is a literal*$\}$ *is called as the attribute component;*
- $R_e = \{r_i | < e, r_i, o_i > \in G$ *and* $o_i$ *is a URI resource or blank node*$\}$ *is called as the relation component;*
- $V_e = \{v_i | < s_i, v_i, e > \in G\}$ *is called as the reverse relation component.*

Given the $EDB$ notion, an RDF graph $G$ can be represented by a set of $EDB$ i.e., $G = \cup_{e \in G} B_e$. By summarising all entity description blocks in $G$, we can get the intermediate summarisation result of $G$ i.e. $\cup_{e \in G} P_e$ . Given this intermediate result, we define a merge operation on EDPs which can further condense the result (c.f. Definition 3).

**Definition 3.** *(EDP Merge) Given a set of EDPs* $\mathcal{P}$, *let* $\mathcal{C}$ *be the set of all class components in* $\mathcal{P}$ *and let* $G_{\mathcal{P}}(c_i)$ *be a subset of* $\mathcal{P}$ *whose elements share the same class components* $c_i$. *Then,* merge *function can be defined as follows:*

$$Merge(\mathcal{P}) = \{(c_i, \bigcup_{P_i \in G_{\mathcal{P}(c_i)}} Attr(P_i), \bigcup_{P_i \in G_{\mathcal{P}(c_i)}} Rel(P_i), \bigcup_{P_i \in G_{\mathcal{P}(c_i)}} Rev(P_i)) | c_i \in \mathcal{C}\} \quad (2)$$

*where*

- *$Attr(P_i)$ denotes the attribute component of $P_i$;*
- *$Rel(P_i)$ denotes the relation component of $P_i$;*
- *$Rev(P_i)$ denotes the reverse relation component of $P_i$.*

The rationale behind this merge operation is that entities of the same type(s) might be viewed as a set of homogeneous things. Given this idea, we can define an EDP function of an RDF graph as Definition 4.

**Definition 4.** *(EDP of RDF Graph) Given an RDF graph G, its EDP function is defined by the following equation.*

$$EDP(G) = Merge(\bigcup_{e \in G} P_e) \tag{3}$$

**EDP Graph** EDP function of an RDF graph results with a set of atomic graph patterns. Most data exploitation tasks can be decomposed into finding more complex graph patterns which are composed by these EDPs. To this end, it would be more beneficial to know how EDPs are connected to each other in the original RDF graph. Such information can be useful not only in decreasing search spaces (e.g., in query generation) but also for guiding the exploitation (e.g., browsing or linkage). With regards to this consideration, we introduce *RDF data summarisation* as the notion of EDP graph (cf. Definition 5) for characterising the linking structures in the original RDF graph.

**Definition 5.** *(EDP Graph) Given an RDF graph G, its EDP graph is defined as follows*

$$\begin{aligned} \mathcal{G}_{EDP}(G) = &\{< P_i, l, P_j > | \exists e_i \in E(P_i), \exists e_j \in E(P_j), < e_i, l, e_j > \in G, \\ &P_i \in EDP(G), P_j \in EDP(G)\} \end{aligned} \tag{4}$$

*where $E(P_i)$ denotes the instances of EDP $P_i$. Specifically, if $P_i$ is not merged EDP, $E(P_i)$ is the set of entities whose EDP is $P_i$; if $P_i$ is a merged one, $E(P_i) = \cup_{P_k \in P} E(P_k)$, where $P$ is the set of EDPs from which $P_i$ is merged.*

## 4 Demos: The Summary Based Data Exploitations

To evaluate and demonstrate the effectiveness of the summarisation in data exploitation scenarios, we implemented a summary based data exploitation system for three types of tasks i.e., gaining big picture and browsing, generating queries and enriching datasets. The demonstration system is available online at http://homepages.abdn.ac.uk/honghan.wu/pages/kd.wp3/.

The user interface is shown in Figure 1 which contains three panels. The upper part is the *Dataset Selection Panel*, which displays the list of datasets in current demo system. To switch to another dataset, one can simply click on its name in this panel. The middle panel is the main interaction and visualisation panel, the *Data Summarisation Panel*. By default, it displays the summarisation of the selected dataset as an interactive graph i.e., the EDP graph. In other situations, relevant subgraphs of the EDP graph will be shown in the data exploitation process. The right panel is the *Data Exploitation Panel*, which shows a bunch of UI compnents supporting various data exploitation operations.

Given the UI, we now demonstrate a list of data exploitation scenarios to illustrate how the summarisation can help the data exploitation tasks.

