Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 112 (2017) 854–863

Procedia Computer Science

www.elsevier.com/locate/procedia

International Conference on Knowledge Based and Intelligent Information and Engineering Systems, KES2017, 6-8 September 2017, Marseille, France

Visualizing Large-scale Linked Data with Memo Graph

Fatma Ghorbel[a,b*], Fayçal Hamdi[a], Nebrasse Ellouze[b], Elisabeth Métais[a], Faiez Gargouri[b]

[a]CEDRIC Laboratory, Conservatoire National des Arts et Métiers (CNAM), Paris, France
[b]MIRACL Laboratory, University of Sfax, Sfax, Tunisie

Abstract

Many studies, in the literature, have affirmed a low level of user satisfaction concerning the understandability and readability of large-scale Linked Data visualizations offered by current available tools. This issue is especially problematic for inexperienced users. To address these requirements, we have extended our previous work Memo Graph, an ontology visualization tool, to provide a user-centered interactive solution for extracting and visualizing Linked Data. It takes aim to provide comprehensible and legible visualization. To manage scalability, it is built on an incremental approach to extract descriptive summarization from a given Linked Data endpoint where it becomes possible to generate a "summary graph" from the most important data (middle-out navigation approach). It offers user interfaces that reduce task complexity for users, especially the inexperienced ones. We tested Memo Graph on a number of Linked Data datasets with encouraging results. We discuss the promising results derived from an empirical evaluation, which affirmed that Memo Graph is useful in visualizing Linked Data and usable.

© 2017 The Authors. Published by Elsevier B.V.
Peer-review under responsibility of KES International

Keywords: Linked Data; Visualization; User-Orientation; Scalability; Data Extraction; Linked Data Summarization; Best Descriptors.

1. Introduction and Motivation

Nowadays, Linked Data is increasingly used in a variety of contexts[1]. Thus, an increasing number of people in the modern knowledge society get in contact with it. It is no longer exclusively used by domain experts but also by inexperienced users.

Data verification, exploration and sensemaking are of great importance in the field of Linked Data[2]. However, it is difficult for humans to manually analyze and explore data[1]. This problem becomes worse with casual users[1].

Visualization of Linked Data helps to overcome this hurdle[1, 3, 4]. It provides users an intuitive way to explore the content[2, 5], identify more easily errors[3], infer correlations and causalities[2] and give expert users new perspectives.

* Corresponding author. *E-mail address:* fatmaghorbel6@gmail.com

1877-0509 © 2017 The Authors. Published by Elsevier B.V.
Peer-review under responsibility of KES International
10.1016/j.procs.2017.08.079

The interweaving of the Linked Data into the Web of Data is regularly done at large-scale[3]. As a consequence, visualizing large-scale datasets has become a major research challenge, of which readability and understandability are vital requirements[2, 3, 6].

Several visualizations for Linked Data datasets have been proposed in the last decade[1, 5, 6, 7]. However, many studies indicated a low level of user satisfaction concerning (1) the readability of large-scale Linked Data visualizations[3, 5] and (2) the lack of useful mechanisms for "content-level visualization", including support for visual descriptive summaries, overviews and selective visualization of Linked Data dataset parts[2, 3, 5]. These issues are especially problematic for casual users.

This requires a new generation of user-friendly visualization tools that offer readable visualizations of Linked Data datasets for all users[3, 5] and provide visual descriptive summaries of the large-scale ones[3, 5, 8].

Attempting to address these requirements, we have extended our earlier work Memo Graph[9], an ontology visualization tool for everyone, to be able in its new version to offer a solution for extracting and visualizing Linked Data, as a way to offer concise readable overviews of the large-scale ones, and support a "middle-out" navigation approach, starting from the most information-rich recourses i.e., "best descriptors". It is designed to be used by everybody, including expert and non-expert users. It offers a rich set of interaction and navigation techniques; all tasks proposed by Ben Shneiderman[10] as well as other functionalities.

The remainder of the present paper is structured as follows: Section 2 describes current approaches for visualizing Linked Data. Section 3 presents the extension of our previous work, Memo Graph, in offering an understandable interactive visualization of large-scale Linked Data. Section 4 gives some applications of the developed intervention. In Section 5, we present the evaluation the usability and the usefulness of Memo Graph in visualizing Linked Data that it is conducted on experienced semantic web users. Section 6 draws conclusions and future research directions.

2. State of the Art: Linked Data Visualization Approaches

Dadzie and Rowe proposed the most comprehensive survey of existing Linked Data visualization approaches[3]. They divided the approaches into 2 categories: browsers with visualization options and text-based. They concluded that the majority of the implemented approaches are designed to be used only by domain expert users and do not provide overviews on the dataset.

