Exploiting Semantic Web Datasets: A Graph Pattern Based Approach

Honghan Wu[1,3(✉)], Boris Villazon-Terrazas[2], Jeff Z. Pan[1], and Jose Manuel Gomez-Perez[2]

[1] Department of Computing Science, University of Aberdeen, Aberdeen, UK
honghan.wu@abdn.ac.uk
[2] ISOCO, Intelligent Software Components S.A., Madrid, Spain
[3] Nanjing University of Information Science and Technology, Nanjing, China

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 graph pattern based framework for realising a customisable data exploitation. Atomic graph patterns are identified as building blocks to construct facilities in various exploitation scenarios. In particular, we demonstrate how such graph patterns can facilitate quick understandings about RDF datasets as well as how they can be utilised to help data exploitation tasks like concept level browsing, query generation and data enrichment.

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 [2]. 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.

In this demo, we will introduce our graph pattern based exploitation system[1] and demonstrate three exploitation tasks of *(Quick Understanding)* big picture presenting and summary browsing, *(Guided Exploitation)* two query generation methods, and *(Dataset Enrichment)* atomic pattern based dataset linkage.

The rest of paper is organized as follows: in Sect. 2, we briefly discuss the related work. Then, in Sect. 3 we introduce the details of the summarisation definition and generation. In Sect. 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.

[1] http://homepages.abdn.ac.uk/honghan.wu/pages/kd.wp3/

D. Zhao et al. (Eds.): CSWS 2014, CCIS 480, pp. 167–173, 2014.
DOI: 10.1007/978-3-662-45495-4_15

2 Related Work

In this section we describe the related work regarding the consumption and exploitation of Linked Data datasets. We split the related works in two groups (1) works that deals with the extraction of metadata from the datasets; and (2) works related to dataset summarization.

In the following we are going to briefly the related works that extract metadata from available datasets

- *LODStats* [3] provides detailed information of LOD datasets such as structure, coverage, and coherence, for helping users to reuse, link, revise or query dataset published on the Web. It is a statement-stream-based approach for collecting statistics about datasets described in RDF. *LODStats* have smaller memory foodprint, and better performance and scalability.
- *make-void*[2] is another tool that computes statistics for RDF datasets and generates RDF output using the VoID vocabulary. It is based on Jena and features advanced criteria, such as the number of links between URI namespaces, or the number of distinct subjects.
- Holst [5] provides an automated structural analysis of RDF data based on SPARQL queries. He identifies, first, a set of 18 measures for structural analysis. Next, he implements the measures in his RDFSynopsis tool, following two approaches (1) Specific Query Approach (SQA) and Triple Stream Approach (TSA). Finally he performs some evaluations over a set of use cases.
- Bohm et al. [1] developed the voiDgen software to analyze RDF graphs and output statistical data in VoID. They propose to compute additional kinds of class, property, and link-based partitions, e.g., sets of resources connected via specific predicates. Moreover, they show that distributed analyses of RDF large datasets are feasible by using a distributed algorithm (Map-Reduce).

Regarding the works related to dataset summarizaton we can say there are some existing efforts such as A-Box Summary [4] for efficient consistency checking, Zhang et al. [8] for summarising ontologies based on RDF sentence graphs, and Li et al. [6] for user-driven ontology summarisation. However, both help the understanding rather than the exploitation, which is usually task oriented.

3 RDF Summarisation: The Entity Description Pattern

Given an RDF graph, the summarisation task is to generate a condensed description which can facilitate data exploitations. Different from existing ontology summarisation work, we put special emphasises on identifying a special type of basic graph patterns in RDF data, which is suitable for data exploitation. The assumption of this special focus is that there exist such building blocks for revealing the constitution of an RDF dataset in a way which can not only help the understanding of the data but also is capable to guide RDF data exploitation. The rationale behind the assumption is that RDF data exploitation are

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

usually based on graph patterns, e.g., SPARQL queries are based on BGP: basic graph patterns.

