MINI-LD: Minimalist Consumption over Linked Data

Jeff Z. Pan and Honghan Wu

Department of Computing Science, University of Aberdeen, UK

Abstract. With the wide adoption of linked data idea and its rapid growth in recent 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. While such big semantic data are bringing in much more value than what we have ever seen, the challenges to consume such data also increase dramatically, which result with overwhelmed digital life for linked data consumers. In this paper, we propose the idea of adopting *minimalist consumption* over linked data by borrowing the idea of *less is more* from the popular *minimalist living* style in real life. We introduce our basic idea for such a preliminary *minimalist consumption* system (http://honghan.info/minild/), and describe the main functionalities and features towards this direction.

1 Introduction

"*Do you ever feel overwhelmed, instead of overjoyed, by all your possessions? Do you secretly wish a gale force wind would blow the clutter from your home? If so, it's time to simplify your life!*". These are questions posted on one Amazon page [1] to advocate the idea of a popular book titled "The Joy of Less, A Minimalist Living Guide". As argued in this book, nowadays, many people start to realise that the abundance in materials does not necessarily lead to happier life, and on the contrary, in some cases, it results in the opposite. This is the key point underlying the idea of "minimalist living", which has been gaining popularity and forming a new and fashionable lifestyle.

Looking at what we are facing in the big data era, the situation in such digital world is extremely similar. The oversized, highly-diverse and ever-growing data, which although definitely possesses much more value than we have ever seen, is causing overwhelming challenges in its processing and consumption. Inspired by the idea of *minimalist living*, in the scenario of linked data consumption, one lifestyle could be minimise the resources including the data, the transmission and/or the computation, while fulfilling the requirements. In this paper, we propose MINI-LD, which is our effort of adopting the spirit of *minimalist living* to target the *minimalist consumption* over linked data.

Technically, we break down the *minimalist consumption* into the following aspects.

[1] http://www.amazon.com/gp/product/0984087311?ie=UTF8&tag=missminimalist-20&linkCode=as2&camp=1789&creative=390957&creativeASIN=0984087311

- **minimise the effort of data understanding** The first challenge to have a minimalist living in the big data era is to understand the data so that it is possible to only retrieve the relevant parts. Given the fact the big Volume challenge, efficient understanding approaches are demanded.
- **minimise the data for your living** The key idea of minimalist living is that of only retrieving the relevant without bothering touching other parts. The challenges in this aspect are twofold. On the one hand, how to decompose the original data, and on the other hand how the decomposed parts can be combined to fulfil the requirements especially in the situation of crossing-dataset data consumption.
- **minimise the resource consumption** In addition to minimising the data itself, the resources utilised in the consumption process are also needed to be minimised. Two main resources are directly relevant, i.e., the network broadband and the computation resources. Respectively, the challenges are 1) minimising the data transmission to get the data; and 2) minimising the computation costs including computation time and resources involved.

2 The Basic Idea: *LEGO Block* for Linked Information Space

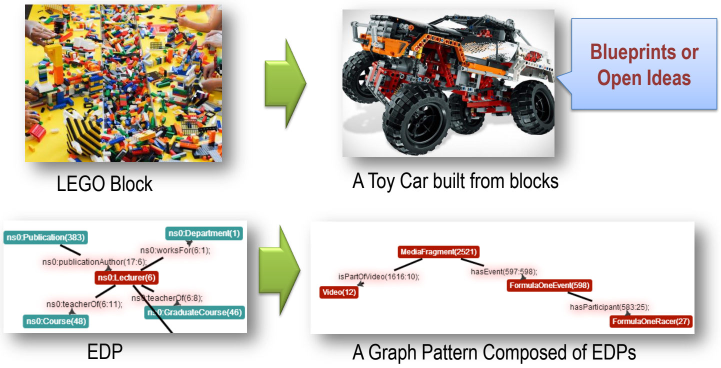
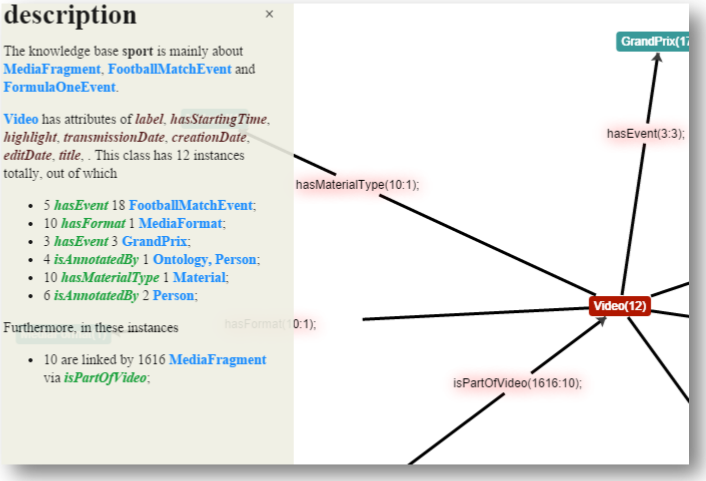


Fig. 1. The LEGO Block for Linked Information Space

The first aspect of *minimalist consumption* is about concise representation of datasets for quick understanding, and the second one is about accurate modelling of data requirements. If we view both datasets and data requirements as information spaces, these two aspects merge into one key challenge: how to represent a linked data information space concisely and accurately.