We organize Linked Data visualization approaches into 2 categories: "Approaches proposed specifically for visualizing Linked Data" and "Approaches proposed mainly for visualizing RDF ontologies supporting the specific requirements of Linked Data". Table1 summarizes the characteristics of some Linked Data visualization approaches.

Table 1. Comparison of some Linked Data visualization approaches.

	Visualization type	Interaction								Development platform	Non-expert users
		Filter	Query (formal syntax)	Query (format or keywords)	Zoom	Overview	Detail on demand	Relate	History		
"Approaches proposed specifically for visualizing Linked Data"											
RelFinder*	Graph.	*	*		*	*	*	*		Web application	
Explorator[11]	Graph.	*	*	*			*			Web application	
Fenfire[12]	Graph.	*			*		*	*		Web application	
LODWheel[13]	Graph and chart.	*			*	*	*	*		Web application	*
Rhizomer[14]	Map, timeline and chart.	*		*	*	*	*	*		Web application	*

* http://relfinder.dbpedia.org

LD-VOWL[15]	Force directed graph (VOWL elements).	*			*	*	*	*		Web application	*
LodLive[16]	Graph.		*	*			*			Web application	
Semaplorer[17]	Map and media.			*	*	*	*	*	*		
Tabulator[18]	Map and timeline.		*	*			*	*	*	Web application	*
LODVizSuite[19]	Treemap, map, tree and bubble chart.				*	*		*			*
Exhibit[†]	Map, timeline and chart.	*		*			*			Web application	*
LESS[20]	Thumbnail.		*				*			Web application	
Zitgist[‡]	Text-based visualization.	*					*		*	Web application	
DBpedia Viewer[21]	Map.	*		*		*	*			Web application	*
VIZVIVO[22]	Graph.	*			*	*	*	*	*	Web application	
Linked Data Maps[23]	Map.	*		*	*	*	*	*		Web application	
LODeX[24]	Graph.	*	*	*			*	*		Web application	
URI Burner[§]	Text-based visualization.		*	*			*			Web application	
Sig.ma[**]	Text-based visualization.	*					*		*		
VizBoard[25]	Chart, scatter and treemap.	*				*	*	*		Web application	
CODE Visualization Wizard[††]	Map.	*	*	*	*	*		*	*	Web application	*
DBpedia Mobile[26]	Map.	*	*	*	*	*	*	*	*	Mobile application	*
Disco[‡‡]	Text-based visualization.						*			Web application	
OpenLink Data Explorer[§§]	Node-link graph.	*	*	*			*	*	*	Web application	
rdf:SynopsViz[27]	Chart, timeline and treemap.	*			*	*				Web application	*
Payola[28]	Chart, tree and circles.		*		*	*	*			Web application	
Map4rdf[29]	Map.	*			*		*			Web application	
Sgvizler[30]	Chart, graph, timeline, map, and treemap.	*			*	*	*	*		Web application	
Dipper[***]	Text-based presentation.						*			Web application	
LDVizWiz[31]	Map, pie and tree.				*	*	*			Web application	
Piggy Bank[32]	Text-based visualization.			*			*		*	Web application	
Marbles[†††]	Text-based visualization.		*				*			Web application	
"Approaches proposed mainly for visualizing RDF ontologies supporting the specific requirements of Linked Data"											
RDF Gravity[‡‡‡]	Graph.	*		*	*	*		*		Java application	
IsaViz[§§§]	Graph			*	*	*		*		Java application	

[†] http://www.simile-widgets.org/exhibit
[‡] http://dataviewer.zitgist.com
[§] inkeddata.uriburner.com
[**] http://sig.ma
[††] http://code-research.eu/
[‡‡] http://www4.wiwiss.fu-berlin.de/bizer/ng4j/disco
[§§] http://ode.openlinksw.com/
[***] http://api.talis.com/stores/iand-dev1/items/dipper.htm
[†††] mes.github.io/marbles
[‡‡‡] http://semweb.salzburgresearch.at/apps/rdf-gravity/
[§§§] https://www.w3.org/2001/11/IsaViz/overview.html

Most available Linked Data visualization approaches are designed to be used only by expert users[1, 3, 7]. For instance, LDVizWiz and Sgvizler are based on SPARQL queries to generate visualizations. Other tools use semantic web-related jargon in the user interface. All text-based visualization approaches, expect Piggy Bank, target experienced users. The others few approach, targeting non-expert users, are not easy to set up and use[3, 5].

