Adapting the LodView RDF Browser for Navigation over the Linguistic Linked Open Data Cloud in Russian and the Languages of Russia

Konstantin Nikolaev[1] and Alexander Kirillovich[1,2]

[1]Kazan Federal University, Kazan, Russia
[2]Joint Supercomputer Center of the Russian Academy of Sciences, Kazan, Russia

konnikolaeff@yandex.ru, alik.kirillovich@gmail.com

Abstract. This paper is dedicated to using of LodView RDF browser for navigation on the Linguistic Linked Open Data cloud in Russian and languages of Russia. We reveal several limitations of LodView, that prevents its using for this purpose. These limitations are: 1) resolution of Cyrillic URIs; 2) Cyrillic URIs in Turtle representations of resources; 3) support for Cyrillic literals; 4) support for URIs with IDs of fragments; 5) human-readable URLs for RDF representations of resources; 6) deployment of embedded resources. We updated the LodView for fix the recovered limitations.

Keywords: LodView, RDF browser, Linked Open Data, Linguistic Linked Open Data.

1 Introduction

Linked Open Data (LOD) is a powerful and flexible tool for sharing and distributing information. The task of LOD is to establish a link between data and its meaning, which allows search engines to move from a keyword-based approach to one based on the meaning of the search query. In addition, research based on the semantic approach to information allows us to identify more significant characteristics of the text, as opposed to a simple syntactic approach. LOD is most actively used in medicine (in the task of establishing links between diseases and possible treatments), in the scientific literature in the task of structuring citations in a huge number of articles published on the web. In addition, LOD finds its active application in problems of computational linguistics: in particular, problems of semantic parsing of texts and problems of generating texts in natural language. Having a prearranged LOD cloud based on the subject of the text allows one to analyze the contexts of words in the text more correctly and suggest the most likely phrases in the generated texts.

On the other hand, it is necessary to develop high-quality tools that allow ordinary users or researchers from other areas that are not related to LOD development to fully interact, research and analyze the data presented in LOD sets. Visual data analysis allows one to combine the capabilities of a computer and human research abilities, to identify new patterns in complex interconnected data. The steps involved in data visualization are as follows:

- Defining data sources
- Getting and extracting data
- Data modeling
- Data preparation
- Visual preparation
- Data visualization

In most cases, all stages except the last one are performed by the dataset creators, since the visualization tools that perform data preparation and visual data preparation cannot be universal for different types and structures of datasets.

There are many ways to visualize data, ranging from basic visualization using histograms, charts of various types (point, column, circle, etc.), to visualization using graphs. In addition, these types of visualizations are often extended by interaction capabilities. Yi et al. [1] provide a taxonomy of ways to interact with displayed data that can be applied in various ways.

- Selection of items
- Exploring the content of a specific element
- Reconfigure: meaning the user can change the location of elements in the visualization field
- Encode: changing the appearance of elements
- Abstract: changing the level of abstraction of information as necessary
- Filter: ability to custom filter the currently displayed data.
- Connect: display and hide links between elements.

These visualization methods with interaction capabilities are suitable for small datasets and are often used in such tasks. However, when visualizing large amounts of data, other methods that take into account the features of big data are necessary. The following are some of the techniques used by these methods:

- Data reduction: a technique based on the use of approximation techniques, which are used to calculate abstract datasets. Many of them are based on sampling, filtering ([2,3]) and aggregation ([4,5])
- Hierarchical data exploration: displaying large datasets as layers with different levels of detail with the ability to navigate between layers ([4,6])
- Progressive data visualization-displaying data in small portions, on the principle of "as fast as possible", a new portion is loaded by any user action ([7,8,9])
- Data structures and indexes: building indexes or dividing a dataset into subgroups according to a specific algorithm (HETree in [4], VisTrees in [10], Hashedcubes in [11] and RawVis in [12]).

Now that we have considered the main methods of data visualization, we need to provide types and examples of existing tools for data visualization. They can be divided into several groups based on the approach to visualization. For each group, we will give some of the most relevant examples.

Browsers and exploratory tools.
Tools of this type are the simplest and earliest implementations of technologies for data visualization. Usually, the data in such tools is presented in a form similar to the classic web browser, but with the addition of functionality for navigation and displaying LD resources. Existing solutions include the following: Haystack[13], Visor, PepeSearch[14], RDFSurveyor. Some solutions use the faceted search technique (/facet[15], BrowseRDF, Explorator[16], Sewelis, Sparklis, tFacet и другие).

