

LOD Explorer: Presenting the Web of Data

Karwan Jacksi

Computer Science Department
University of Zakho
Zakho, Iraq

Subhi R. M. Zeebaree

Computer and Communications
Engineering Department
Nawroz University
Duhok, Iraq

Nazife Dimililer

Information Technology Department
Eastern Mediterranean University
Gazimagusa, N. Cyprus

Abstract—The quantity of data published on the Web according to principles of Linked Data is increasing intensely. However, this data is still largely limited to be used up by domain professionals and users who understand Linked Data technologies. Therefore, it is essential to develop tools to enhance intuitive perceptions of Linked Data for lay users. The features of Linked Data point to various challenges for an easy-to-use data presentation. In this paper, Semantic Web and Linked Data technologies are overviewed, challenges to the presentation of Linked Data is stated, and LOD Explorer is presented with the aim of delivering a simple application to discover triplestore resources. Furthermore, to hide the technical challenges behind Linked Data and provide both specialist and non-specialist users, an interactive and effective way to explore RDF resources.

Keywords—Semantic web; linked open data; linked data browsers; exploratory search systems; RDF; SPARQL

I. INTRODUCTION

Day after day, the amount of uploaded data to the Web grows, due to the simple uploading process offered by World Wide Web (www) [1]. Thus, the Web has transformed into a giant semi-structured collection of data, which makes information retrieval a challenging task. Search engines are typically used for information retrieval from the Web, but finding highly relevant retrievals, efficient search skills are necessary.

Marchionini has categorized the search approaches into two groups: lookup and exploratory search [2]. In the lookup search approach, also called keyword-based search, database systems are used to find information using keywords. This is the widely used approach in the existing Web, aka Syntactic Web, where the data sources are mainly text formats and the search elements are known [3].

Exploratory search is a special information seeking method, where the goal of users is not essentially identified through the search process [4]. In this approach, learning and investigation are more important for a user than retrievals of facts and replies to queries. The user compares, investigates and learns new ideas and concepts for the retrieved information [5], [6].

The information retrievals in the Syntactic Web is limited to keywords. Thus, search engines use the user query keywords to retrieve information, where the quality of retrieved results is rather poor. To develop the issue, the contents of the Syntactic Web are enriched with annotations forming the Semantic Web [1], [7].

The Semantic Web is an extension and next generation of the WWW through standards by the W3C¹. The data of the Semantic Web has well-defined meanings, can be understood and processed by machines, and allows machines and people to work in collaboration [8]. The Semantic Web combines the technologies of RDF², OWL³, and XML⁴ to enable the replacement of the Syntactic Web so as to provide search engines capability to understand the meaning of data [7].

The Semantic Web cannot be completed by only annotating the data on the Web, but the data has to be linked with each other so as the Web of data can be formed and be discovered by machines and people [9]. Linked data makes it possible to discover related data of a term once only a subset is given. Hence, the terms Semantic Web and Linked Data have been coined by Berners-Lee and defined the Linked Data as “Semantic Web done right” [10].

The Linked Data (LD) term points out to a set of steps to distribute and connect structured data on the Web. These steps were introduced by Berners-Lee in his impressions about Web architecture design issues and soon turned out to be the principles of LD [11].

In the hypertext Web, HTML documents are connected with each other using untyped hyperlinks, whereas LD depends on the documents having RDF formats to create typed links that connect things globally forming the Web of Data [12], [13]. Once the LD is presented under an open license, it's called Linked Open Data (LOD).

The rest of the research is organized as follows: DBpedia dataset is described in Section 2. In Section 3, related works are addressed, and LOD Explorer is presented in Section 4. Evaluation of the application elaborated in Section 5, and results of the evaluation is detailed in Section 6. Conclusions and future work are given in Section 7.