Only few approaches support visualization of large-scale Linked Data. However, they "fail to take into account issues related to performance"[2]. Most of these approaches require the loading of the whole graph on the user interface e.g., Fenfire, LODWheel, RelFinder, RDF Gravity, IsaViz and LODeX, which has a negative impact on the understandability and readability of the visualization; especially for non-expert users. To manage scalability, current approaches are based on 2 main techniques. The first method consists on suggesting a reasonable workflow for datasets exploration: "1-Overview, 2-Filter, 3-Visualize" e.g., Rhizomer and RDF Gravity. The second method uses the clustered graph navigation paradigm. All text-based approaches do not handle scalability[1]. All scalable graph-based approaches do not offer an efficient readable visualization[1, 6]. DBpedia Viewer manages the scalability issue. However, it visualizes only one Linked Data source – DBpedia.

All approaches surveyed do not provide descriptive summary of Linked Data sources. LD-VOWL is based on knowledge base summary approach. However, it visualizes only the schema information of the Linked Data sources and do not display the Assertional Box (ABox).

Graph-based approaches help in obtaining a better understanding of the data structure[1, 6]. However, "Bombarding end users with rows of text is not effective"[3].

Only few visualizations approaches show all key elements of the Linked Data sources. However, a smaller number of these approaches offer different and clear visual representations of each element. For instance, RDF Gravity does not provide a visual distinction between object properties and datatype properties.

Only very few approaches extract and visualize the schema information of the Linked Data sources[6] e.g., CODE, VizBoard and LD-VOWL.

Only very few ontology visualization tools support the visualization of Linked Data sources.

3. Visualizing Linked Data with Memo Graph

This section focuses on visualizing Linked Data with Memo Graph. It is an ontology visualization tool that supports the particular requirements for visualizing Linked Data. It is designed to be used by all, including expert and non-expert users. It tends to offer an understandable visualization for both small-scale and large-scale inputs.

3.1. Force-Directed Graph-Based Visualization

Memo Graph visualizes Linked Data sources as graphs, using a force-directed algorithm. This method reflects the relative importance of the nodes in the generated graph as it arranges the nodes in a way that the highly connected ones are placed more to the center of the visualization, while the less connected ones are placed in the periphery. It increases the legibility of the graph as it tends to avoid the problem of edge crossings. The force-directed visualization may be adapted, as the attraction forces between the different nodes can be modified in favour of a manual repositioning of the elements.

Nodes are identified using both labels and pictures. The picture is automatically extracted from Google if it is not provided by the user. The use of the picture makes nodes distinguishable from each other. Role relations between the related nodes are represented using labeled edges. Memo Graph extracts and visualizes the schema information of the Linked Data dataset. Class nodes are slightly larger than instance nodes.

The layout of the user interface is designed to be simple as possible. The interface is divided into 3 parts: The "MEMO GRAPH Viewer" displaying the visualization, the "MEMO GRAPH Details" listing details about a selected graph node, and the "MEMO GRAPH Search" providing a keyword search option.

3.2. Incremental Best Descriptors Extraction Approach

Memo Graph takes aim to find a balance between visualizing large-scale Linked Data endpoints and avoiding no readable dense visualizations. It is built on Incremental Best Descriptors Extraction (IBDE) algorithm– *incremental* being the operative word. The data extraction and visualization process is going to be undertaken gradually. Initially, Memo Graph generates an “initial graph summary” of N nodes (N is set by the user), then allowing iteratively the generated visualization of additional resources as needed. The extraction process is not done without rhyme or reason. The IBDE algorithm extracts a specified number N of “best descriptors” of a given Linked Data endpoint i.e., the most important elements, which best summarize it. The “best descriptors” are instance having the highest number of object properties.

IBDE algorithm takes a large-scale Linked Data endpoint B as input. It returns the N “best descriptors” in B, which best summarize it. The IBDE algorithm is presented below.

```
Inputs B: large-scale Linked Data endpoint
Outputs N: "best descriptors"
Initialization of N1; S:= Size of B;
Extract N1 "best descriptors" of B;
Enrich the N1 "best descriptors" with the associated properties;
Generate an "initial graph summary"(N1 nodes);
While (The user wants to display Ai additional nodes) do
   If ([Ni-1 + Ai] < S)
      Extract [Ni-1 + Ai] "best descriptors" of B;
      Enrich the [Ni-1 + Ai] "best descriptors" with the associated properties;
      Update the "graph summary"( Ni-1 + Ai nodes);
   Else
      Extract "all descriptors" of B;
      Enrich "all descriptors" with the associated properties;
      Generate the "final graph" (S nodes);
      Return;
   End if
End while
```

Algorithm 1. Incremental Best Descriptors Extraction algorithm.

