

RDF Graph Visualization by Interpreting Linked Data as Knowledge

Rathachai Chawuthai^{1,2}(✉) and Hideaki Takeda^{1,2}

¹ SOKENDAI (The Graduate University for Advanced Studies),
Kanagawa, Japan

{rathachai, takeda}@nii.ac.jp

² National Institute of Informatics, Tokyo, Japan

Abstract. It is known that Semantic Web and Linked Open Data (LOD) are powerful technologies for knowledge management, and explicit knowledge is expected to be presented by RDF format (Resource Description Framework), but normal users are far from RDF due to technical skills required. As we learn, a concept-map or a node-link diagram can enhance the learning ability of learners from beginner to advanced user level, so RDF graph visualization can be a suitable tool for making users be familiar with Semantic technology. However, an RDF graph generated from the whole query result is not suitable for reading, because it is highly connected like a hairball and less organized. To make a graph presenting knowledge be more proper to read, this research introduces an approach to sparsify a graph using the combination of three main functions: graph simplification, triple ranking, and property selection. These functions are mostly initiated based on the interpretation of RDF data as knowledge units together with statistical analysis in order to deliver an easily-readable graph to users. A prototype is implemented to demonstrate the suitability and feasibility of the approach. It shows that the simple and flexible graph visualization is easy to read, and it creates the impression of users. In addition, the attractive tool helps to inspire users to realize the advantageous role of linked data in knowledge management.

Keywords: Graph simplification · Knowledge representation · Linked data · RDF visualization · Semantic web application · Triple ranking

1 Introduction

It is known that Semantic Web and Linked Open Data (LOD) technologies aim to enable the connection among pieces of data around the world, and turn them into a global knowledge space [1]. For this activity, Resource Description Framework (RDF) becomes a standard for representing explicit knowledge. Thus, all pieces of knowledge in every repository are expected to be stored in RDF format in order to have data be exchangeable and linkable across repositories via the Internet. At the moment, many organizations such as research institutes, governments, and industries start opening their own data. More local data are continuously interconnected through the LOD cloud. Thus, it can say that we are in the age of the growing world knowledge management system [1–3].

Many pieces of research regularly manage RDF at the data tier in order to improve searching ability, because the advantages of knowledge representation and knowledge reasoning can construct rich machine-readable data in the form of a graph of knowledge [4]. Large amount of connected data are required, however, RDF data are mostly provided by tech users [5] or ones who know Semantic Web. Encouraging lay users [5], or ones who have less knowledge about Semantic Web, to contribute RDF data is very challenging, because they never realize how linked data work and RDF syntax itself is not user-friendly [6, 7]. It is resulted in a barrier between human and linked data.

For this reason, RDF data should be located not only at the data tier but also at the presentation tier in order to have users be familiar with Semantic Web. In this case, we question, “How users can access linked data in a suitable way?” Since a concept-map or a node-link diagram can enhance the learning ability from beginner to professional level, RDF graph visualization becomes a suitable way for enabling users to learn knowledge described in RDF and making them appreciate the role of linked data in knowledge management [8–10]. However, converting RDF data into an easily-readable graph visualization is difficult due to a lot of issues caused by the behaviors of RDF data together with the reasoning results. As we analyzed data, we found some significant issues. First, the graph is highly connected like a hairball due to inferred data, so it is too hard to be read by users. Second, since there is no ordering to the triples in an RDF graph, it is interrupting the flow of reading of users who pay attention to gain knowledge from a graph. Last, users are not convenient to focus what they want due to a large number of data presented.

This research aims to offer an approach to the presentation of RDF graph visualization as a learning tool by interpreting RDF data as knowledge structures. The following features are initiated to address the mentioned problems.

- Graph Simplification: To simplify a graph by removing some redundant triples that are resulted from ontological reasoning processes.
- Triple Ranking: To give a ranking score of each triple from common information (background content) to topic-specific information (main content), and to allow users to filter a graph based on this score.
- Property Selection: To allow users to filter a graph by selecting some properties in order to display or hide some triples.
- User Interaction: To control the above operations according to user demand.

This paper is organized as follows. The background and motivation are introduced in this section. Related work is reviewed in Sect. 2. The data are analyzed in Sect. 3. Our approach is described in Sect. 4. The prototype is demonstrated in Sect. 5. The outcome is discussed in Sect. 6. Last, the conclusion is drawn in Sect. 7.

2 Literature Review

There are pieces of research that work on the issue about RDF visualization. They aimed to operate a complex network in any visualization canvas to be friendly for general users.

We first reviewed some network visualization tools. Motif Simplification [11] considered some topologies of subgraphs, and replaced them with basic shapes such as diamonds, crescent, and tapered diamonds. It intended to give a big picture of a network rather than the detail of node-link. Gephi Open Viz Platform [12] is a powerful visualization tool that generated a well-shaped layout of network, allowed users to filter nodes and links, and had an option to set colors according to user preference. Both tools are suitable for general networks, but they are not designed for dealing with RDF data.