II. DBpedia DATASET

DBpedia is a leading project for publishing LD started by individuals at the Free University of Berlin and Leipzig University in cooperation with OpenLink Software. The project was first published in public as a Linked Open Data dataset in 2007 with the intention of becoming a large,

¹ World Wide Web Consortium: www.w3.org

² Resource Description Framework: www.w3.org/RDF

³ Web Ontology Language: www.w3.org/OWL

⁴ Extensible Markup Language: www.w3.org/XML

multilingual, semantic knowledge graph for an open data infrastructure. It is now the center of Linked Open Data cloud⁵. The data of the dataset is created from the extracted information of the Wikipedia using the DBpedia Information Extraction Framework. The latest release (2016-10) of DBpedia consists of 13 billion pieces of information (RDF triples) where 1.7 billion pieces were English edition extractions of Wikipedia, 6.6 billion from other language editions and 4.8 from Wikipedia Commons and Wikidata. The English edition of the DBpedia dataset defines 6.6 million entities out of which 4.9 million have abstracts and 1.7 million have depictions. Altogether, 5.5 million resources are classified in a reliable ontology, containing 1.5 million persons, 840 thousand places, 496 thousand works such as films and music albums, 286 thousand organizations, 306 thousand species, 58 thousand plants and 6 thousand diseases. In addition to 6.6 million entities, the overall count of DBpedia for the English version is 18 million resources which include 1.7 million of SKOS concepts (categories), 7.7 million redirect pages, 269 thousand disambiguation pages and 1.7 million intermediate nodes [14].

Entities of DBpedia have different varieties of information, they normally have types, links, categories, labels, links of LD, and literal descriptions related to them. Within the DBpedia dataset, there are relations to identical entities for other languages (for instance ar.dbpedia.org), and there are associations to corresponding entities reside in other datasets as in case of YAGO dataset. Additionally, there are specific domain classes and properties such as the Person typed entity dbpedia:Carl_XVI_Gustaf_of_Sweden has dbo:spouse which donates to the entity dbpedia:Queen_Silvia_of_Sweden.

From the time when DBpedia dataset was publicly published, various services and tools have been developed around it. DBpedia Spotlight⁶, which is a tool for robotically annotating mentions of DBpedia resources in text [15]. DBpedia Lookup⁷, a web service which allows to look up for DBpedia entities by related keywords. The DBpedia mappings wiki⁸, an exertion to improve the DBpedia information by obtaining mappings between the dataset ontology and Wikipedia Infoboxes. The DBpedia Extraction Framework⁹ uses the mappings to standardize information extracted from Wikipedia before creating structured information in RDF. Besides DBpedia tools, further independent tools and services have been developed which use DBpedia as their dataset. In the following section, a few of such tools and services are employed.

III. RELATED WORKS

In recent times, Linked Data (LD) usage on the Web has remarkably enlarged. However, for the lay-users, it is still challenging to be used. Dealing with LD to be used and visualized has been known as problems from the time when the foundation of the Semantic Web [16]. The growing

development of LD applications resulted in providing a set of approaches to let users interact and grasp the LD notion. Some of approaches present LD as outline and table modes as in Tabulator¹⁰ and Explorator¹¹, others present LD as graphs as in Graphity¹² and RelFinder¹³, whereas a combination of both features can be found in other systems such as LODmilla¹⁴. Authors of [17] present a rich and state of the art survey of LD exploration systems.

SWOC¹⁵ uses semantic connections in the DBpedia dataset to let humans explore its resources [3]. Besides of using the semantic properties of DBpedia, the system uses Web search engines and social tagging systems as external resources making a hybrid approach to present DBpedia nodes. The system made up of two main modules: back-end, where the calculations of pairs between DBpedia resources are performed to produce similarities for the initial node, and a flash-based front-end presenting the results of the back-end.

At the front-end, DBpedia lookup service¹⁶ is utilized to select an initial. The selected node, which should be of the ICT area, is presented on the webpage surrounding with most ten similar resources computed in the back-end. At the right side, a windowpane is available to present basic information about the selected resource.