To tackle this challenge, we apply a summarisation approach to represent information spaces. As shown in Fig. 1, the idea is analogous to the LEGO toy blocks. The same set of LEGO blocks can be utilised to build fancy toys either according to pre-designed blueprints or to open ideas of the player. Similarly, we propose an idea of EDP, entity description pattern, which is the *LEGO building block* for linked data information space. Informally, an EDP is a concise and summarised representation for representing those entities (nodes in an RDF graph) which share the same schema (classes and properties used to describe the entity). Formally, an EDP is a tuple of (C, A, P, R), where C is the set of classes, A is a set of data type properties, P is a set of object properties, and R is a set of inverse properties (displayed as four rectangles in the system cf. Fig. 3). The information space, either a dataset or a data requirement (e.g., a SPARQL query), is analogous to a toy built up from building blocks, i.e., EDPs in our scenario.

Firstly, the EDP summarisation is a concise representation of linked datasets. For example, the EDP summary of DBpedia 3.9(en) dataset is 38.3MB, which is only 0.03% of the original data. Secondly, the users can describe his data requirements by building it from EDPs just like assembling a fancy toy according to one's own idea. Technically, an EDP is a star-shaped graph pattern. A combination of EDPs, a.k.a a data requirement, can be converted to an SPARQL query straightforwardly.



description

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

Video has attributes of ***label***, ***hasStartingTime***, ***highlight***, ***transmissionDate***, ***creationDate***, ***editDate***, ***title***, . This class has 12 instances totally, out of which

- 5 ***hasEvent*** 18 **FootballMatchEvent**;
- 10 ***hasFormat*** 1 **MediaFormat**;
- 3 ***hasEvent*** 3 **GrandPrix**;
- 4 ***isAnnotatedBy*** 1 **Ontology, Person**;
- 10 ***hasMaterialType*** 1 **Material**;
- 6 ***isAnnotatedBy*** 2 **Person**;

Furthermore, in these instances

- 10 are linked by 1616 **MediaFragment** via ***isPartOfVideo***;

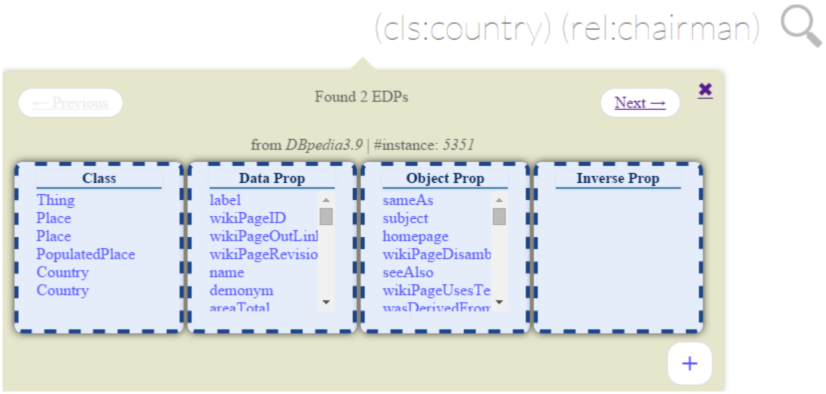
Fig. 2. Dataset Understanding

3.1 Reduce the effort of data understanding

To help the user understand a dataset, we generate an interactive UI to visualise the summary of a dataset as an EDP graph. For example, Fig. 2 shows a subgraph of the EDP graph for a sport dataset. Each node in this graph is an EDP. Users can select a node to inspect the detail information, which is shown in panel on the left of the figure. Initially, the graph is rendered using a force directed graph drawing technique, which can illustrate natural clusters of EDP nodes in a dataset. Such clusters can give user a clear view about the data distribution in concept level. Once a user want to check details of the graph, (s)he can zoom into a part by selecting one node in the cluster. The system will display a subgraph focused on user's selection. By keeping selecting/deselecting nodes, users can browse the graph gradually.