3.3. Interaction Techniques Provided by Memo Graph

Memo Graph offers all interaction tasks proposed by Ben Shneiderman as well as 2 other interactions techniques:

- Overview: Although Memo Graph is based on an incremental Linked Data summarization approach, it provides the alternative to show the overview of the whole knowledge base;
- Filter: This functionality is provided to let users focus on specific elements in the graph;
- Zoom: The right-clicking on the visualization allows users to zoom-out to analyze the whole structure of the graph or zoom-in to explore certain graph parts;
- Keyword search: Memo Graph supports the keyword search task. It highlights the concerned element in the graph and display details about it. A set of pre-specified SPARQL queries are used. We provide 2 modalities for typing: the traditional mode and the dictation or speech-to-text modality. We give the possibility of alternating between modalities;
- Relate: Memo Graph shows the relationships between the related nodes;
- Details-on-demand: To ensure the readability of the visualization, we display details only on demand. We display the datatype properties of the node which is selected. Clicking on a node allows it to be bigger and distinguishable from the other ones;
- Extract: This functionality is ensured through the IBDE algorithm;
- History: Memo Graph supports undo/redo actions at both macro and micro levels;
- Animation: Users can pan the graph and move the elements around, which results in repositioning of the nodes by animated transitions, generated by the force-directed layout.

4. Applications of Memo Graph

This section gives some applications of Memo Graph. It is integrated in CAPTAIN MEMO to visualize data organized using the OntoMemo temporal ontology. It is used, as a standalone application, for visualizing a small-scale Linked Data dataset using PersonLink ontology, and a large-scale Linked Data dataset (DBpedia).

4.1. Integration of Memo Graph in CAPTAIN MEMO

The CAPTAIN MEMO memory aid is a semantic web application to help aged people showing early or moderate sign of Alzheimer's disease to palliate mnesic problems (non-expert domain users).

We integrated Memo Graph into CAPTAIN MEMO to generate, for each patient, his or her associated family/entourage tree from data structured using the OntoMemo dynamic ontology. The temporal dimension is concatenated to the concerned object property to be represented, in the graph, as labeled edges. Fig. 1 shows a screenshot of CAPTAIN MEMO which shows the family/entourage tree generated using Memo Graph. The figure shows six instances of the "Person" class and one instance of the "Animal" class. We select the "ABBOTT DRIRA" node to display its details (datatype properties) it in the "details" frame.

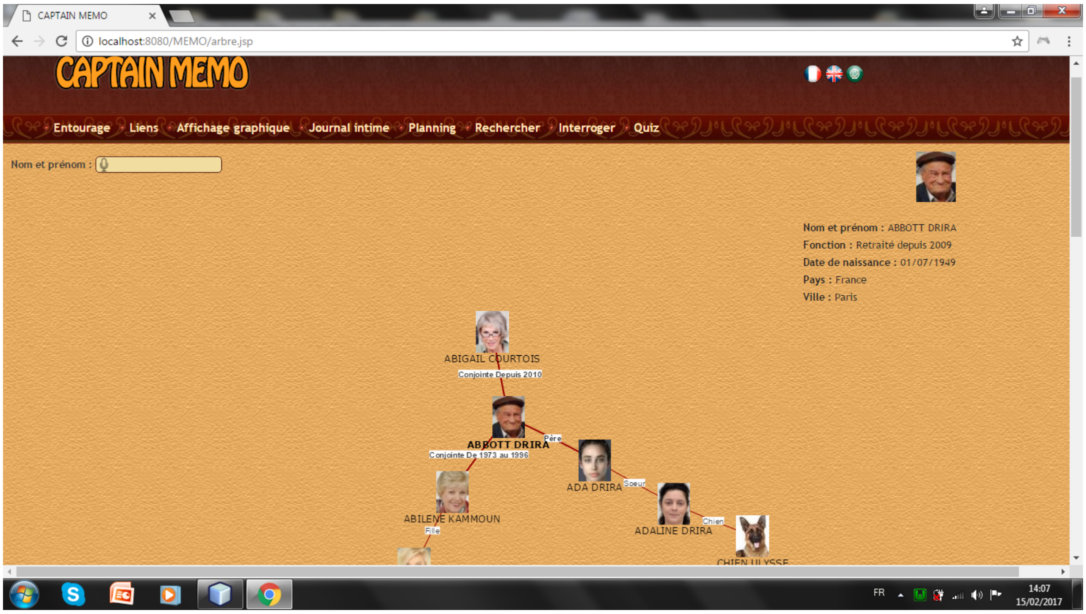

Fig. 1. Screenshot of the CAPTAIN MEMO prosthesis which shows the family/entourage tree generated using Memo Graph.