LED¹⁷ utilizes DBpedia dataset to provide users related resources to a query [18]. It uses DBpedia lookup service to return a resource of RDF dataset. Later, the system forms a cloud of tags that are semantically related to the selected resource. New tags from the formed cloud can be added to the main query resulting in a new query of the combined resources in a new tab. A pop-up pane for each resource is available while hovering on a tag presenting a description of the tag.

Aemoo¹⁸ uses Encyclopedic Knowledge Pattern (EKP)¹⁹ to explore the data of DBpedia [19]. When the system gets a query, it uses DBpedia first to process the query, then Wikipedia, Twitter, and Google News are used as external sources to assemble and combine the data from. The combination of data is achieved by principles of cognitively sound approaches by using knowledge patterns, the structure of hypertext links, and utilizing technologies of the semantic web. To present the retrieved data, EKP filters are used so that only related data is presented. A further utility called *curiosity* is offered by the system so that to show the filtered information by the EKP.

LodLive²⁰ explores RDF resources and visualizes them as dynamic graphs [20]. Resources in this system can be

¹⁰ w3.org/2005/ajar/tab

¹¹ tecweb.inf.puc-rio.br/explorator

¹² graphity.org

¹³ visualdataweb.org/relfinder.php

¹⁴ lodmilla.sztaki.hu

¹⁵ sisinlab.poliba.it/semantic-wonder-cloud

¹⁶ wiki.dbpedia.org/Lookup

¹⁷ sisinlab.poliba.it/led

¹⁸ wit.istc.cnr.it/aemoo

¹⁹ ontologydesignpatterns.org/ekp

²⁰ en.lodlive.it

⁵ lod-cloud.net/versions/2017-02-20/lod.svg

⁶ demo.dbpedia-spotlight.org/

⁷ github.com/dbpedia/lookup

⁸ mappings.dbpedia.org

⁹ github.com/dbpedia/extraction-framework

connected from different endpoints. By using the Sesame Framework, RDF data can be parsed even when they are not in a SPARQL endpoint. This can be achieved by remotely creating graphs in order to store the requested resources temporarily for making queries. The system can also be used as a tool for the ontology definitions in its early stages so as to check the validity of an RDF schema and select a solution among several ones visually. The application is built using JavaScript and presents the calls from endpoints in HTML5 web pages. The retrievals of JSON format of JSONP (JSON with Padding) calls from endpoints are parsed to HTML documents without the need of a server-side programming.

LODmilla²¹ is a LOD browser and editor that combines the features of both textual and graph-based LD browsers [21]. The system provides the abilities to connect to several LD datasets and browse the LD resources. Editing the resources is one of the main features of the application. The system consists of two main parts, a frontend side and a server side. The frontend is constructed using JavaScript while Java has been used for the server side. A dedicated server has been set for the system so as to enable search functions and support caching and fast loading of RDF triples. Two techniques can be used when loading RDF triples: a SPARQL-based query and actionable URIs. Using the Jena toolkit at the server side, several serializations can be obtained from parsing RDF data including JSON. Hence, multiple datasets can be used in parallel regardless of configuring the details of datasets at the frontend. The editing functions of the system give users abilities to add or remove resources or to make new connections between resources of a dataset.

LD Viewer is an adaptable framework of several tools to present a user-friendliness exploration of LD datasets [22]. The main target of the project is to provide a unified and powerful featured interface that can easily be accepted by several LD datasets. The retrieved information from the RDF datasets is presented in a tabular form of properties. Forward and reverse exploration of properties for each of the retrieved resource are offered, furthermore, a pagination feature for reverse properties of a large amount of values is available. Based on the nature of triples, each triple in the property table has action(s) which can be clicked. For instance, annotations to DBpedia dataset can be accomplished if the action is applicable for such triple. The application is implemented by JavaScript and largely by using AngularJS framework, and components of JASSA library²² (JAVascript Suite for Sparql Access). Configuring the application with an LD dataset does not need to understand the core of the application.