3.2 Reduce the data for your living

As we mentioned in section 2, the data requirement description process is analogous to building a LEGO toy by connecting LEGO blocks together. Our system provides a set of functionalities to help the user assembling such *toys*. In our model, each dataset is an EDP graph as shown in Fig. 2. We know that datasets in the linked data can be linked to each other. For example, there are many published linkages between different datasets as shown in the linked data cloud. Such linkages are used in our system to construct a global EDP graph for the linked data. Although in current system, we only indexed a limited number of popular datasets, given necessary time and resources, we can extend our index to cover most of the linked datasets, if not all. Hence, theoretically, one can use any EDP blocks from the linked data to fulfil her needs.



Found 2 EDPs

from *DBpedia3.9* | #instance: *5351*

Class	Data Prop	Object Prop	Inverse Prop
Thing	label	sameAs	
Place	wikiPageID	subject	
Place	wikiPageOutLinl	homepage	
PopulatedPlace	wikiPageRevisio	wikiPageDisamb	
Country	name	seeAlso	
Country	demonym	wikiPageUsesTe	
	areaTotal	wasDerivedFrom	

Fig. 3. Advanced Keyword Search for EDPs

- **Finding the *EDP* block** We provide a search functionality for users to find EDP blocks. Each EDP is modelled as a tuple of four components, i.e. classes, data properties, object properties and inverse properties. Hence, the search functionality allows the user to specify constrains in all components. In the simplest way, users can specify a keyword query without specifying on which component the keyword should appear, e.g., using *country* to search EDPs describing countries. This would result with 72 different EDPs in current system. In a advanced search, users can specify constrains in components. For example, a search can be *country* in class component and *chairman* in object property component. This will further filter the previous results into 2 EDPs (cf. Fig. 3).

- **Connecting** *EDP* **blocks** A data requirement might have more than one EDPs, which are usually connected to each other. We provide a UI for users to create such connections. Two types of connections are supported. The first type is to create connections by using object properties and inverse properties of two EDPs. This indicates that only originally compatible EDPs can be connected. The second type is customised connections, which are specified by the user manually. If a customised connection is specified by renaming existing properties, this indicates a mapping from the original ontology to user's own ontology. For example, a mapping can say that two persons who have co-authored one paper in DBLP dataset know each other. If a customised connection is created from scratch, this indicate that the user is to create linkages between instances of the two EDPs.

By using these two main functionalities, the user can construct a precise data requirement, which enables reduced data reuse. For example, if only country data is needed from DBpedia, it does not make any sense to download whole DBpedia dataset. Although this sounds naive, unfortunately, many linked data reuses nowadays are conducted in a dataset level. Furthermore, in the demonstration, we will show how EDP approach can support constructing data requirement by combining EDP blocks from various data sources. More interestingly, we will also show how the mapping works in a view-like way.

3.3 Reduce the resource consumption

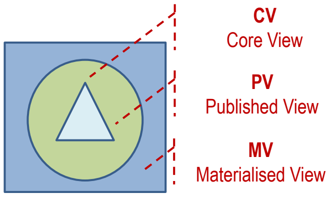


Fig. 4. Three views for modelling the data redundancies

Once users have specified the data requirement, a possible next step could be extract the data from their physical sources for local use. However, the specified data might contain some redundancy, which might lead to unnecessary resource consumption. A redundancy-aware data extraction is needed to achieve more efficient data reuse. Generally speaking, there are three views for published A-Box in the linked data as shown in Fig. 4. The first view is Published View

(PV), which is the data as it is on the web, or in our scenario, the user specified view of the data. The smaller view is called Core View (CV), which is the minimised A-Box by applying semantic compression techniques [3]. The Materialised View(MV) is the largest view got by generating all assertions via T-Box rules. Hence, an efficient data transmission is to extract the CV of the A-Box instead of PV.

To reduce the data transmission, we propose a virtual EDP materialisation approach to explicitly model the three views in linked data consumption. Based on this approach, the system provides services to reduce the semantic redundancies before data transmission. A graph pattern based rule system is used to avoid the extraction of unnecessary data by providing a set of rules for the local side to recover the removed data.

In addition to remove redundancies, our explicit redundancy modelling technique also support the efficient computation by making use of the data redundancies (i.e., the part of PV / CV). As we will show in the demonstration, the relation between data redundancies and the T-Box rule set can be utilised to avoid unnecessary reasoning task in data consumptions like ontology based data access scenarios.