The main novelty of our summarisation approach is that it summarises an RDF graph by another much smaller graph structure based on atomic graph patterns. The linking structure in such summary graph can be utilised to significantly decrease search spaces in various data exploitation tasks e.g., query generation and query answering. Furthermore, statistic results of pattern instances are precomputed and attached to the summary, which can help both better understanding about the dataset and more efficient exploitation operations on it.

Specifically, in this paper, we propose one definition of such building blocks, i.e., *Entity Description Patterns* (EDPs for short), which is defined in Definition 1.

Definition 1. *(Entity Description Pattern) Given a resource e in an RDF graph G, the entity description pattern of e is* $P_e = \{C, A, R, V\}$*, in which C is the set of its classes, A is a set of its data valued properties, R is the set of its object properties, and V is the set of e's inverse properties.*

To get a more concise representation of an RDF graph, we define a merge operation on EDPs which can further condense the graph pattern result (c.f. Definition 2).

Definition 2. *(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:*

$$Merge(\mathcal{P}) = \{(c_i, \bigcup_{P_i \in G_{\mathcal{P}(c_i)}} Attr(P_i), \bigcup_{P_i \in G_{\mathcal{P}(c_i)}} Rel(P_i), \bigcup_{P_i \in G_{\mathcal{P}(c_i)}} Rev(P_i)) | c_i \in \mathcal{C}\} \quad (1)$$

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 3.

Definition 3. *(EDP of RDF Graph) Given an RDF graph G, its EDP function is defined by the following equation.*

$$EDP(G) = Merge(\bigcup_{e \in G} P_e) \quad (2)$$

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 can be 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 4) for characterising the linking structures in the original RDF graph.

Definition 4. *(EDP Graph) Given an RDF graph G, its EDP graph is defined as follows*

$$\mathcal{G}_{EDP}(G) = \{< P_i, l, P_j > | \exists e_i \in E(P_i), \exists e_j \in E(P_j), < e_i, l, e_j > \in G, \\ P_i \in EDP(G), P_j \in EDP(G)\} \tag{3}$$

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.

Annotated EDP Graph. EDP graphs are further annotated with statistic results. For each node e, it is annotated with a number which is the number of solutions to $Q(x) \leftarrow C_e(x)$. For each edge $l(C_e, C_f)$, there is a tuple of (n_1, n_2), whose elements are the numbers of solutions to $Q_1(x) \leftarrow C_e(x), l(x, y), C_f(y)$ and $Q_2(y) \leftarrow C_e(x), l(x, y), C_f(y)$ respectively.

4 Demos: The Summary Based Data Exploitations

To evaluate and demonstrate the effectiveness of our definition of data building blocks i.e., EDP, in data exploitation scenarios, we implemented an EDP based data exploitation system for three types of tasks i.e., gaining big picture and browsing, generating queries and enriching datasets.

The user interface is shown in Fig. 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

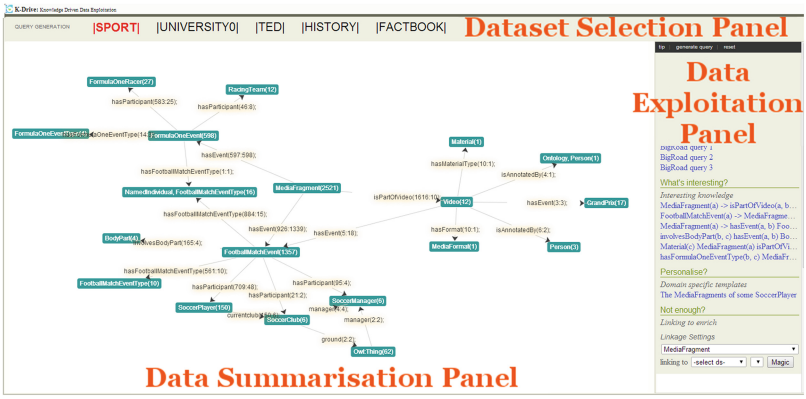

Fig. 1. Data Exploitation UI

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 components 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.