IV. LOD EXPLORER

Thus far, the size of the LD growing intensely, subsequently, a lot of LD projects are available to be used and millions of triples have been put away in triple datasets. But from the opposing point of view, it is challenging to find exploring tools truly based on RDF standards and capable to

validate the efficiency of these standards. LOD Explorer²³ has been developed with the aim of:

- RDF datasets exploration employing a dynamic visual graph
- using different RDF datasets to be used and connected with each other
- expanding the norm and standardization space of LD
- providing an easy application to be used by everybody for LD Exploration
- presenting data properties of LD resources
- searching within the resources to find it's connections
- fetch and display an image of the resource
- providing flexibility for adding plugins.

The fundamental idea of the LOD Explorer is to deliver an easy approach to discover, understand, and learn the published resources along with the W3C standards for Semantic Web.

The novelty of the proposed approach is the capability to straightforwardly explore a SPARQL endpoint utilizing the greatness of JavaScript and its libraries without a necessity of a server-side module.

LOD Explorer uses the technologies of JSONP calls to the constructed endpoints fetching JSON formatted data to be parsed by JavaScript and presents the LOD resources in an HTML5 web page. The resources are presented as graph nodes while their properties as textual information with the aim of mixing the best of both worlds. Hence, this way, the significance of using SPARQL endpoints can be proved and promote using triplestores to develop federated queries.

LOD Explorer processes RDF data in advance and organizes them to be presented. The system presents all existing materials in RDF datasets without hiding any of its portions. For instance, property types are used to group In/Out properties.

The exploration process can be started by querying the endpoint for a particular resource either by using a resource name or a resource URI. A couple of resource examples are provided as well where one can start from. Afterward, exploring the resource is easy as can be through an attractive information presentation and following the related incoming and outgoing connections. New resources can be added to the graph and each of the newly opened resources will automatically connect to the ones already opened if and only if there is a semantic connection between them.

The system is constructed using the following technologies:

- Pure JavaScript
- jQuery libraries
- jsPlumb toolkit²⁴ to draw nodes of graph
- an HTML5 page