4 Conclusion

Table 1. The Comparison of Linked Data Consumption Methods

Approaches	Usability	Data Modelling	Search	Multiple Datasets	Efficiency
(Semantic) Search Engine	**High**	Inaccurate	Inaccurate	NS	NS
SPARQL Endpoints	Low	**Accurate**	**Accurate**	NS	NS
MINI-LD	**Hight**	**Accurate**	**Accurate**	**Supported**	**Supported**

In this paper, we propose the idea of *minimalism* over linked data consumption. We realise our implementation by using a *LEGO*-like linked data profiling technique. To illustrate the features of our MINI-LD system, we compare it with existing *lifestyles* of linked data consumers in Table 1. The first lifestyle is to use search engines for locating relevant data. In addition to popular web search engines, there are some semantic search engines dedicated to semantic documents (Falcons [1] and Sindice [2]) or semantic web entities (Falcons). The second popular lifestyle is to use structural queries to query SPARQL endpoints. We compare the two with MINI-LD in four dimensions as follows.

- Usability: this is mainly to check how easy it is for end-users to use the approach. It also considers whether prior knowledges or technical background are needed. For example, SPARQL endpoints require the users to know the vocabulary of the dataset and also need basic knowledge of Semantic Web techniques and SPARQL languages. This decreases its usability.
- Data Modelling: this dimension is to check whether the approach's first citizen objects can be used to accurately match data consumers' information

need. For example, search engines are working on documents or entities, which are either too coarse-grained or too fine-grained for matching data consumer's requirements.
- Search: this dimension is to check whether the search/query functionalities can accurately represent users's needs.
- Multiple-dataset: this dimension is about whether the approach can seamlessly support data consumption from multiple datasets. Instead of only checking the information sources are diverse, this dimension is more about the support of combining data units from multiple sources to fulfil requirements. For example, search engines can get a list of results from multiple sources. However, it leaves the task of extraction and combination of distributed data units to the users. Hence, it does not support multiple datasets.
- Efficiency: this dimension is to check whether the system can support efficient data consumption, e.g., reduce data size for downloading or avoid unnecessary computation.

The results of comparison are listed in Table 1. According to the comparison, MINI-LD is superior in several ways for consuming linked data. This qualitative analysis suggests that *Minimalist Consumption* can be a promising way to achieve efficient data consumption in big semantic web era.

5 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.

6 Appendix

In the following Table 2 and 3, we summarise how we fulfill each of the requirements of the Big Data Track.

References

1. Gong Cheng, Weiyi Ge, and Yuzhong Qu. Falcons: searching and browsing entities on the semantic web. In *Proceedings of the 17th international conference on World Wide Web*, pages 1101–1102. ACM, 2008.
2. Eyal Oren, Renaud Delbru, Michele Catasta, Richard Cyganiak, Holger Stenzhorn, and Giovanni Tummarello. Sindice. com: a document-oriented lookup index for open linked data. *International Journal of Metadata, Semantics and Ontologies*, 3(1):37–52, 2008.
3. Jeff Z. Pan, Jose Manuel Gomez-Perez, Yuan Ren, Honghan Wu, Haofen Wang, and Man Zhu. Graph pattern based rdf data compression. In *Proc. of 4th Joint International Semantic Technology Conference (JIST)*, page (To Appear).

Table 2. Minimal Requirements

Criterion	Rating	Explanation
Data Volume	High	Summarised DBpedia 3.9 (en) datasets plus many other medium and small sized datasets including DBLP2013, linkedMDB and etc. Total triples processed at this moment is near 1 Billion. However, we are keeping processing datasets. Hopefully, at the time of presentation, we will be able to index more than 10 billion triples.
Data Variety	Medium	For heterogeneous data sources, as shown in the comparison in the conclusion section, MINI-DL is superior in dealing with heterogeneous and previous unknown schemas. This is supported by our summarisation based approach. The limitation in this regard is that we are mainly focusing on linked data.
Data Velocity	High	The framework is a EDP based approach, which summarising the data from the schema level. Hence, if the data changes mainly in the instance level, it is not necessary to update the summary. Only the statistics needs to be updated. If new schema comes, it is similar to processing new datasets. The main job is to expand the EDP index, which is already implemented in the system. If we have a chance, we will present the statistics about how we deal with fast-changing data in three weeks time.

Table 3. Additional Desirable Features

Criterion	Rating	Explanation
Usability	High	We designed a set of visualisation functionalities for data understanding, data requirement composition, redundancy analysis and other functions in data consumption tasks. Users can use the system without any knowledge about the datasets and the semantic web techniques.
Value	High	The EDP graph based Linked Dataset profiles are very useful for data users to find useful datasets. In addition, it supports fine-grained data reuse, which has not been seen in the community. This will enable very efficient data reuse. Furthermore, the customised mapping can generate useful linkages to the linked datasets.
Functionality	High	Our system aims at efficient linked data consumption, which are broken down to three subcategories. The first functionality is summary based data understanding. The second is EDP based requirement composition. And, the last is the redundancy-aware data consumption. Each of them is supported by a set of visualisation UIs and services.