4.2. Visualization of a Small-Scale Linked Data Dataset using the PersonLink Vocabulary with Memo Graph

Memo Graph is used for visualizing a small-scale Linked Data dataset using the PersonLink vocabulary[33]; as shown in Fig. 2. The figure shows 105 instances of the "Person" class and the associated object properties.

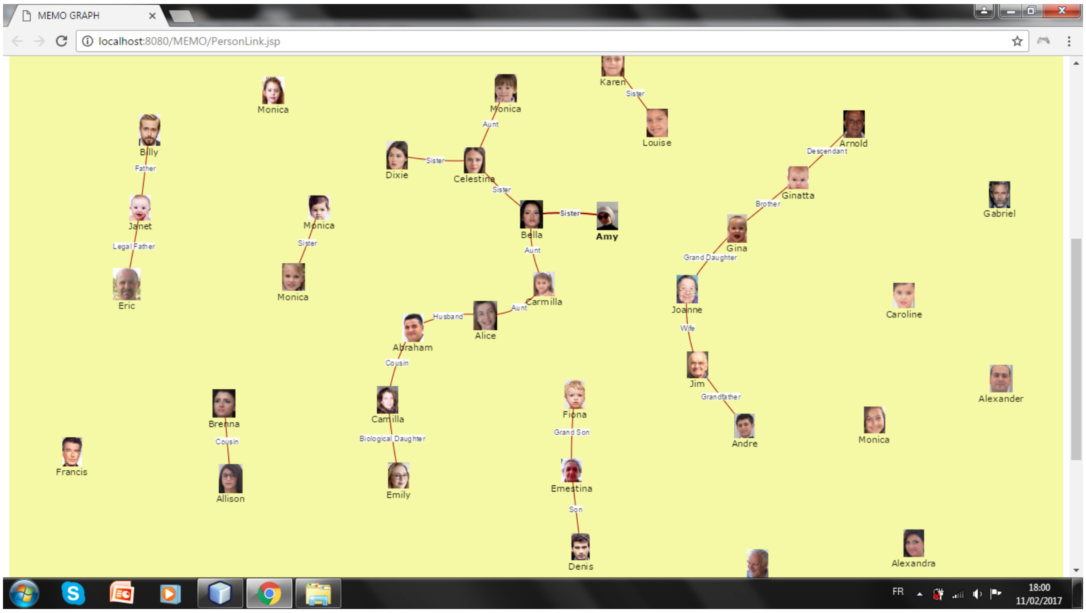

Fig. 2. Screenshot of Memo Graph showing the visualization of a small-scale Linked Data dataset using the PersonLink vocabulary.

4.3. Structure Visualization of a Large-scale Linked Open Data Dataset (DBpedia) with Memo Graph

Memo Graph is tested on DBpedia, large-scale endpoint which is a semantic knowledge base built from structured and extracted information from Wikipedia. Do date; DBpedia covers 685 classes described by 2,795 datatype and object properties and about 4,233,000 instances. Fig. 3 shows a screenshot of Memo Graph that it is utilized for visualizing N_1= 20 "instance best descriptors" of the "Sport" class.

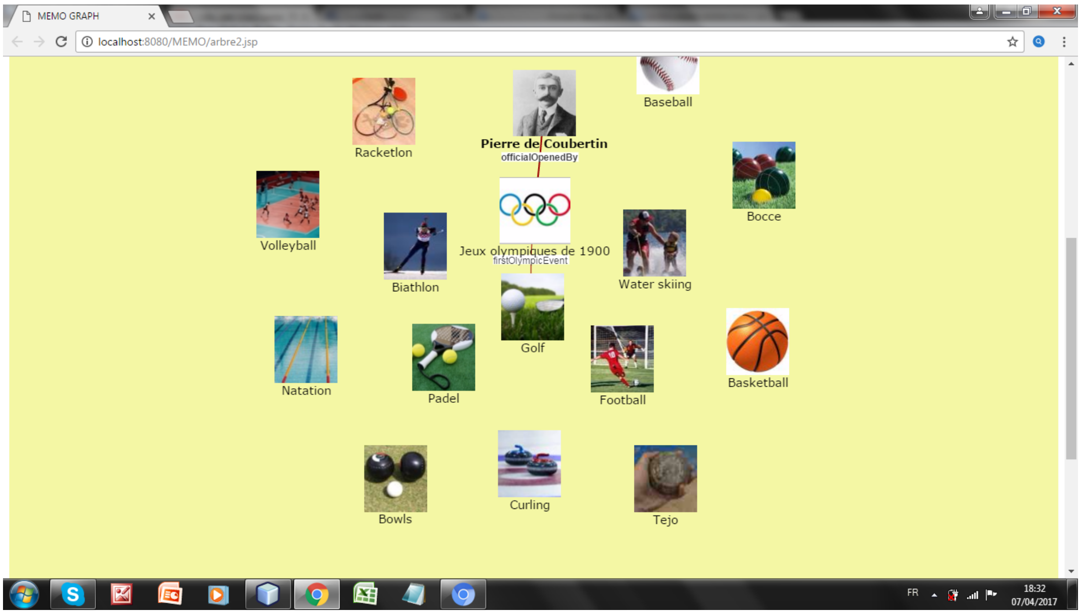

Fig. 3. Screenshot of Memo Graph that it is utilized for visualizing 20 "instance best descriptors" of the "Sport" class (DBpedia).