One important issue of RDF data is a large number of inferred links creating a hairball-like graph, so the tools should consider this behavior in order to simplify a graph. RDF Gravity [13] provided an interactive view. Users could zoom a graph to view much more detail, and get details of nodes in the focus area using text overlay. Next, Fenfire [14] gave an alternative view of RDF. It displayed the full details of the focused node and its immediate neighbors, but the other links were faded away according to the distance from the focus node. Both RDF Gravity and Fenfire offered well-organized displays, but they do not point out the issue of redundant data from inferred triples. Moreover, IsaViz [15] is an interactive RDF graph browser that used graph style sheets to draw a graph. It provided meaningful icons describing the type of each node such as *foaf:Person*, and grouped its metadata into a table in order to reduce highly interlinked data. It also allowed users to filter some nodes or properties to sparsify a dense graph, but this task required human effort to select some preferred URIs one by one.

The other issue is about the readability of RDF data, because RDF data are not well arranged for reading from introduction to main contents. Some works target to rank query triples. Several approaches used Term Frequency- Inverse Document Frequency (TF-IDF) to extract keywords from a content [16, 17]. PageRank [18] gave a score to each page by estimating the number of links and the quality of neighbors. TripleRank [19] ranked query result by applying a decomposition of a three-dimensional tensor that is originated by HITS [20] to retrieve relevant resources and predicates. Ichinose [21] employed the idea of TF-IDF to identify how important of resources and predicates of each subject under the same classification for ranking the query result. Nevertheless, they did not discuss about how to order triples for supporting the readability of users.

3 Preliminary Data Analysis

This research views that besides storing RDF data at the data tier, they should be presented at the visualization tier in order to have users to realize how importance of linked data in knowledge management. Using graph visualization for presenting knowledge is a suitable way for users to read and understand Semantic Web data [8, 10].

The well-displayed graph visualization should be simple and sparse [22]. In other words, it should be similar to the original RDF data; however, a query result contains both raw RDF data and inferred data because of the manner of a SPARQL engine. In this case, querying a graph by accessing the whole neighborhood of a given node within two hops is recommended to be a general input for this research due to the following scenario. Let *bt* be a short-hand writing of the transitive property named *skos:broaderTransitive*, if raw RDF data are $\langle \textit{Dog}, \textit{bt}, \textit{Mammal} \rangle$, $\langle \textit{Mammal}, \textit{bt}, \textit{Animal} \rangle$, and $\langle \textit{Animal}, \textit{bt}, \textit{LivingThing} \rangle$; the inferred triples can be $\langle \textit{Dog}, \textit{bt}, \textit{Animal} \rangle$,

$\langle \text{Dog}, bt, \text{LivingThing} \rangle$, and $\langle \text{Mammal}, bt, \text{LivingThing} \rangle$. The well-displayed graph should be like Fig. 1(a), but in practice, it is hardly possible to obtain this result directly. Querying the information of the given node within one hop does not provide enough triples for constructing an informative structure of a graph, because the original triples are rarely found as shown in Fig. 1(b). In contrast, querying within two hops can maintain the mostly complete structure of the raw data as shown in Fig. 1(c), so it has an opportunity to be transformed into a simple graph by removing some inferred triples out of the query graph. The following expression can be used as a guideline to query the whole neighborhood of a given node (*uri*) within two hops.

```
CONSTRUCT { ?s ?p ?o.    ?o ?p1 ?o1. }
WHERE      { ?s ?p ?o.    ?o ?p1 ?o1.    FILTER(?s = <uri>) } .
```

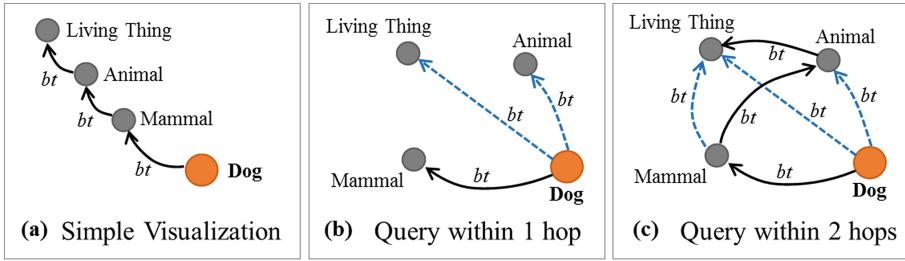


Fig. 1. Query result of the given term “Dog”. (Note: “*bt*” denotes *skos:broaderTransitive*, a black solid line indicates an original triple, a blue dashed line specifies an inferred triple, and a big yellow node represents the given node.)

Due to the nature of linked data and reasoning output, the according SPARQL statement usually forms some giant components in a graph. A hairball-like graph, as shown in Fig. 2, gives a bad impact to the readability of users and makes them be unsatisfied the way of learning and teaching using linked data. As we analyzed the query data from DBpedia [23] and LODAC [24] databases, we found two major issues.

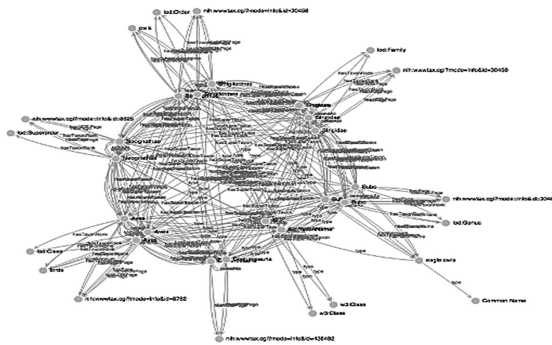


Fig. 2. Original RDF graph visualization from whole query result

Data Redundancy. A closer look at the data indicates that most inferred triples make a graph be highly complex. More than half of query triples are mainly formed by the reasoning results of *owl:sameAs*, *rdf:type* together with *rdfs:subClassOf*, and transitive properties. This behavior increases the average degree of the network and leads to have giant components, which have users be inconvenient to read.