Tools using multiple visualization types.
These visualization tools allow the user to choose the desired way of data visualization, such as diagrams, tables, Euler circles, maps, graphs, and so on. Here are a few tools that have the most complete set of visualization capabilities: Rhizomer[17], Payola, SynopsViz, VisWisard[18], YDS.

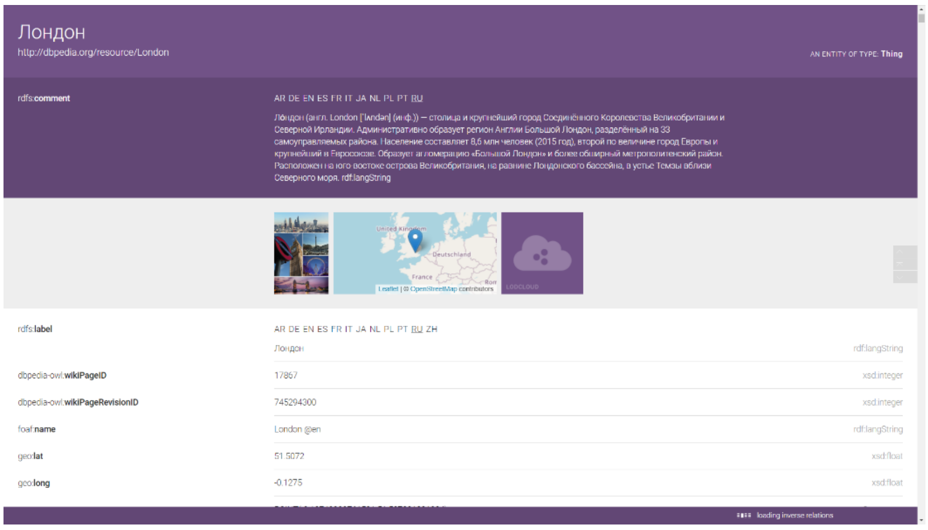
Graph-based visualization tools.
Tools in this category rely on representing LD as graphs. Many of the tools in the previous group support the graph representation, so here we present the tools for which graph representation is the main one: LodLive[19], Aemoo, LODmilla[20], graph-Visdb, RDF4U, H-BOLD, Tarsier etc.

Domain, vocabulary-specific, and device-oriented visualization tools.
This type of tool is intended for a specific type of data. Such tools are not of particular interest in this article, so we will point out just a few examples: LinkedGeoData, Spacetime, Facete.

Ontology visualization tools.
Data visualization tools have been actively developed over the past 20 years. In this section, we should mention tools that are suitable for different LD, rather than for one specific data type. The following services are the most flexible: OWLPropViz (addon for Protégé), OntoStudio, OWLGrEd (addon for Protégé), SOFA, Ontodia. Besides, OntoSphere [21], Onto3DViz [22], and MVOV [23] implement three-dimensional data visualization.

2 LodView

As part of our project related to linguistic LOD, there was a need for a tool that will allow us to display the contents of the Russian linguistic LOD set. LodView[24] was chosen as the optimal one in terms of functionality and simplicity, since it supports displaying fields of various resources in text format, and linguistic data assumes a plain text format. In addition, LodView is based on hierarchical data exploration, meaning it displays a list of objects related to the current one, making it possible to navigate from one object to another. Fig. 1 shows an example of using LodView to visualize the London concept from dbpedia.org.


Лондон
http://dbpedia.org/resource/London
AN ENTITY OF TYPE: **Thing**

rdfs:**comment**	AR DE EN ES FR IT JA NL PL PT RU Ло́ндон (англ. London [ˈlʌndən] (инф.)) — столица и крупнейший город Соединённого Королевства Великобритании и Северной Ирландии. Административно образует регион Англии Большой Лондон, разделённый на 33 самоуправляемых района. Население составляет 8,6 млн человек (2015 год), второй по величине город Европы и крупнейший в Евросоюзе. Образует агломерацию «Большой Лондон» и более обширный метрополитенский район. Расположен на юго-востоке острова Великобритания, на равнине Лондонского бассейна, в устье Темзы вблизи Северного моря. rdf:langString	
rdfs:**label**	AR DE EN ES FR IT JA NL PL PT RU ZH Лондон	rdf:langString
dbpedia-owl:**wikiPageID**	17867	xsd:integer
dbpedia-owl:**wikiPageRevisionID**	745294300	xsd:integer
foaf:**name**	London @en	rdf:langString
geo:**lat**	51.5072	xsd:float
geo:**long**	-0.1275	xsd:float

