J-GLOBAL knowledge: Japan's Largest Linked Open Data for Science and Technology

Takahiro Kimura[1], Takahiro Kawamura[1], Katsutaro Watanabe[1], Naoya Matsumoto[1], Tomonori Sato[1], Tatsuya Kushida[2] and Katsuji Matsumura[1]

[1] Japan Science and Technology Agency
[2] National Bioscience Database Center, Japan

Abstract. In order to develop innovative solutions in science and technology, Japan Science and Technology Agency (JST) has published J-GLOBAL knowledge (JGk), which provides papers, patents, researchers' information, technological thesaurus, and scientific data, which have been accumulated by JST since 1957 as Linked Open Data. The total of all datasets surpasses 15.5 billion triples, and the JGk website provides a SPARQL endpoint to access the datasets. This paper describes several issues on schema design to construct such a large-scale Linked Data, and methods for linking to external datasets like DBpedia. Finally, we indicate performance problems and the future works.

1 Introduction

According to a design of knowledge infrastructure in the 4th Science and Technology Basic Plan of Japan[3] in 2011, Japan Science and Technology Agency (JST) has distributed credible, high-quality data for science and technology with licenses accessible to the public and machine-readable format, and then promoted its use to more researchers and developers with the aim of development of innovative solutions. Thus, JST released J-GLOBAL knowledge[4] on May 28, 2015, which includes part of Linked Open Data (LOD) for science and technology information, such as papers, patents, researchers' information, technological thesaurus, and scientific data. The size of the datasets surpasses 15.5 billion triples.

Nowadays, most researchers and developers use Internet search engines like Google to find scientific and/or technological information. The results of the search engines, however, occasionally include untrustable noisy data, and might not include useful data in the Deep Web. But DB services for patents and papers, such as PatFT[5], SCOPUS[6] are unfamiliar to the ordinary users, and pose difficulty for cross-category retrieval, such as through patents and papers. Thus, JGk put a common identifier to entities of multiple categories, and then linked each other. As a result, when a user becomes interested in a paper, she/he can find information about an author of the paper, then also reach to patents applied

[3] `www.gov-online.go.jp/eng/publicity/book/hlj/html/201201/201201_08.html`
[4] `stirdf.jglobal.jst.go.jp`, `datahub.io/dataset/j-global-knowledge` (for English)
[5] `patft.uspto.gov`
[6] `www.scopus.com`

Table 1. Categorization of J-GLOBAL knowledge (as of June 1, 2015)

Category	Content	# of content	# of triples
Researcher	Researchers belonging to univ. or institution in Japan	240k	124,237,623
Paper	Papers of major scientific, engineering, medical and pharmacological journals and conferences in the world	36,260k	13,175,534,543
Patent	Patent applications published by the Patent Office in Japan	11,510k	2,278,276,314
Subject	Research subjects supported by funding agencies	60k	6,332,982
Organization	Universities, public offices, institutions, and companies in Japan	330k	11,469,063
Technical thesaurus	Thesaurus with synonyms of technical terms	1,100k terms	6,464,398
Chemical material	Organic compound information	3,370k	120,283,251
Bio material	Human gene information	220k	3,747,247
Article	Scientific and technological articles, other than papers	220k magazines	4,052,113
Resource	Research databases in univ. and institutions in Japan	5k	128,640
Total			15,730,526,174

by the author. JGk aims to derive ideas and clues by tracing the relationship of the scientific and/or technological entities through multiple categories (free of charge, but there are paid contents in external links). The descriptions of the categories are shown in Table 1.

The rest of the paper is organized as follows. Section 2 briefly introduces methods for building large-scale LOD sets, and then Section 3 describes issues on schema design and triplification of JGk. Finally, Section 4 indicates performance problems and the future works.

2 Related Work

In DBpedia, which is the *de facto* hub of LOD, DBpedia Information Extraction Framework is used for extracting infobox templates from dump data of Wikipedia, and ontology mapping, which is a voluntary-based collaborative tool is used for mapping items in templates to properties in Linked Data. Yet Another Great Ontology (YAGO) is a knowledge base with 120 million triples, composed of categories, redirects and infoboxes of Wikipedia, synonyms and hyponyms of WordNet, and GeoNames. But properties are manually defined by experts. In JGk, we carefully designed schemas with experts like YAGO, and linked entities to external datasets like DBpedia in order to provide credible, high-quality data for science and technology.

3 Design of J-GLOBAL knowledge

3.1 Schema issues

According to four principles of Linked Data outlined by Tim Berners-Lee, all the entities in the categories are represented as resources (Uniform Resource Identifiers, URIs), and thus can be dereferenced as much as possible. We broadly investigated related schemas with Linked Open Vocabularies, and then reused common schemas, which are already adopted in external datasets. Then, as necessary, we defined our own properties considering reusability.

Flat model. Resource Description Framework (RDF) provides high-flexibility of design, and then schemas occasionally get complicated to reconstruct original structures in Relational Databases (RDBs). We thus attempted to adopt flat models of schemas, as users can intuitively imagine the schema structure.

Disclosure/Nondisclosure. Dataset for each category also has metadata of the dataset for management like modification date. Contents of the datasets are open to the public, but the metadata are closed by distinguishing each property with `jst:disclosure-segment`.