Low Readability. In general, most well-organized articles such as academic papers prepare background knowledge of some essential concepts before bringing readers to the main content. Thanks to a well-outlined paper, beginners can understand it by reading from the beginning to the end, while experts of its domain can skip the introduction part and go to the main content directly. However, it is hardly possible to do with RDF data, because triples have no ordering. Thus, to give a ranking score to each triple is necessary.

In this case, we observed the distribution of URIs. It is found that the frequency of each URI in a query result (fQ) are distributed as shown in Fig. 3(a), where the horizontal axis shows individual URIs and the vertical axis shows the frequency of them. Several URIs have high degree. As we valued, most of them are important to display in a graph as key concepts. For example, if we query a term *dbpedia:Tokyo*, the high-degree URIs are *dbpedia:Tokyo*, *dbpedia:Japan*, *dbpedia:Honchu* (The island where Tokyo located), *rdf:type*, *owl:sameAs*, *dc:subject*, etc. The first three URIs are interesting because they are key concepts of “Tokyo”, whereas the last three URIs are not much important for domain experts. Thus, we learned that using the frequency of each term in a query result alone is not enough. Next, we analyzed the frequency of every URI in a dataset (fD), and compared each to the fQ chart one-by-one as shown in Fig. 3(b). This chart was drawn on a logarithmic scale because its distribution is extremely high variance. As fD of every URI found in the query result are estimated, a lot of high frequent ones are common properties such as *rdf:type*, *owl:sameAs*, *dc:subject*, etc. while *dbpedia:Tokyo*, *dbpedia:Japan*, and *dbpedia:Honchu* are not much high.

This characteristic of the data is meaningful. As query results are carefully analyzed, we found that URIs having high fQ can be treated as key concepts in the graph, while URIs having high fD indicate common information of the key concepts. This fundamental analysis will be utilized for ranking triples in the next section.

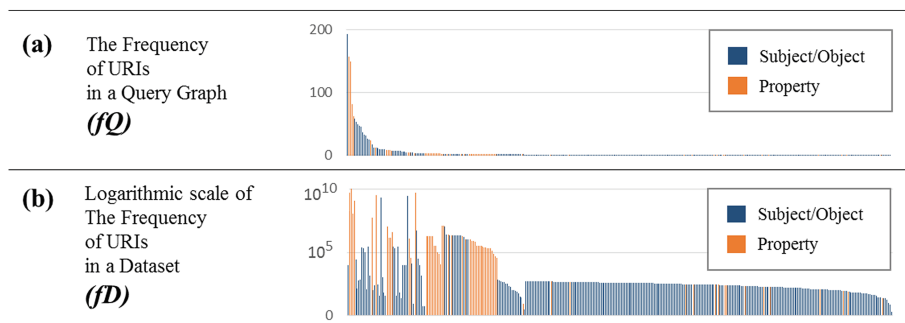


Fig. 3. Statistical analysis of URIs in the query result from DBpedia.

4 Proposed Approach

As we discussed, to motivate users to consume and contribute RDF data, they have to familiar with the knowledge representation of linked data. In this case, graph visualization is a suitable way to reduce a gap between human and Semantic Web. Understanding knowledge from a graph is quite challenging, because a graph is just a mathematical graph containing nodes and edges. In order to deliver graph-based knowledge to readers, an application should interpret all nodes and links as knowledge structures and make decision to maintain or eliminate some triples. To achieve this goal, we have to address the issues that are mostly discussed in the previous section. Thus, this research is initiated to serve the following purposes.

- To simplify a complex graph by removing redundant triples which are resulted from ontological reasoning.
- To serve different subgraphs on the basis of reading levels from common to topic-specific information.
- To filter a graph based on user preference.

4.1 Graph Simplification

As we mentioned, some well-prepared RDF repositories did reasoning on ontologies in order to support a SPARQL service, however, the inferred triples resulted in having giant components in a graph. As we investigate, equivalent or same-as instances (*owl:sameAs*), transitive properties (e.g. *skos:broaderTransitive*), and hierarchical classification (*rdf:type* together with *rdfs:subClassOf*) are commonly found in any complex RDF graph. Thus, this method aims to remove some redundant triples automatically by using rules that are defined in Table 1 and some description as follows:

- *R.1* – *R.3*: To merge some same-as nodes into one and remain only unique links.
- *R.4*: To remove implicit links that resulted by the chain of transitive links.
- *R.5*: To remove inferred links that caused by hierarchical classification.

Several rules use the occurrence number of a URI counted across data repositories in order to choose the most popular node from a same-as pair, because it has high opportunity to discover more knowledge in the next query.

4.2 Triple Ranking

The section of data analysis mentioned that the arrangement of any content is necessary for readers by preparing background knowledge at first in order to understand the main content well. As we reviewed, existing works focused on seeking relevant data according to a query expression, but they less mentioned about how to order them according to readability. Thus, this research introduces a simple method to sort triples on the basis of different levels of knowledge.