5. Evaluation

We conducted a human-centered study to evaluate the usability and the usefulness of Memo Graph in visualizing Linked Data. 16 domain expert users entered the study. They were researcher members from the MIRACL laboratory (University of Sfax, Tunisia). Participants were selected based on their scientific profiles. Before beginning the experiment, they were asked to fill a questionnaire about their prior knowledge in the field of semantic web. The participants were allocated into 2 groups, labeled GpA and GpB. Members of the group GpA carried out the experiment using RDF Gravity. Members of group GpB used Memo Graph. None of the participants had any prior experience with the visualization tool that he or she used in this study.

RDF Gravity supports both visualization of Linked Data endpoints and ontologies. It displays all key elements. However, it does not provide a visual distinction between object properties and datatype properties. It offers zoom, search and filter interaction tasks. However, the search functionality does not highlight the searched elements in the visualization. Without applying any filter, the visualization appears very dense, which has a negative impact on its understandability and readability.

5.1. Evaluation of the Usefulness of Memo Graph

The evaluation of the usefulness of Memo Graph consisted of 3 different phases: “quantitative study”, “qualitative feedback” and “free exploration”.

5.1.1. Quantitative Study

We prepared 3 quantitative tasks. For each task, the participant was given a 5 minutes time slot to give the right response. Within 5 minutes, the task was recorded as “failed”. Table 2 summarizes the results of this study.

Table 2. Linked Data engineering tasks and experimental results.

	RDF Gravity (GpA)		Memo Graph (GpB)	
	Mean (min:secs)	Number of “failed”	Mean (min:secs)	Number of “failed”
Task 1: Which is the node that has the highest number of direct connection nodes (object properties)?	04:17	4	01:57	0
Task 2: Which are the data properties of the node “*Sport*”?	04:18	4	00:19	0
Task 3: Which are the object properties of the node “Activity”?	04:54	4	00:29	0
Overall mean and Total number of ‘failed”	04:30	12	00:55	0

All tasks performed with Memo Graph were completed within the time limit; while 12 tasks, performed with RDF Gravity, were recorded as “failed”. The overall mean time of Memo Graph is about 3 minutes and 35 seconds faster than RDF Gravity.

For all tasks, only 4 participants using RDF Gravity completed successfully within the time limit. At the beginning of the session, they applied filters to alleviate the initial dense graph and have a more readable one. The other participants failed to give the correct response.

For the first task, Memo Graph is about 2 minutes and 20 seconds faster than RDF Gravity. This time gap is explained by the fact that Memo Graph displays the Linked Data dataset using a force-directed algorithm.

Compared to RDF Gravity, Memo Graph is about 4 minutes faster, for the second and third tasks. The temporal gap is explained as RDF Gravity does not highlight the searched elements in the graph. Thus, the participants took more time to find the specific element node by visually scanning the visualization.

5.1.2. Qualitative Feedback

After finishing all quantitative tasks, the participants filled a post-session questionnaire on which they had to rate the performance of the tool that they used. The participants answered the questions following IsoMetrics in a range started from 1 to 5. The questionnaire covers the following criteria: “overall reaction”, “clarity”, “interactions techniques”, “findability”, “execution time” and “scalability”. Table 3 shows the results.

Table 3. The overall results of the post-session questionnaire, Participants' perception of RDF Gravity and Memo Graph.

Subjective Feedback	RDF Gravity (GpA)	Memo Graph (GpB)
Overall, I am satisfied with the performance of the used tool.	2.25	4.87
It offers legible graph visualization.	1	5
It offers interactions techniques that meet my expectations.	3	4.87
Keyword search results meet my expectations.	1.5	5
The response time meets my expectations.	4.5	4.75
It is easy to make visually difference between key elements.	2.75	5
It manages very well large-scale Linked Data.	1	4.5
Overall mean	1.96	4.85

The results derived from this phase confirm the results derived from the "Quantitative Study" phase.

