Exploiting Semantic Web Datasets: a Summarization 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]. In this context it is important to promote the reuse and linkage of datasets, and to this end, it is necessary to know the structure of datasets. One step forward for knowing in depth the structure of a given dataset is to provide a summary of the dataset.

In this paper we present fancy-name[3], a simple tool to help users get a quick understanding of RDF dataset ...

The rest of paper is organized as follows: firstly, we introduce the details of the summarisation definition and generation. Then, we present the good properties of this summarisation and propose a set of useful services from the summarisation, both of which can be utilised to guide data exploitation tasks like query writing, ontology reasoning, data compression, and data diagnosis. Finally, we evaluate our approach in several typical data exploitation scenarios. Experiments on real word datasets show that our approach can guide very efficient data exploitation.

2 Related Work

There are available works such as (1) *LODStats*[4] that provides the information related to a dataset, and (2) *make-void* [5] that computes statistics about RDF

[3] http://homepages.abdn.ac.uk/honghan.wu/pages/summ/

[4] http://stats.lod2.eu/

[5] https://github.com/cygri/make-void

files. However, LODStats is thought for the whole set of LOD datasets registered in The Data Hub [6], and it is based on declarative descriptions of those datasets; and *make-void* is thought for RDF files but not for RDF datasets.

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 summarize 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 such 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*

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

where s and p are resources in G.

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, essentially, an RDF graph G is a set of EDB i.e. $G = \cup_{e \in G} B_e$. By summarizing all entity description blocks in G, we can get the intermediate summarization 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 summarization. Definition 3 defines the merge operation on EDPs which share the same class component.

[6] http://thedatahub.com

Definition 3. *(EDP Merge) Given a set of EDPs:*$\{P_i\}_{i=1..n}$ *whose elements have identical class component* C*, we can merge these EDPs into a representative EDP as follows:*

$$Merge(\{P_i\}_{i=1..n}) = (C, \bigcup_{i=1..n} Attr(P_i), \bigcup_{i=1..n} Rel(P_i), \bigcup_{i=1..n} Rev(P_i)) \tag{2}$$

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 set of entities conforms to the EDP* P_i*. If* P_i *is not merged EDP,* $E(P_i)$ *is the set of entities from which* P_i *can be generated; if* P_i *is a merged one,* $E(P_i) = \cup_{P_k \in P} E(P_k)$*,* P *is the set of EDPs from which* P_i *is merged.*

4 Demos: The Summary Based Data Exploitations

Based on the summary method, to evaluate and demonstrate the effectiveness of the summarisation, we implemented a summary based data exploitation system

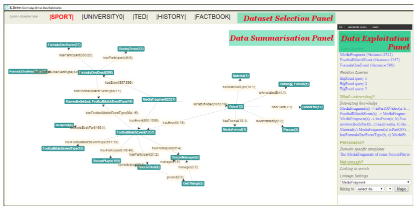
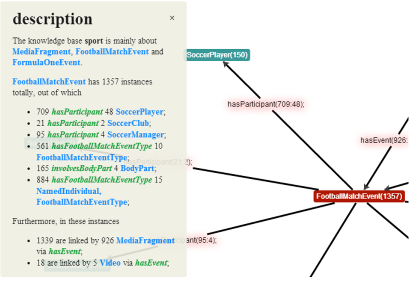


Fig. 1. Data Exploitation UI


description ×

The knowledge base **sport** is mainly about MediaFragment, FootballMatchEvent and FormulaOneEvent.

FootballMatchEvent has 1357 instances totally, out of which

- 709 *hasParticipant* 48 SoccerPlayer;
- 21 *hasParticipant* 2 SoccerClub;
- 95 *hasParticipant* 4 SoccerManager;
- 561 *hasFootballMatchEventType* 10 FootballMatchEventType;
- 165 *involvesBodyPart* 4 BodyPart;
- 884 *hasFootballMatchEventType* 15 NamedIndividual, FootballMatchEventType;

Furthermore, in these instances

- 1339 are linked by 926 MediaFragment via *hasEvent*;
- 18 are linked by 5 Video via *hasEvent*;

Fig. 2. Node Browsing

for three types of data exploitation 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, *Data Summarisation Panel*. The middle panel is the main interaction and visualisation 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 functions to illustrate how the summarisation can help the data exploitation tasks.

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 [7]. 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 closely connected groups 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 and the natural language description of the node displayed in a pop-up

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

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 [2] are helpful for either novice or advanced users because technique skills and dataset knowledge are prerequisite 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 reveal main concepts and relations in the datasets. These two query types are called *Big City 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 association rule mining [2]. This method is good at revealing insightful knowledge in the data in terms 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 be answered with another dataset is linked in 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.

5 Conclusions and Future Work

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. 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.