Table 1. A set of rules used to simplify a highly connected graph resulted by inferred triples. (Note: The term $fD(uri)$ is a frequency of a URI occurred in datasets.)

Rule	Triples	Condition	Display only
To merge nodes			
R.1		$fD(s_1) > fD(s_2)$	
		$fD(o_1) > fD(o_2)$	
R.2		$fD(s_1) > fD(s_2)$	
		$fD(o_1) > fD(o_2)$	
R.3		$fD(o_1) > fD(s_2)$	
To remove links			
R.4		$:p1 \text{ rdf:type owl:TransitiveProperty .}$	
R.5			

A general article contains different roles of concepts. **General Concepts** are terms that are commonly known such as “name”, “address”, and “class”, and they are always found in a corpus. Besides, **Key Concepts** are important terms that are always found in the given article but rarely found in a dataset. The key concepts are more relevance to the given article rather than the general ones. In terms of RDF, concepts are resources (including subjects and objects) and properties. The key concepts always present thorough the article, while general concepts are used as composition information for giving background knowledge of the key concepts as shown in Fig. 4(a).

In addition, different levels of information are defined. **Common Information** explains background knowledge that supports readers to understand the main content. It generally gives introduction of key concepts by using general terms. It means that triples being common information consist of general concepts rather than key concepts as shown in Fig. 4(b). In contrast, **Topic-Specific Information** contains specific terms that are highly relevance to the article. Thus, some triples acting as topic-specific information comprise of key concepts rather than general concepts as shown in Fig. 4(c).

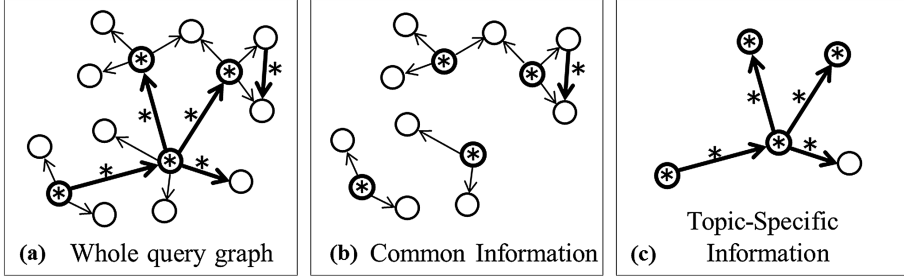


Fig. 4. Graphs display the idea of common information and topic-specific information. Nodes and links with stars (*) indicate key concepts, whereas the others are general concepts.

The level of each concept is valued based on a query result. As we analyzed, the key concepts are commonly found in the query result but they are rarely found in the dataset, while the general concepts are frequently appeared in the dataset. This manner is consistent with the TF-IDF method, however an RDF dataset contains only separated triples but not documents of many words, so this method has to be adapted for RDF data.

In this research, we intend to define that a key concept has higher score than a general concept, so the scoring function of a URI ($w(uri)$) is the occurrence number of a URI in a query result ($fQ(uri)$) weighted by the its occurrence number found in datasets ($fD(uri)$). Since the data analysis informed that the variance of $fD(uri)$ is extremely high, logarithm is taken for this term. The function w is defined by the equation:

$$w(uri) = \frac{fQ(uri)}{\log(fD(uri) + 1)}$$

Next, a function named Visualization-Weight (vw) is defined to measure a triple ($\langle s, p, o \rangle$) that should be in the direction of common or topic-specific information. It is the summary of weighting scores of subject (s), predicate (p), and object (o) of a triple as presented by the following equation.

$$vw(s, p, o) = \frac{\alpha \cdot w(s) + \beta \cdot w(p) + \gamma \cdot w(o)}{\alpha + \beta + \gamma}$$

The coefficients (α , β , and γ) of these terms are 1.0 by default; however, they can be adjusted if some domains place important to each term differently.

For example, the scores, $vw(\langle :Aves, :hasTaxonName, :Birds \rangle) = 0.33$ and $vw(\langle :Aves, :hasParentTaxon, :Coelurosauria \rangle) = 0.59$, show that the former is in the direction of common information, while the latter is more likely to be topic-specific information.

4.3 Property Selection

Moreover, although the problems discussed in the previous part can be solved, there are much more triples remained in the visualization. These data contain both necessary and unnecessary triples for readers. Since users have their own expectation to view a graph, they should customize the graph based on their interest by themselves. They always prefer to filter a graph by selecting only properties that they are interested.

This additional method named “Property Selection” is lastly described in this paper. The method helps users to focus on information that they desire to view. It is a simple technique that is always found in any visualization tool. In addition, we learn that most triples related to RDF, RDFS, and OWL are not needed by readers, for example, $\langle foaf:Person, rdf:type, rdfs:Class \rangle$, $\langle foaf:Person, rdfs:subClassOf, foaf:Agent \rangle$, $\langle foaf:Person, owl:disjointWith, foaf:Organization \rangle$, etc. In this case, filter out some of these properties and resources consume much user effort. Thus, the triples containing some vocabularies from RDF, RDFS, and OWL can be removed from a graph by considering the namespaces of subjects, predicates, and objects.

5 Prototype

The proposed approach originates an idea to organize RDF data for graph visualization. In order to verify the suitability and the feasibility of the proposed methods, a prototype has been developed.