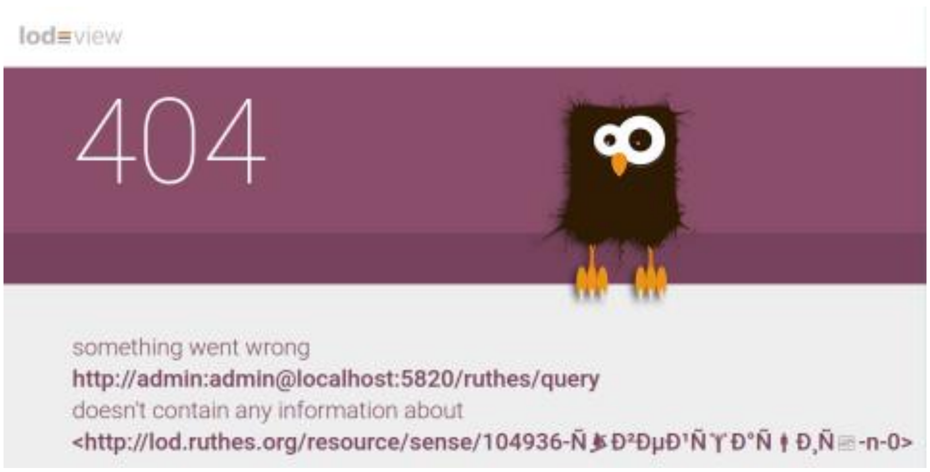
loading inverse relations

Fig. 1. Example of LodView usage in dbpedia.org data visualization

However, LodView requires some improvements, which will be given in the next section.

3 Necessary Improvements of the LodView

1. **Resolution of Cyrillic URLs.** Datasets from the Russian-language LLOD can use URIs containing Cyrillic characters. For example, in the RuThes Cloud set, the URI used to denote the lexical unit "машина" is <http://lod.ruthes.org/resource/entry/RU-машина-n>. However, on some configurations, resolving the Cyrillic URIs in LodView leads to an error (see Figure 2). We must add support for URIs containing Cyrillic characters to LodView.



```
something went wrong
http://admin:admin@localhost:5820/ruthes/query
doesn't contain any information about
<http://lod.ruthes.org/resource/sense/104936-Ñ[illegible]Ð²ÐµÐ¹Ñ[illegible]Ð°Ñ[illegible]Ð¸Ñ[illegible]-n-0>
```

Fig. 2. Error while resolving a link containing Cyrillic literals

2. **Cyrillic URIs in the Turtle representations of resources.** In a machine-readable representation of a resource in the Turtle serialization format, Cyrillic URIs are encoded. Figure 3A shows a fragment of the Turtle representation of the "Автомобиль" concept with encoded URIs. Representing URIs in encoded form makes it difficult to perceive them. In the Turtle representation, URIs must be represented in decoded form (for example, as in Figure 3B)

A) Encoded URIs:

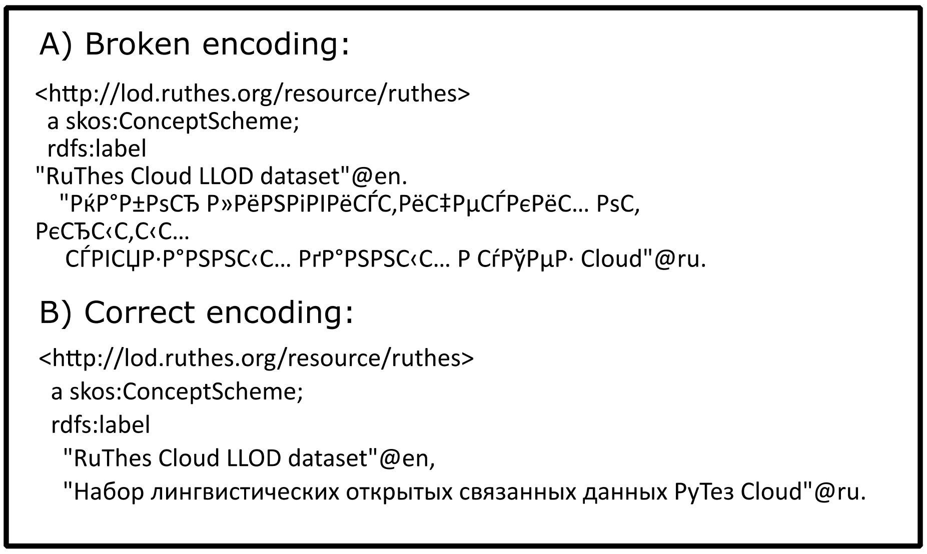
```
<resource/concept/154>
 a skos:Concept;
 lemon:isReferenceOf
   <resource/sense/154-%D0%BC%D0%B0%D1%88%D0%B8%D0%BD%D0%B0-n-0>,
   <resource/sense/154-%D0%B0%D0%B2%D1%82%D0%BE%D0%BC%D0%BE%D0%
B1%D0%B8%D0%BB%D1%8C-n-0>,
 <...>.
```

B) Decoded URIs:

```
<resource/concept/154>
 a skos:Concept;
 lemon:isReferenceOf
   <resource/sense/154-машина-n-0>,   <resource/sense/154-автомобиль-n-0>,
   <...>.
```

Fig. 3. Fragment of a Turtle representation of the "Car" concept from the RuThes Cloud resource with encoded and decoded URIs

3. **Support for Cyrillic literals**. On some configurations, literals are represented with a broken encoding in the machine-readable representation of the resource. This happens even if the dataset has Unicode encoding. Figure 4A shows a Turtle representation of the RuThes Cloud resource with a broken encoding. It is necessary to display Cyrillic literals correctly on all configurations (as in Figure 4B).


A) Broken encoding:

```
<http://lod.ruthes.org/resource/ruthes>
  a skos:ConceptScheme;
  rdfs:label
"RuThes Cloud LLOD dataset"@en.
   "РќР°Р±РѕСЂ Р»РёРЅРіРІРёСЃС‚РёС‡РµСЃРєРёС… РѕС‚
РєСЂС‹С‚С‹С…
    СЃРІСЏР·Р°РЅРЅС‹С… РґР°РЅРЅС‹С… Р СѓРўРµР· Cloud"@ru.
```

B) Correct encoding:

```
<http://lod.ruthes.org/resource/ruthes>
  a skos:ConceptScheme;
  rdfs:label
   "RuThes Cloud LLOD dataset"@en,
   "Набор лингвистических открытых связанных данных РуТез Cloud"@ru.
```

Fig. 4. Fragment of the Turtle representation of the RuThes Cloud resource

4. **Support for URIs with fragment IDs**. Datasets from the LLOD cloud can use URIs that contain Cyrillic fragment IDs. These URIs are used to indicate a resource that is closely related to some other resource. For example, in the RDF representation of the Princeton WordNet, the URI used to denote the lexical unit "cat" is <http://wordnet-rdf.princeton.edu/wn31/cat-n>, and the URI used to indicate the canonical form of this lexical unit is <http://wordnet-rdf.princeton.edu/wn31/cat-n#CanonicalForm>. In this case, the URI for the canonical form of the lexical unit "cat" consists of the URI of this lexical unit, to which the ID of the fragment "#CanonicalForm" is added.

However, LodView does not support URIs containing the IDs of the fragment. So, for example, when resolving the URI <http://wordnet-rdf.princeton.edu/wn31/cat-n#CanonicalForm>, LodView returns a representation not of the resource that is designated by this URI, but of the resource that is designated by the URI <http://wordnet-rdf.princeton.edu/wn31/cat-n>. This problem occurs because of the HTTP Protocol: when sending a request, only a fragment of the URI is sent without the fragment ID. To solve this problem, when resolving any URI, LodView must return a representation not only of the resource designated by this URI, but also of all resources whose URI is formed by combining the URI of the source resource and some fragment ID. For example, it is necessary that when resolving the URI <http://wordnet-rdf.princeton.edu/wn31/cat-n>, LodView returned not only a representation of the resource denoted by this URI, but also of the resource denoted by the URI <http://wordnet-rdf.princeton.edu/wn31/cat-n#CanonicalForm>.