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 Fig. 1, the EDP graph is visualised by using force-directed graph drawing techniques[3]. 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 [7] 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 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

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

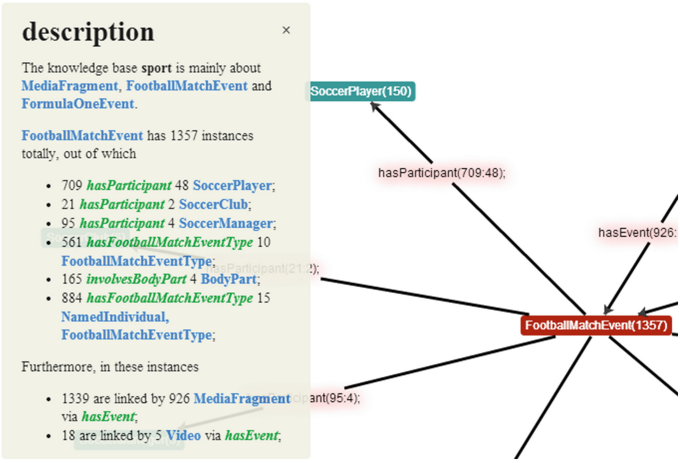

Fig. 2. Node Browsing

association rule mining [7]. 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 graph pattern based approach to facilitate RDF dataset exploitation. The rational behind this approach is to provide LEGO-like building blocks for users to play with RDF datasets. The building block proposed in this paper is so-called EDP, which is a basic entity description graph pattern. We have shown that (i) how EDPs can be constructed as an interactive summary to help quick understanding; (ii) how EDPs can be utilised to generate *interesting* queries under different interestingness definitions; (iii) how EDPs from different sources can be linked to each other so that unanswerable queries are turned to be answerable after this data enrichment. The future work will focus on investigating the properties of the summary and in-depth studies in above scenarios.

Acknowledgement. This research has been funded by the European Commission within the 7th Framework Programme/Maria Curie Industry-Academia Partnerships and Pathways schema/PEOPLE Work Programme 2011 project K-Drive number 286348 (cf. http://www.kdrive-project.eu). This work was also supported by NSFC with Grant No. 61105007 and by NUIST with Grant No. 20110429.

References

1. Bhm, C., Lorey, J., Naumann, F.: Creating void descriptions for web-scale data. Web Semant.: Sci., Serv. Agents World Wide Web **9**(3), 339–345 (2011)
2. Bizer, C., Heath, T., Berners-Lee, T.: Linked data-the story so far. Int. j. semant. web inf. syst. **5**(3), 1–22 (2009)
3. Auer, S., Demter, J., Martin, M., Lehmann, J.: LODStats – an extensible framework for high-performance dataset analytics. In: ten Teije, A., Völker, J., Handschuh, S., Stuckenschmidt, H., d'Acquin, M., Nikolov, A., Aussenac-Gilles, N., Hernandez, N. (eds.) EKAW 2012. LNCS, vol. 7603, pp. 353–362. Springer, Heidelberg (2012)
4. Fokoue, A., Kershenbaum, A., Ma, L., Schonberg, E., Srinivas, K.: The summary abox: cutting ontologies down to size. In: Cruz, I., Decker, S., Allemang, D., Preist, C., Schwabe, D., Mika, P., Uschold, M., Aroyo, L.M. (eds.) ISWC 2006. LNCS, vol. 4273, pp. 343–356. Springer, Heidelberg (2006)
5. Holst, T.: Structural analysis of unknown RDF datasets via SPARQL endpoints. Master thesis defense 11 (2013)
6. Li, N., Motta, E.: Evaluations of user-driven ontology summarization. In: Cimiano, P., Pinto, H.S. (eds.) EKAW 2010. LNCS, vol. 6317, pp. 544–553. Springer, Heidelberg (2010)
7. Pan, J.Z., Ren, Y., Wu, H., Zhu, M.: Query generation for semantic datasets. In: Proceedings of the seventh international conference on Knowledge capture, pp. 113–116. ACM (2013)
8. Zhang, X., Cheng, G., Qu, Y.: Ontology summarization based on rdf sentence graph. In: Williamson, C.L., Zurko, M.E., Patel-Schneider, P.F., Shenoy, P.J. (eds.) WWW, pp. 707–716. ACM (2007)