5.1 User Requirement

Apart from the data analysis, we have gathered requirements from different users who have different levels of experience with Semantic Web and domain knowledge. In this part, the requirements from users are summarized into the following topics.

General Requirements.

- An application should provide different input interfaces for different types of users. A simple interface allows users to enter a URI, and then a graph is automatically queried. Besides, tech users are allowed to input a SPARQL expression for the advanced query.
- It is known that URIs are fundamental components in Semantic Web, and they are used as identifiers for machine-readable data on the web. However, most of them are difficult to be read by lay users. Thus, in a visualization, it should display human-readable labels in a graph for general users by default, and also provide an option to display URIs for tech users.

- Users are possible to move any node in the graph.
- In an RDF statement, its subject and property are URI, and object can be either URI or literals. Pairs of a datatype property and a literal node are used to be metadata of only one resource, and literal nodes may be long string, so they are not suitable to display in the limit area of the node-link diagram. Since these data are useful for readers, they should be displayed in another panel that users can access them easily.

Graph Simplification.

- Users can simplify a graph by merging same-as nodes and removing transitive links.

Triple Ranking.

- Since users have different background knowledge in a specific topic, beginners may interested in reading common information before getting topic-specific information, while experts may prefer to read only topic-specific information. Thus, the application should dynamically alter a graph according to the level of knowledge that users can customize and access on demand.

Property Selection.

- Users can select only properties that they prefer to view.
- Some triples containing vocabularies from RDF, RDFS, and OWL can be ignored.

5.2 Implementation

According to the user requirements, the prototype is implemented on the basis of the following features.

- Graph Simplification: To simplify a graph by removing redundant triples.
- Triple Ranking: To give ranking scores to triples based on common and topic-specific information.
- Property Selection: To filter a graph by selecting preferred properties.
- User Interaction: To control a graph according to user demand.

For this prototype, the functional diagram that described user actions and system workflows is shown in Fig. 5, the user interface is demonstrated in Fig. 6, and example graphs that are resulted from user actions are displayed in Fig. 7. The prototype is accessible at the following URL.

<http://rc.lodac.nii.ac.jp/rdf4u/>

This prototype is a web application that is mainly developed using the force layout of the D3 JavaScript library¹. The main user scenarios are described in the following topics together with each step in Figs. 5, 6 and 7.

¹ Data-Driven Documents (D3) <http://d3js.org/>.

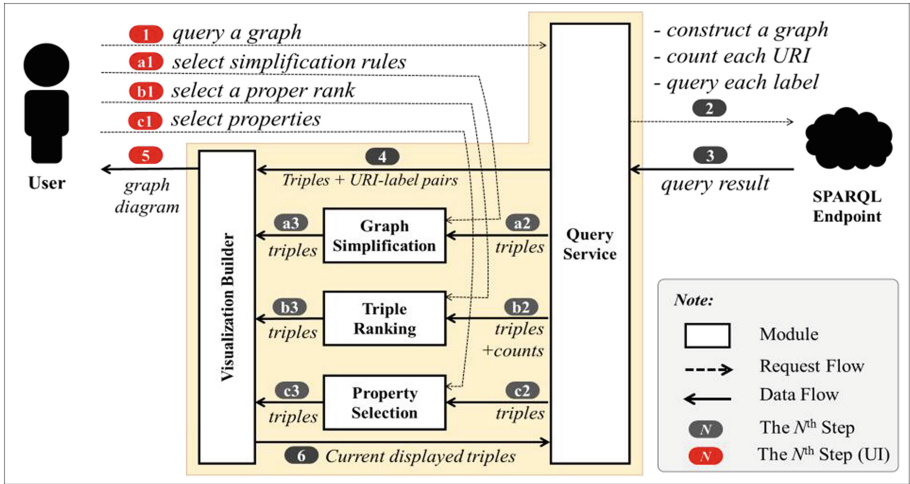


Fig. 5. Functional diagram

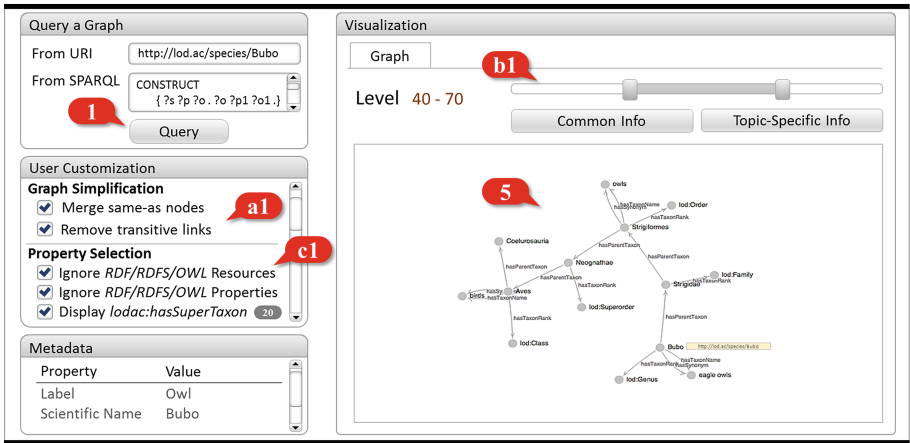


Fig. 6. User interface

General Requirements. First, the main flow visualization is the query of graph. Users request a graph (Step 1 in Fig. 5) by giving either a single URI or a SPARQL CONSTRUCT statement (Step 1 in Fig. 6). After that, the module “Query Service” forwards the query statements for getting a graph, counting the number of each URI, and inquiring the label of each URI to any SPARQL endpoint (Step 2 in Fig. 5). When the result is returned to the Query Service (Step 3 in Fig. 5), it forwards to the module “Visualization Builder” (Step 4 in Fig. 5). Then, the Visualization Builder generates graph visualization to users (Step 5 in Figs. 6 and 7). Since inferred triples are also