²³ lodexplorer.uoz.edu.krd

²⁴ jsplumbtoolkit.com

²¹ lodmilla.sztaki.hu

²² aksw.org/Projects/Jassa.html

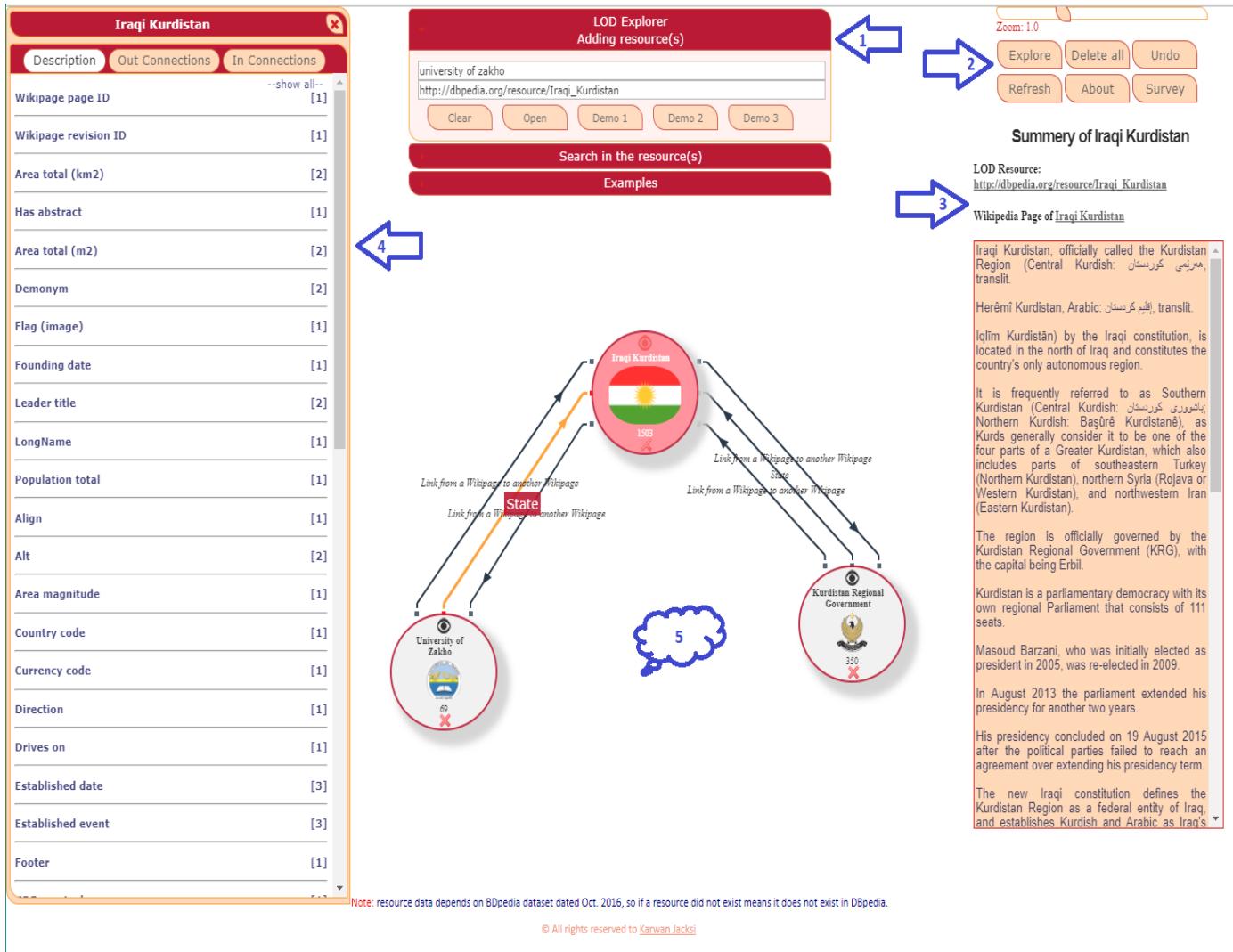


Fig. 1. System interface.

The user interface of the application consists of the following parts as in Fig. 1: System interface.

- 1) Search panel (center)
- 2) Toolbox (top right)
- 3) Description box (right)
- 4) Details Panel (left)
- 5) and the ground

The ground of the application is where the resources are presented in the form of graph nodes. The search panel is the main part of the system where resources can be found from LOD datasets and presented on the background. The resources are opened from this panel using either the resource name or the resource URI. When using resources names, an autocomplete search is offered by the system so as to select one of the offered resources. While when using URI of a resource, the available open button has to be clicked. Hence, the resources are opened this way and are drawn on the ground as graph nodes. Moreover, the nodes can shrink and enlarge by zooming them in and out, and they can be moved around anywhere on the mouse.

When a resource is opened, a search function from the search panel is activated so that to search inside the opened resource and find related information to the resource, as in Search in the resources. When multiple resources are open, the search function searches inside all of the opened resources. Results of search within resources are given in the form of active autocomplete combining suggestions of all of the related information to the opened resources right below its input box. The selected suggestion from the results opens the details panel.

The details panel contains all the details of the opened resource. This panel can be opened by either clicking on the eye button as in Resource as a graph node, or through the search within resource results. The panel consists of three main parts: 1) the description tab, 2) the out connections tab and 3) the in connections tab. The description tab contains detailed properties about the resource itself that are of the type literals. In and out connections are defined by the direction property and are presented in groups as labels having elements with targeted URIs. The panel is labeled with the resource label so

to realize the opened resource, and it can be closed to give more space to the background.

During this process, some presumptions are set to the nodes to enhance the visual appearance. For instance, a searching icon is set to let the user wait for the process to get completed. The node image is taken from the value of resource property dbpedia:thumbnail and foaf:depiction. And if that value is not available, the values of rdf:type property are used to show predefined icons such as no endpoint, person, group, work...etc. The values of foaf:name, rdfs:label, skos:prefLabel or dc:title properties are used for the node label.