5.1.3. Free exploration

All participants were invited to use Memo Graph and give their feedbacks i.e., think-aloud method. We did not give the members of the GpA group a demonstration of Memo Graph. The participants explored the visualization in groups of 3 or 4. They discussed their thoughts and opinions with each other during the experiment. They were encouraged to utilize Memo Graph as long as they wanted. They provided encouraging feedbacks:

- All participants of GpA group autonomously discovered how to use Memo Graph;
- All participants said that the visualization is readable and understandable;
- All participants appreciated the incremental visualization of DBpedia;
- All participants did not have difficulty in distinguishing between the deferent key elements;
- All participants appreciated the use of pictures for visualizing nodes. They pointed out that node can easily be recognized due to the picture. They appreciated that the pictures can be automatically extracted from Google;
- They appreciated the animated force-directed graph. They considered it beneficial, as nodes arrange themselves in a way that the highest connected ones are placed in the center of the screen;
- Members of GpA appreciated that it supports the search task by highlighting the concerned element in the graph;
- Few participants used the zoom mechanism. By taking into account this result, we think that Memo Graph requires a clear way to show that the zoom feature is available (i.e., via a plus and minus zoom buttons);
- Few participants said that the search functionality via the speech-to-text modality is not always useful e.g., using the dictation mode, the classe "TeamSport" is written "Team Sport" which does not correspond to any element.

5.2. Evaluation of the Usability of Memo Graph

Participants filled out a questionnaire to describe their opinion towards the usability of Memo Graph. We used the standard System Usability Scale[34] (SUS). It is a 10-item attitude Likert scale to measure the usability of any system. Scores range on a scale from 1 to 5. To get the SUS score, we multiply the sum of the item score contributions by 2.5. The mean usability score of Memo Graph is 70.9. It indicates a high level of usability.

In our previous work[9], we evaluated the usability of Memo Graph with Alzheimer's disease patients. The results show that it is an easy to use visualization tool.

6. Conclusion

In this paper, we presented an extension of Memo Graph that proposes a comprehensible and legible visualization of Linked Data datasets. Memo Graph was designed to be used by both expert and non-expert users. It tends to avoid the problems of scalability by limiting the number of visible nodes. To do that, Memo Graph is based on an incremental knowledge base summarization algorithm, IBDE, which automatically extract the "most important" concepts in the knowledge source by generating a "graph summary" of the Linked Data dataset, while hiding away the "less important" ones. At the beginning, we elaborated a state of art, focusing on approaches for visualizing Linked Data. This study showed that the majority of the approaches target expert users and do not offer an understandable graph, when visualizing large-scale inputs. Based on these requirements, we developed Memo

Graph. Then, we tested it on a number of Linked Data sources with encouraging results. Finally, we conducted a user-centered study to evaluate the usability and usefulness of Memo Graph in visualizing Linked Data. The results confirm that it is efficient and user-friendly. Future work will be devoted to take into account the time variance.