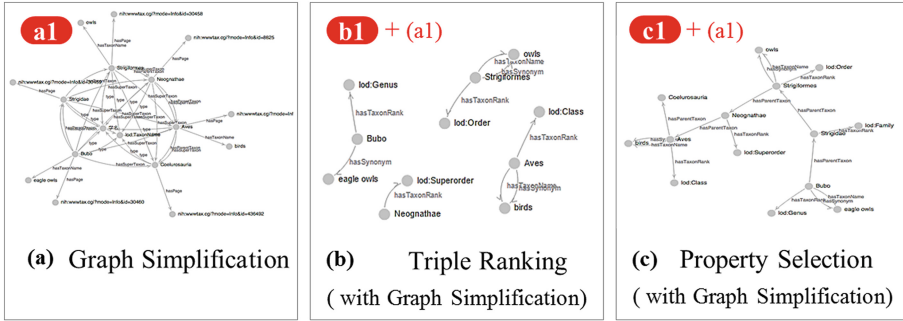


Fig. 7. Output from the prototype

retrieved, the original graph is highly complicated as shown in Fig. 2. In addition, each node in the graph is moveable, all of labels are human readable, a URI is shown when a user move a pointer over a node or a link. When a node is double-clicked, the literal information of a node is shown (In panel “Metadata” in Fig. 6). Moreover, every displayed triple is synchronized to the Query Service to be input data for other modules in next user actions (Step 6 in Fig. 5).

Graph Simplification. Second, users are allowed to select simplification rules (Step a1 in Figs. 5 and 6). When users click on any options, the module “Graph Simplification” executes some related rules and forwards result triples to the Visualization Builder (Step a2 and a3 in Fig. 5). As a result, the graph visualization in Fig. 7(a) shows that the simplified graph is easier to read than the original one. In the experiment, some redundant triples that are about 50–70 % of the original query graph are removed during this process.

Triple Ranking. Next, users can select the range of visualization ranking (Step b1 in Fig. 5) by moving a two-way slider bar or clicking either the button “Common Information” or “Topic-Specific Information” (Step b1 in Fig. 6). The former button displays triples having the lower *vw* score, while the latter one displays triples having the higher *vw* score. In the visualization tier, the integer number indicated the percentile of *vw* score is used as visualization level, because it is easier to be recognized by users rather than using the floating number of the *vw* value. Then, the module “Triple Ranking” computes and returns the triples that satisfy user input (Step b2 and b3 in Fig. 5). The result of this action together with the graph simplification is shown in Fig. 7(b). It displays common information that contains some key concepts and some general concepts in order to give background knowledge of the key concepts.

Property Selection. Last, users can customize a graph by selecting only preferred properties (Step c1 in Fig. 5). The interface allows users to hide resources and predicates that are vocabularies of RDF, RDFS, and OWL; and to show triples containing selected properties (Step c1 in Fig. 6). Then, the module “Property Selection” filters the triples according to the user input, and forwards the result to the Visualization Builder (Step c2 and c3 in Fig. 5). An example result of this scenario together with the graph simplification is shown in Fig. 7(c).

In summary of this section, the prototype demonstrates that our approach is possible and suitable to implement. The features that we provide satisfy all requirements that we have previously discussed.

6 Discussion

This research aims to provide a suitable RDF graph visualization that users are easily to consume knowledge by learning from relationship among concepts. Thus, the three main methods: graph simplification, triple ranking, and property selection, are proposed to deliver an easily-readable graph to readers. The first and the second methods are major contribution, while the last one is an additional method used for fulfilling some minor requirements. In this paper, we intend to introduce the according methods rather than a new fully-functioned visualization tool. Thus, this section points to the discussion about the usefulness, uniqueness, novelty, and prospect of this research.

6.1 Usefulness

Since a graph generated by RDF data is complicated by nature, users are not convention to read and understand knowledge from a graph. The analysis of mathematical features of a graph alone is not enough to simplifying the complexity of an RDF graph, because the RDF graph has semantic relationships that should be interpreted as knowledge structures. We carefully examined the actual behavior of RDF datasets, and found that the semantic structure of the datasets is meaningful in terms of knowledge representation, and it is useful for our research. The observation includes data redundancy such as same-as nodes and inferred relationships. When same-as nodes are merged and some inferred triples are filtered out by the simplification rules, some giant components in a network are eliminated, so the interactive graph on two-dimensional canvas becomes more sparse and convenient for users to control and read.