5. **Human-readable URLs for the RDF representation of resources.** LodView implements the negation content principle in the process of resolving URIs. In accordance with this principle, when a user requests the URI of a certain resource, LodView redirects the user to the URL where the web page with human-readable resource representation is located. When a resource URI is requested by a software agent, LodView redirects the agent to a URL with a machine representation of the resource.
   URLs for representing resources as web pages have a user-friendly format: < resource URI>.html. For example, a web page for a resource with the <http://lod.ruthes.org/resource/entry/RU-машина-n> URI has the <http://lod.ruthes.org/resource/entry/RU-машина-n.html> URL. However, URLs for RDF representations of resources are not easy to read. For example, a Turtle representation for a resource with <http://lod.ruthes.org/resource/entry/RU-машина-n> has the <http://lod.ruthes.org/resource/entry/RU-машина-n?output=application%2Frdf%2Bxml> URL. Machine-readable RDF resource representations must have the form like <resource URI>.ttl, <resource URI>.n3 etc. For example, the Turtle representation of <http://lod.ruthes.org/resource/entry/RU-машина-n> resource must have the <http://lod.ruthes.org/resource/entry/RU-машина-n.ttl> URL.
6. **The unfolding of the embedded resources.** On a web page with a resource view, the resource properties are presented as a table. The first column contains the property name, and the second column contains its value.
   If the property value is the URI of another resource, in order to find out the properties of this resource, the user must follow the link of the resource. The system must allow the user to expand embedded resources so that they can see the properties of these embedded resources without leaving the source page.
   If the property value is an anonymous resource (blank node), the property value is the surrogate ID of this anonymous resource, and the description of the anonymous resource itself is located at the bottom of the page. Similarly, the system should allow the user to deploy an anonymous resource to see its properties "in place", without scrolling the page.

4 Implementation of Improvements

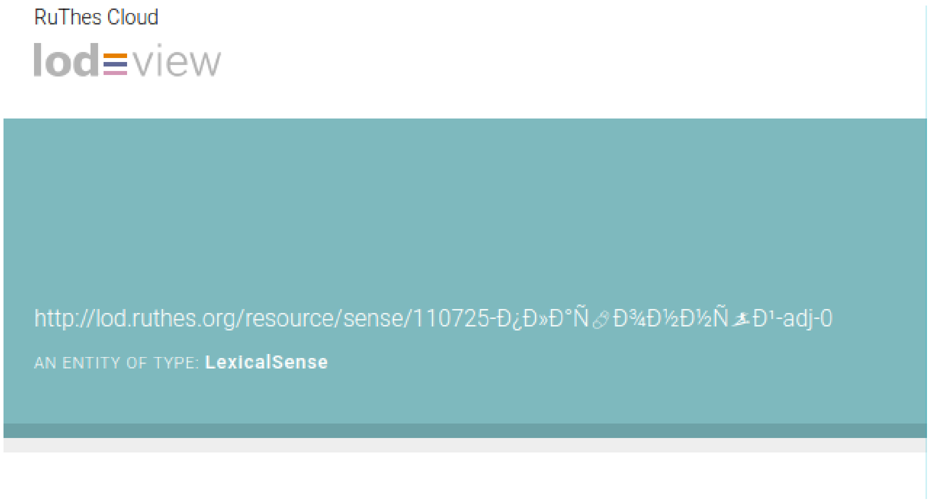
At the moment, we have fully implemented improvements 1-3 and 5, as well as partially implemented improvement 6.

Problem of resolving Cyrillic URIs (improvement №1).

After analyzing the source code of LodView, we found that this problem is related to the fact that the URI text is represented in the SPARQLEndPoint class in Western ISO-8859-1 encoding. This problem was solved by adding a single line of code to the specified class that recodes the URI'I from ISO-8859-1 to Unicode:

IRI = new String (URI.getBytes ("ISO-8859-1"), "UTF-8");

After we solved this problem, LodView began to correctly resolve Cyrillic URLs and return a web page displaying the corresponding resource when accessing them. However, after this, we found a new problem: the resource URI is also displayed incorrectly on the returned web page (see Figure 5).