Newly opened resources are inserted to the page without affecting the existing ones, this is helpful to let the surfer realize the new resource and to provide a least disruptive technique. After inserting new resources, the search within resources' array gets enriched with new information from the new resource. Any opened resource can be deleted as well as individual, this can be done using the cross sign (X) from the node. As a result, all LD related to the deleted node is removed from the search array.



Fig. 2. Search in the resources.

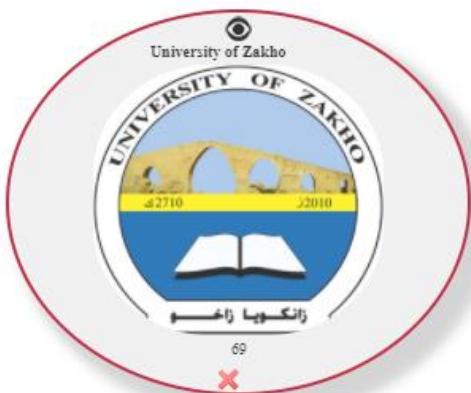


Fig. 3. Resource as a graph node.

The right-hand up buttons are working as follow: the Explore button is used to expand the exploration process by inserting a predefined number of connections from the configuration file (currently set to 5). So, when this action is clicked, the system inserts 5 new nodes related to the selected node and present them to the page having direct connections to the selected node. The aim of this method is to help users get a larger vision of LOD exploration and to give them a better idea of how the system works.

A Delete all button, as from its name, it deletes all the opened resources and removes them from the search in the resources array. An Undo utility has been employed as well so as to go back to the last actions the user made sequentially.

V. EVALUATION

To assess the proposed system, a user survey is conducted. The survey is based on System Usability Scale (SUS)²⁵, which is an effective tool for evaluating the usability of a product and signifies a self-reported survey metric. The SUS scores score can range from 0 to 100, the highest score the highest level of efficiency, productivity, and satisfaction to the application [23].

The users of the survey have to work on the system first prior starting the evaluation. Therefore, the system has been uploaded to an online host for that purpose. With this survey, realizing whether the users in general like the system and how intuitive they're experiencing it are the targets.

The survey consists of two main parts: the first part includes questions to build a simple user profile. Only questions about users' affiliation, academic rank and degree, and discipline are asked. The second part of the survey is the standard SUS questions, which consists of 10 questions with 5 response options to show an average user satisfaction or dissatisfaction. At the end of the questionnaire, a suggestion field is also added. The SUS questions are listed below:

- Q1. I think that I would like to use this system frequently.
- Q2. I found the system unnecessarily complex.
- Q3. I thought the system was easy to use.
- Q4. I think that I would need the support of a technical person to be able to use this system.
- Q5. I found the various functions in this system were well integrated.
- Q6. I thought there was too much inconsistency in this system.
- Q7. I would imagine that most people would learn to use this system very quickly.
- Q8. I found the system very cumbersome/awkward to use.
- Q9. I felt very confident using the system.
- Q10. I needed to learn a lot of things before I could get going with this system.

²⁵ en.wikipedia.org/wiki/System_usability_scale

The response format is: strongly agree (SA), agree (A), neutral (N), disagree (DA), and strongly disagree (SDA).

VI. EVALUATION RESULTS

The survey is sent to 80 individuals, out of which 62 were responded. Around 19% were Ph.D. degree holders, 8% were Ph.D. students, and 50% were Masters. The academic rank of the participants was as follows: 2% Profs, 11% Assist Profs, 10 Lecturers, 42% Assistant Lecturers and 36% with no academic title. Discipline was an important factor in the survey so as to know the feedback from the more specialized participants. 74% of the participants were from Computer Science specialists, and the rest were from Chemistry, Biology, History, Economics, Environmental Science, Law, Civil, Mechanical and Petroleum Engineering.