Reference

1. J. M. Brunetti, S. Auer, R. García, J. Klímek, and M. Nečaský, “Formal linked data visualization model”, In International Conference on Information Integration and Web-based Applications & Services, 2013.
2. N. Bikakis, and T. Sellis, “Exploration and visualization in the web of big linked data: A survey of the state of the art”, 2016.
3. A. S. Dadzie, and M. Rowe, “Approaches to visualising linked data: A survey”, In Semantic Web, vol. 1, no 1-2, 2011.
4. M. Weise, S. Lohmann, and F. Haag, “LD-VOWL: Extracting and Visualizing Schema Information for Linked Data”, In Visualization and Interaction for Ontologies and Linked Data, 2016.
5. G. A. Atemezing, and R. Troncy, “Towards a linked-data based visualization wizard”, In International Conference on Consuming Linked Data-Volume, 2014.
6. F. Valsecchi, M. Abrate, C. Bacciu, M. Tesconi, and A. Marchetti, “Linked Data Maps: Providing a Visual Entry Point for the Exploration of Datasets”, In IESD@ ISWC, 2015.
7. J. Klímek, J. Helmich, and M. Necaský, “Application of the Linked Data Visualization Model on Real World Data from the Czech LOD Cloud”, In LDOW, 2014.
8. O. Peña, U. Aguilera, and D. López-de-Ipiña, “Linked open data visualization revisited: a survey”, In semantic-web-journal, 2014.
9. F. Ghorbel, N. Ellouze, E. Métais, F. Hamdi, F. Gargouri, and N. Herradi, “MEMO GRAPH: An Ontology Visualization Tool for Everyone”, In International Conference on Knowledge Based and Intelligent Information and Engineering Systems, 2016.
10. B. Shneiderman, “The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations”, In IEEE Symposium on Visual Languages, 1996.
11. S. Araujo, D. Schwabe, and S. Barbosa, “Experimenting with explorator: a direct manipulation generic RDF browser and querying tool”, In Visual interfaces to the social and the semantic web, 2013.
12. T. Hastrup, “Browsing Linked Data with Fenfire”, In Linked Data on the Web Workshop at WWW’2008, 2008.
13. M. Stuhr, D. Roman, and D. Norheim, “LODWheel -JavaScript-based Visualization of RDF Data”, In COLD, 2011.
14. J. M. Brunetti, R. Gil, and R. García, “Facets and Pivoting for Flexible and Usable Linked Data Exploration”, In Interacting with Linked Data Workshop, 2012.
15. M. Weise, S. Lohmann, and F. Haag, “LD-VOWL: Extracting and Visualizing Schema Information for Linked Data”, In Visualization and Interaction for Ontologies and Linked Data, 2016.
16. D. V. Camarda, S. Mazzini, and A. Antonuccio, “LodLive, exploring the web of data”, In I-SEMANTICS, 2012.
17. S. Schenk, C. Saathoff, S. Staab, and A. Scherp, “SemaPlorer—interactive semantic exploration of data and media based on a federated cloud infrastructure”, In Web Semantics: Science, Services and Agents on the World Wide Web, Vol. 7, no. 4, 2009.
18. T. Berners-Lee, Y. Chen, L. Chilton, D. Connolly, R. Dhanaraj, R. Hollenbach, and D. Sheets, “Tabulator: Exploring and analyzing Linked Data on the semantic web”, In International Semantic Web User Interaction Workshop, 2006.
19. L. De Vocht, A. Dimou, J. Breuer, M. Van Compernolle, R. Verborgh, E. Mannens, P. Mechant, and R. Van de Walle, “A visual exploration workflow as enabler for the exploitation of linked open data”, In Proceedings of the 3rd Workshop Intelligent Exploration of Semantic Data, 2014.
20. S. Auer, R. Doehring, and S. Dietzold, “LESS –Template-Based Syndication and Presentation of Linked Data”, In ESWC, 2010.
21. D. Lukovnikov, D. Kontokostas, C. Stadler, S. Hellmann, and J. Lehmann, “DBpedia Viewer-An Integrative Interface for DBpedia leveraging the DBpedia Service Eco System”, In LDOW, 2014.
22. M. Javed, S. Payett, J. Blake, and T. Worrall, “VIZ–VIVO: Towards Visualizations-driven Linked Data Navigation”, Visualization and Interaction for Ontologies and Linked Data, 2016.
23. F. Valsecchi, M. Abrate, C. Bacciu, M. Tesconi, and A. Marchett, “Linked Data Maps: Providing a Visual Entry Point for the Exploration of Datasets”, IESD@ ISWC, 2015.
24. F. Benedetti, L. Po, and S. Bergamaschi, “A Visual Summary for Linked Open Data sources”, In ISWC, 2014
25. M. Voigt, S. Pietschmann, L. Grammel, and K. Meißner, “Context-aware Recommendation of Visualization Components”, In eKNOW, 2012.
26. C. Becker and C. Bizer, “Exploring the Geospatial Semantic Web with DBpedia Mobile”, In Web Sem, 2009.
27. N. Bikakis, M. Skourla, and G. Papastefanatos, “rdf:SynopsViz- a framework for hierarchical linked data visual exploration and analysis”, In European Semantic Web Conference, 2014.
28. J. Klímek, J. Helmich, and M. Necaský, “Payola: Collaborative Linked Data Analysis and Visualization Framework”, In ESWC, 2013.
29. A. D. Leon, F. Wisniewki, B. Villazón-Terrazas, and O. Corcho, “Map4rdf- Faceted Browser for Geospatial Datasets”, In Using Open Data: policy modeling, citizen empowerment, data journalism, 2012.
30. M. G. Skja eveland, “Sgvizler: A javascript wrapper for easy visualization of sparql result sets”, In Extended Semantic Web Conference, 2012.
31. G. A. Atemezing and R. Troncy, “Towards a linked-data based visualization wizard”, In COLD, 2014.
32. D. Huynh, S. Mazzocchi, and D. R. Karger, “Piggy Bank: Experience the Semantic Web Inside Your Web Browser”, In ISWC, 2005.
33. N. Herradi, F. Hamdi, E. Métais, A. Soukane, “PersonLink: A Multilingual and Multicultural Ontology Representing Family Relationships”, KEOD 2015, 2015.
34. J.R. Lewis, J. Sauro, “The factor structure of the system usability scale”, In: International Conference on Human Centered Design, 2009.