In addition, the degree of importance of triples such as distinction between common and topic-specific information was also investigated. For this reason, we have to realize the importance of triples depended on the expertise level of users. For domain experts, only topic-specific information is needed to show, while common information should be more emphasized for beginners. A case of multiple links between two nodes caused by the hierarchy of property demonstrates how this method is suitable for arranging data for readers. In general, a super property in an upper ontology is labeled by a common vocabulary describing the broader meaning, while a sub property is used by a specific domain. After reasoning, the number of a super property is certainly greater than the number of a sub property, so the super property trends to be displayed at the common level while the sup property often appears at the topic-specific level.

6.2 Uniqueness

The uniqueness of this research is discussed by functional comparison. The functionality of some visualization tools: Motif [11], Gephi [12], RDF Gravity [13], Fenfire [14], and IsaViz [15]; are studied according to the key methods of this research.

Graph Simplification. There are several works support this feature but the strategies are different. Motif replaces a dense component by an abstract shape, so a graph seems simple, but its detail is omitted. Gephi uses mathematical characteristics of a graph such as a node degree and a weight on edge, but it does not employ the knowledge structure of Semantic Web to reduce some redundant links. FenFire fades away some far nodes, but the subgraph including the focused node and its neighbors can produce giant components. Next, RDF Gravity and IsaViz can simplify a graph by having users to query inside the graph or select some URIs to be visible or hidden. However, they less discuss about options to merge same-as nodes and remove transitive links, which are the main issues of having dense parts in a graph. Unlike these existing tools, our approach adopts Semantic Web rules to interpret data and eliminate this issue automatically.

Triple Ranking. The according visualization tools do not mention about a way to arrange contents in a graph for serving different levels of knowledge to different users. A workaround is to filter some resources or properties based on user interest, but users have to put their effort to learn what they want to view and how to filter. Thus, in this case, our work provides a smart way to solve this issue by analyzing the statistical feature of data and then automatically ordering a graph from common to topic-specific information.

Property Selection. Filtering a graph by selecting preferred properties is a common feature that most visualization tools provide. Our work was implemented in the same way. In addition, we added an option to show or hide triples containing some vocabularies from RDF, RDFS, and OWL automatically, so users do not have to remove them one by one.

In summary, considering these three features, our solution has advantage over the existing visualization tools because our approach does not only allow users to customize a graph but also automatically deliver an easy-readable graph based on the knowledge interpretation and the statistical analysis of Semantic Web data.

6.3 Novelty

Due to the contradictory requirements from different types of users, we adapted TF-IDF method for ordering triple from common to topic-specific levels. The degree of commonness versus specificity is calculated by evaluating the nature of the dataset with the algorithm. After that, the RDF visualization application is designed to allow users to choose how common or domain-specific information that they need by clicking a button or controlling a two-way slider bar. The prototype was demonstrated and it got positive impression from users. Moreover, it can be considered that this work is a novel

approach because it operates a graph at the knowledge level by concerning domain independent, so this approach is applicable to any domain datasets.

6.4 Prospect

Since the arrangement of triples for reading is a novel approach, it has opportunity to be value-added by the community of Semantic Web researchers. This approach can be extended by applying various algorithms in order to satisfy diverse characteristics of data in other domains. We are going to apply this system as a learning and teaching tool for a specific domain such as biodiversity informatics [25, 26], because a graph diagram can enhance the learning of biology [10], and it can be clearly to identify the level of users from beginners (e.g. high school students) to experts (i.e. researchers).

Moreover, as our observation, although this RDF graph visualization application does not give technical knowledge of RDF to lay users directly, it makes them appreciate and understand the role of linked data for future knowledge management. This one important task that attempts to break a barrier between human and Semantic Web.

7 Conclusion

This paper aims to deliver a suitable RDF graph visualization for every level of users, because a node-link diagram can enhance the learning ability of users and the amount of LOD is positively growing. Since the nature of RDF data makes a graph be complicated, it is difficult for users to read and understand. As we analyze, the root causes are data redundancy due to inferred triples, and low readability due to lacking of reading flow in a RDF graph. This research initiates three main methods to support readers. First, Graph Simplification executes the proposed Semantic Web rules to remove some inferred data. Second, Triple Ranking prepares different sections of a graph from common to topic-specific information for different levels of users by adapting TF-IDF algorithm for an RDF graph. Last, Property Selection is additionally developed to allow users to display or hide triples by selecting some properties, and to help users to filter some triples containing some vocabularies from RDF, RDFS, and OWL. These methods mostly use the statistical feature of a RDF graph together with the interpretation of RDF data as knowledge structures in order to produce an easily-readable node-link diagram for readers. The prototype is implemented by including interactive RDF visualization in order to verify the suitability and feasibility of our approach. It demonstrates that our methods can be developed on the basis of today's technologies and the prototype enables users to realize the power of Semantic Web and LOD for enhancing the ability of knowledge management.

In future, we plan to measure the expertise level of users, and allow system to adjust the visualization by applying various algorithms for different domain-specific datasets in order to deliver a more appropriate RDF graph for all levels of users.