The initial survey shows participants overall like the application. Responses to Q1 were 39%SA, 42%A, and 16%N, which indicate the users like to use the system. Around 15% found the system is unnecessarily complex, while 84% (44%SA, 40%A) through the application is easy to use. 15% of the participants need assistance to use the application, and they're mostly from unspecialized people. 78% (23%SA, 55%A) went for Q5, and 8% thought there is inconsistency in the application. For the question: most people would learn to use this application very quickly the responses were: 26%SA, 47%A, and 19%N. Feedback for Q8 was 37%SDA, 39%DA, and 15%N. 87% felt very confident to use the application, while 13% needed to learn many things before using the application.

Most comments to the system were to compliment the efforts taken building this application while one of them was interesting since it was talking about the found resources are not up-to-date and this is of course not a fault of the system since it depends on the DBpedia dataset version 2016-10.

The suggestions part of the survey was an important plan to improve the system. Nine suggestions for the system have been recorded, some of which were well valued. Somebody suggested disabling the search within resources function when there are no resources on the ground to search within it; this has been implemented and added to the system. Someone else advised adding auto-correction feature to the search process, while other one said to include more datasets and provide an ability for users to select a shape from a list of shapes for nodes such as squares or hexagons.

The scores of SUS have been converted to a new number of all items by normalizing the scales to a range from (0-4). For positive formulated questions (or odd questions), the normalization is as follow: for the highest score, 4 is given to strongly agree and 0 to strongly disagree. But, for negative expressed questions (even questions), the range is given as 0 to strongly agree and 4 to strongly disagree. Later, the numbers are multiplied by 2.5 to transform the original scores from 0-40 to 0-100.

Based on studies, a score of a SUS survey that is below 68 is considered as below average, and above that benchmark considers above average. The SUS scores for the proposed application are 76.01 which exceed by far the benchmark of 68. However, further improvements can be made to deliver

even higher levels of usability and satisfaction. The evaluation results for each question can be seen from Average Scores to SUS Questions.

TABLE I. AVERAGE SCORES TO SUS QUESTIONS

Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Over-all
3.1 5	2.7 9	3.2 7	2.8 4	3.0 2	2.9 2	2.9 5	3.1 3	3.4 2	2.9 2	76.0 1

VII. CONCLUSION AND FUTURE WORK

The amount of publishing data consistent with the standards of Linked Data is growing dramatically. But, consumption is still limited for professionals who understand the technologies of Linked Data. Thus, a tool for intuitive presentation of Linked Data is crucial. LOD Explorer, an interactive and easy-to-use tool for exploring RDF resources, is presented. The application is made using pure JavaScript and jQuery libraries without the need for a server-side software. An evaluation of the application is employed using the known user survey System Scalability Scale (SUS) tool, and the evaluation results were by far acceptable.

The future plans for the tool are to enrich it with several further functions such as adding more RDF datasets, giving users an opportunity to select a desired shape for the nodes, adding pathfinding feature so as to find the exact relationship between two or more resources.