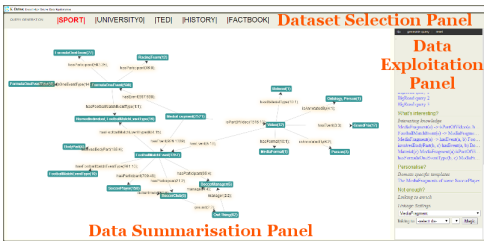


**Fig. 1.** Data Exploitation UI

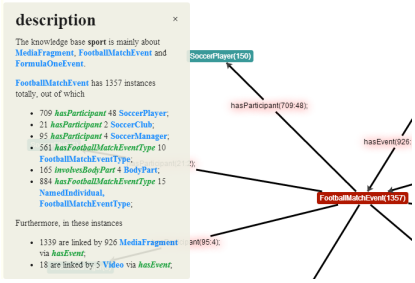


**Fig. 2.** Node Browsing

**The Big Picture and Browsing Operations** When facing an unfamiliar dataset, users usually pursue a quick and rough *big picture* of it before (s)he can assess whether it is interesting or not, e.g, what are the data describing (concepts), how are the main concepts connected to each other (relations) and which are the important parts (clusters). To help the users gain answers to these questions quickly, as shown in the *Data Summarisation Panel* of Figure 1, the EDP graph is visualised by using force-directed graph drawing techniques [6]. Each node in the graph describes a concept. In addition to the concept name, a node is also attached with the number of instances it has in the dataset. Such statistics(c.f. Figure 2) helps to assess the importance of each concept in the dataset (in terms of data portions). The relations between (instances of) these concepts are rendered as edges, and such edges are used to calculate groups of closely related nodes, which are in turn rendered as clusters in the graph.

Two browsing operations are supported on the summary graph. The first is *node browsing*. By clicking on one node in the graph, users can gain detailed description about the concept (c.f. Figure 2) including the subgraph centralised on this node which is displayed in the middle panel and the natural language description of the node displayed in a pop-up panel on the left. The second browsing operation is *graph browsing*. After selecting a node, users can keep selecting/de-selecting interconnected nodes in current subgraph to grow or shrink it. This operation enables focused investigation on relations between interested nodes.

**Query Generation** A typical usage on Semantic Web datasets is querying it. Query generation techniques [3] are helpful for either novice or advanced users because technical skills and dataset knowledge are prerequisites to write SPARQL queries. Based on the EDP summarisation, we implemented two types of query generation techniques. One is called guided query generation, which generates queries by utilising the EDP graph and statistics information attached in the graph. Such technique is good at generating queries for revealing main concepts and relations in the datasets. These two query types are called *Big City*

[6] Arbor Javascript Library (http://arborjs.org/introduction) is used for the EDP graph rendering.

*Queries* and *Big Road Queries* in the *Data Exploitation Panel* of the system. They are analogous to big cities and highways in a geography map. The other generation technique makes use of the links in the summarisation to do efficient association rule mining [3]. This method is good at revealing insightful knowledge in the data in the form of corresponding graph patterns. Such queries are called *interesting knowledge* in the system. Clicking on any of these generated queries will bring out an illustrating subgraph in the middle part of the UI.

**Dataset Enrichment** One of the promising features of Semantic Web techniques is the ability to link data silos to form a more valuable information space. Instead of instance-level linkage or ontology mapping, in our system, we introduce a new data linkage operation on EDPs. Such EDP-level linkage makes it possible to investigate what kinds of possibilities would be enabled after cross-dataset EDPs are linked, e.g., previously unanswerable queries might turn to be answerable by linking another dataset via EDP linkage. In the demo, we will demonstrate EDP-linkage between TED and Factbook datasets and show how such linkage can benefit a specific scenario of filtering tenders by country relations.

## 5 Conclusions and Future Work

We described a dataset summarisation approach, which was shown to be useful for various data exploitation scenarios. The future work will focus on investigating the properties of the summary and in-depth studies in above scenarios.

## References

1. Tom Heath and Christian Bizer. Linked data: Evolving the web into a global data space. *Synthesis Lectures on the Semantic Web Theory and Technology*, 1(1):1–136, 2011.
2. Ning Li and Enrico Motta. Evaluations of User-Driven Ontology Summarization. In Philipp Cimiano and Helena Sofia Pinto, editors, *EKAW*, volume 6317 of *Lecture Notes in Computer Science*, pages 544–553. Springer, 2010.
3. Jeff Z Pan, Yuan Ren, Honghan Wu, and Man Zhu. Query generation for semantic datasets. In *Proceedings of the seventh international conference on Knowledge capture*, pages 113–116. ACM, 2013.
4. Xiang Zhang, Gong Cheng, and Yuzhong Qu. Ontology summarization based on rdf sentence graph. In Carey L. Williamson, Mary Ellen Zurko, Peter F. Patel-Schneider, and Prashant J. Shenoy, editors, *WWW*, pages 707–716. ACM, 2007.