Use of literals. Although entities should have URIs in Linked Data, rich literal values enhance convenience for keyword searches. Therefore, we described properties of importance with object properties with URIs, and datatype properties with literals, such as `rdfs:label, dc:title, foaf:name`. As a result, we had to put a blank node to describe the two properties in parallel.

Representation of ordered list. Although the order has valuable information such as in author lists of papers, RDF is a graph model, and then the order of triples in RDF/XML or Terse RDF Triple Language (Turtle) formats are not preserved. Therefore, there are several notations to keep the order, such as 1. RDF List (`rdf:List`), 2. RDF Container (`rdf:Seq`) with `rdf:_1, rdf:_2` etc., and 3. structural combination of blank nodes and the order numbers. In our schema, we adopted 1. RDF List as recommended by W3C for now, although the notation is slightly complicated and there is no standard way to search by SPARQL. However, most services using Linked Data expect flat models, and thus we also described lists by 2. RDF Container as a popular enumeration notation. 3. is the simplest way, but not standardized and depends on specific schema implementations. Also, since most of common properties cannot take an RDF List as an object, we redefined the properties of the same meaning with our namespace `jstd:`.

3.2 Triplification issues

We first outputted tables in PostgreSQL as key-value pairs in JavaScript Object Notation (JSON) format, and then transformed them to Turtle format based on the above schema policies.

Literal matching in datasets. To link the datasets in JGk with each other, literal node matching has been conducted. Literals to be matched are bibliographic information, such as researchers' and organizations' names in 36,260k papers and 11,510k patents. We first normalize the literals and then calculate weighted sum of similarities between attributes of the literals. In the case of researcher names in the papers, the attributes are co-authors, research keywords, affiliations, journal and/or conference names, etc. The similarity of attributes is calculated based on Longest Common Subsequence and the predefined dictionary. We divided all the papers into several datasets by the date of publication while partially overlapping, and then created pairs of the literals. If the similarity of a pair is higher than a threshold, we put the same id to the pair. The preliminary evaluation using a sampling approach showed 98.6% precision and 90.8% recall for researcher names, and 95.3% precision and 95.0% recall for organization names. Figure 1 shows the relationship of datasets in JGk.

Link to external datasets. To link the datasets in JGk to external datasets, we set a resource type (class in ontology) to a main entity in the categories.

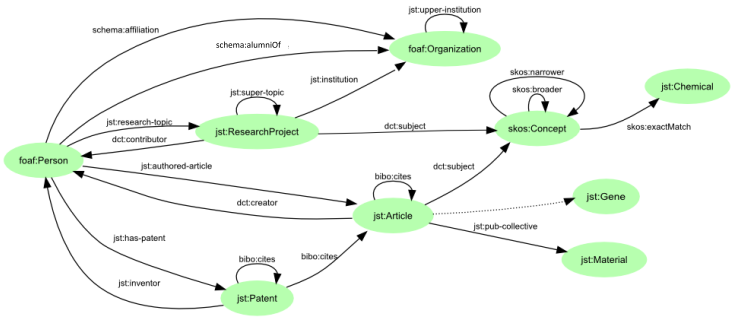

Fig. 1. Relationship among all categories

Table 2. Types and links to external datasets in each category (JSPS: Japan Society for the Promotion of Science, JWO: Japanese Wikipedia Ontology)

Category	Type	Property	External dataset
Researcher	foaf:Person	kaken:researcherNumber	JSPS researcher ID
Paper	jst:Article	prism:doi, bibo:pmid	DOI, ISSN, PubMed
Patent	jstpat:Patent	jst:ipc, etc.	Patent Office DB
Subject	jst:ResearchProject		
Organization	foaf:Organization		
Technical thesaurus	skos:Concept	foaf:page, skos:closeMatch	Wikipedia, DBpedia
Chemical material	jst:Chemical	jwo-infobox:CASNo	JWO
Bio material	jst:Gene	hgnc:xEntrezGene, bio2rdf:pubmedID	Entrez GeneID, PubMed
Article	jst:Material	prism:issn, bibo:coden	ISSN, CODEN

Also, we prepared several properties as links to external datasets. Table 2 shows the type, the properties to external datasets, and the corresponding datasets in each category. As a result, users can search on the datasets with type constraints, and search through the databases including the external datasets. If the external datasets have identifiers in RDF, we linked to the ids. However, in the case that the official dataset is in HTML, and the corresponding dataset in RDF is created by the third party, we individually determined which dataset should be linked.

4 Conclusion and Future Work

This paper introduced a large-scale Linked Open Data for science and technology in Japan. As the future work, we intend to prepare Web APIs for analysis of the datasets including external sources.

The server is currently running on 80 VCPU, 1 TB memory, 3.6 TB HDD on CentOS 6.3 with Virtuoso 7, and simple queries to retrieve specific resources get the results in 7 (ms) with multiplicity 10. The performance almost remains with more queries. However, complicated queries like aggregation, sorting, string matching return the results in 20 (s)–60 (s) with multiplicity 10, and the performance becomes worse according to the multiplicity. Loading of all the triples required 12 days, but we confirmed parallel processing improved the performance. Dumping of all the triples required 3 days. In the near future, we intend to triplify datasets of Web of Science and SCOPUS, which will become 180 billion triples and then require several months for loading. Thus, we need to address deletion of redundant triples and parallel processing of queries.