REFERENCES

- [1] G. Madhu, D. A. Govardhan, and D. T. Rajinikanth, "Intelligent Semantic Web Search Engines: A Brief Survey," ArXiv Prepr. ArXiv11020831, 2011.
- [2] G. Marchionini, "Exploratory Search: From Finding to Understanding," Commun ACM, vol. 49, no. 4, pp. 41–46, Apr. 2006.
- [3] R. Mirizzi, A. Ragone, T. D. Noia, and E. D. Sciascio, "Semantic Wonder Cloud: Exploratory Search in DBpedia," in Current Trends in Web Engineering, F. Daniel and F. M. Facca, Eds. Springer Berlin Heidelberg, 2010, pp. 138–149.
- [4] T. Jiang, "Exploratory Search: A Critical Analysis of the Theoretical Foundations, System Features, and Research Trends," in Library and Information Sciences, Springer, 2014, pp. 79–103.
- [5] G. Marchionini and B. Shneiderman, "Finding Facts vs. Browsing Knowledge in Hypertext Systems," Computer, vol. 21, no. 1, pp. 70–80, Jan. 1988.
- [6] K. Jacksi, N. Dimililer, and S. R. Zeebaree, "a survey of exploratory search systems based on lod resources," Proc. 5th Int. Conf. Comput. Inform. ICOCI 2015, pp. 501–509, 2015.
- [7] J. A. R and M. Kurian, "A Survey on Tools essential for Semantic web Research," Int. J. Comput. Appl., vol. 62, no. 9, pp. 26–29, Jan. 2013.
- [8] T. Berners-Lee, J. Hendler, and O. Lassila, "The semantic web," Sci. Am., vol. 284, no. 5, pp. 28–37, 2001.
- [9] T. Berners-Lee, "Linked Data - Design Issues," Linked Data, 27-Jul-2006. [Online]. Available: <http://www.w3.org/DesignIssues/LinkedData.html>. [Accessed: 20-Dec-2017].
- [10] K. Krieger and D. Rosner, "Linked Data in E-Learning: A Survey," Semantic Web 0, pp. 1–9, 2011.
- [11] T. Heath and C. Bizer, Linked Data: Evolving the Web into a Global Data Space, 1st edition. Morgan & Claypool., 2011.
- [12] G. Klyne, J. J. Carroll, and B. McBride, "Resource Description Framework (RDF): Concepts and Abstract Syntax," Resource Description Framework (RDF): Concepts and Abstract Syntax, 10-Feb-2004. [Online]. Available: <http://www.w3.org/TR/2004/REC-rdf-concepts-20040210/>. [Accessed: 20-Dec-2017].

- [13] C. Bizer, T. Heath, and T. Berners-Lee, “Linked data-the story so far,” *Int. J. Semantic Web Inf. Syst.*, vol. 5, no. 3, pp. 1–22, 2009.
- [14] “DBpedia version 2016-10 | DBpedia.” [Online]. Available: <http://wiki.dbpedia.org/datasets/dbpedia-version-2016-10>. [Accessed: 23-Dec-2017].
- [15] P. N. Mendes, M. Jakob, A. García-Silva, and C. Bizer, “DBpedia spotlight: shedding light on the web of documents,” presented at the Proceedings of the 7th international conference on semantic systems, 2011, pp. 1–8.
- [16] V. Geroimenko and C. Chen, *Visualizing the semantic web: XML-based internet and information visualization*. Springer Science & Business Media, 2006.
- [17] K. Jacksi, N. Dimililer, and S. R. Zeebaree, “State of the Art Exploration Systems for Linked Data: A Review,” *Int. J. Adv. Comput. Sci. Appl. IJACSA*, vol. 7, no. 11, pp. 155–164, 2016.
- [18] R. Mirizzia, A. R. T. Di Noiaa, and E. Di Sciascioa, “Lookup, Explore, Discover: how DBpedia can improve your Web search,” 2010.
- [19] A. Musetti et al., “Aemoo: Exploratory search based on knowledge patterns over the semantic web,” *Semantic Web Chall.*, 2012.
- [20] D. V. Camarda, S. Mazzini, and A. Antonuccio, “Lodlive, exploring the web of data,” presented at the *Proceedings of the 8th International Conference on Semantic Systems*, 2012, pp. 197–200.
- [21] A. Micsik, Z. Tóth, and S. Turbucz, “LODmilla: Shared Visualization of Linked Open Data,” presented at the *Theory and Practice of Digital Libraries–TPDL 2013 Selected Workshops*, 2014, pp. 89–100.
- [22] D. Lukovnikov, C. Stadler, and J. Lehmann, “LD viewer-linked data presentation framework,” presented at the *Proceedings of the 10th International Conference on Semantic Systems*, 2014, pp. 124–131.
- [23] A. Bangor, P. Kortum, and J. Miller, “Determining what individual SUS scores mean: Adding an adjective rating scale,” *J. Usability Stud.*, vol. 4, no. 3, pp. 114–123, 2009.