References

1. Heath, T., Christian, B.: Linked data: evolving the web into a global data space. *Synth. Lect. Semant. Web: Theory Technol.* **1**(1), 1–136 (2011)
2. Suchanek, F., Weikum, G.: Knowledge harvesting in the big-data era. In: *Proceedings of the 2013 ACM SIGMOD*, pp. 933–938. ACM (2013)
3. Bizer, C., Heath, T., Berners-Lee, T.: Linked data-the story so far. In: *Semantic Services, Interoperability and Web Applications: Emerging Concepts*, pp. 205–227 (2009)
4. Hitzler, P., Krotzsch, M., Rudolph, S.: *Foundations of Semantic Web Technologies*. CRC Press, Boca Raton (2009)
5. Dadzie, A.-S., Rowe, M.: Approaches to visualising linked data: a survey. *Semant. Web J.* **2**(2), 89–124 (2011)
6. Bezerra, C., Freitas, F., Santana, F.: Evaluating ontologies with competency questions. In: *Web Intelligence (WI) and Intelligent Agent Technologies (IAT)*, vol. 3 (2013)
7. Zemmouchi-Ghomari, L., Ghomari, A.: Translating natural language competency questions into SPARQLQueries: a case study. In: *The First International Conference on Building and Exploring Web Based Environments*, pp. 81–86 (2013)
8. Schwendimann, B.: Concept maps as versatile tools to integrate complex ideas: from kindergarten to higher and professional education. *Knowl. Manage. E-Learn.* **7**(1), 73–99 (2015)
9. Edelson, D., Gordin, D.: Visualization for learners: a framework for adapting scientists’ tools. *Comput. Geosci.* **24**(7), 607–616 (1998)
10. Liu, S., Lee, G.: Using a concept map knowledge management system to enhance the learning of biology. *Comput. Educ.* **68**, 105–116 (2013)
11. Dunne, C., Shneiderman, B.: Motif simplification: improving network visualization readability with fan, connector, and clique glyphs. In: *SIGCHI*, pp. 3247–3256 (2013)
12. Mathieu, B., Sebastien, H., Mathieu, J.: Gephi: an open source software for exploring and manipulating networks. In: *AAAI 2009* (2009)
13. Goyal, S., Westenthaler, R.: *RDF Gravity (Rdf Graph Visualization Tool)*. Salzburg Research, Austria (2004)
14. Tuukka, H., Cyganiak, R., Bojars, U.: Browsing linked data with Fenfire. In: *LDOW 2008 at WWW 2008* (2008)
15. Pretorius, J., Jarke, J., Van, W.: What does the user want to see? What do the data want to be?. *Inf. Vis.* **8**(3), 153–166 (2009)
16. Lee, S., Kim, H.J.: News keyword extraction for topic tracking. In: *NCM 2008* (2008)
17. Li, J., Zhang, K.: Keyword extraction based on tf/idf for Chinese news document. *Wuhan Univ. J. Nat. Sci.* **12**(5), 917–921 (2007)
18. Brin, S., Page, L.: The anatomy of a large-scale hypertextual web search engine. In: *WWW 1998* (1998)
19. Franz, T., Schultz, A., Sizov, S., Staab, S.: TripleRank: ranking semantic web data by tensor decomposition. In: Bernstein, A., Karger, D.R., Heath, T., Feigenbaum, L., Maynard, D., Motta, E., Thirunarayan, K. (eds.) *ISWC 2009. LNCS*, vol. 5823, pp. 213–228. Springer, Heidelberg (2009)
20. Kleinberg, J.: Authoritative sources in a hyperlinked environment. *J. ACM (JACM)* **46**(5), 604–632 (1999)
21. Ichinose, S., Kobayashi, I., Iwazume, M., Tanaka, K.: Ranking the results of DBpedia retrieval with SPARQL query. In: Kim, W., Ding, Y., Kim, H.-G. (eds.) *JIST 2013. LNCS*, vol. 8388, pp. 306–319. Springer, Heidelberg (2014)

22. Novak, J., Cañas, A.: The theory underlying concept maps and how to construct and use them. In: Florida Institute for Human and Machine Cognition (2006)
23. Lehmann, J., et al.: DBpedia – a large-scale, multilingual knowledge base extracted from Wikipedia. *Seman. Web J.* **6**(2), 167–195 (2015)
24. Minami, Y., Takeda, H., Kato, F., Ohmukai, I., Arai, N., Jinbo, U., Ito, M., Kobayashi, S., Kawamoto, S.: Towards a data hub for biodiversity with LOD. In: Takeda, H., Qu, Y., Mizoguchi, R., Kitamura, Y. (eds.) *JIST 2012. LNCS*, vol. 7774, pp. 356–361. Springer, Heidelberg (2013)
25. Chawuthai, R., Takeda, H., Hosoya, T.: Link prediction in linked data of interspecies interactions using hybrid recommendation approach. In: Supnithi, T., Yamaguchi, T., Pan, J.Z., Wuwongse, V., Buranarach, M. (eds.) *JIST 2014. LNCS*, vol. 8943, pp. 113–128. Springer, Heidelberg (2015)
26. Chawuthai, R., et al.: A logical model for taxonomic concepts for expanding knowledge using Linked Open Data. In: *Workshop on Semantics for Biodiversity* (2013)

Semantic Technology

5th Joint International Conference, JIST 2015, Yichang,
China, November 11-13, 2015, Revised Selected Papers

Qi, G.; Kozaki, K.; Pan, J.Z.; Yu, S. (Eds.)

2016, XIII, 332 p. 84 illus., Softcover

ISBN: 978-3-319-31675-8