RuThes Cloud

lod≡view

http://lod.ruthes.org/resource/sense/110725-Ð¿Ð»Ð°Ñ🔗Ð¾Ð½Ð½Ñ⛷Ð¹-adj-0

AN ENTITY OF TYPE: **LexicalSense**

Fig. 5. Incorrect display of the resource URI

We solved this new problem by performing a similar recoding in the code of the corresponding web page (resource.jsp).

Problem with displaying Cyrillic literals (improvement №3).

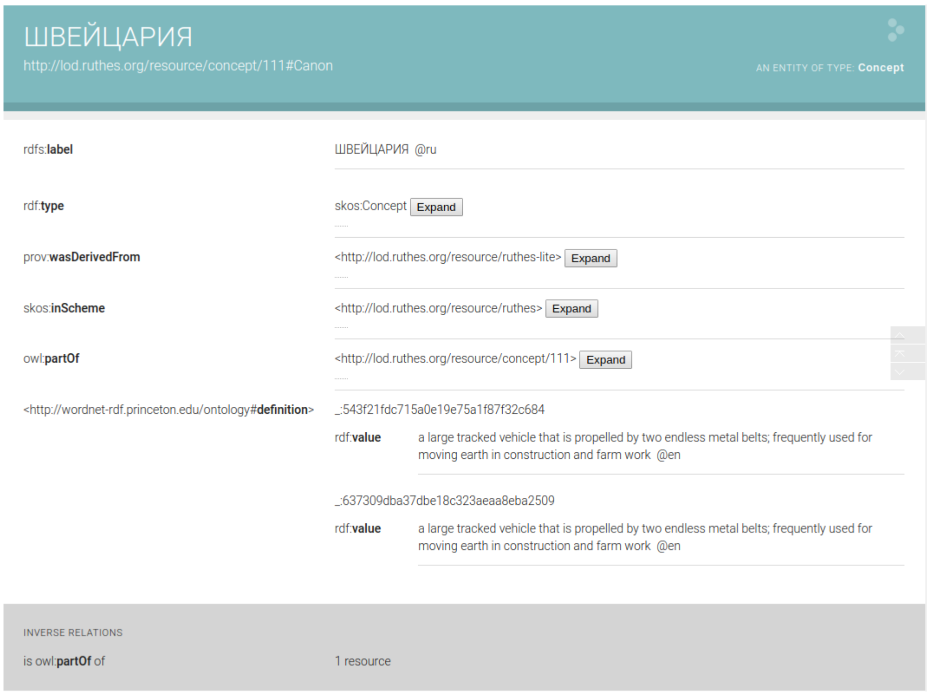
This problem was solved in the same way as the previous one, using a similar recoding.

Problem with encoded URIs (improvement №2).

To solve this problem, we have made changes to the ResourceBuilder class.

Problem with fragment identifiers (improvement №4).

In the course of solving this problem, we encountered that it is not possible to correctly send the # symbol to the server, which is why the server displays the basic version of the resource, and not the requested one. The solution to this problem (although not the most elegant) is to replace the # symbol with another character (rarely used in normal texts and URLS) on the client side and decode it to # on the server side. This way, we can correctly pass the address of the resource we are looking for to the server (Fig. 6).



ШВЕЙЦАРИЯ
http://lod.ruthes.org/resource/concept/111#Canon
AN ENTITY OF TYPE: **Concept**

rdfs:**label**	ШВЕЙЦАРИЯ @ru
rdf:**type**	skos:Concept Expand
prov:**wasDerivedFrom**	<http://lod.ruthes.org/resource/ruthes-lite> Expand
skos:**inScheme**	<http://lod.ruthes.org/resource/ruthes> Expand
owl:**partOf**	<http://lod.ruthes.org/resource/concept/111> Expand
<http://wordnet-rdf.princeton.edu/ontology#**definition**>	_:543f21fdc715a0e19e75a1f87f32c684 rdf:**value** a large tracked vehicle that is propelled by two endless metal belts; frequently used for moving earth in construction and farm work @en _:637309dba37dbe18c323aeaa8eba2509 rdf:**value** a large tracked vehicle that is propelled by two endless metal belts; frequently used for moving earth in construction and farm work @en

INVERSE RELATIONS

is owl:**partOf** of 1 resource

Fig. 6. Blank nodes representation in the resource with fragment identifier.

The disadvantage of this approach is that we can only go to a resource with the fragment ID from another resource. If we enter an address <http://wordnet-rdf.princeton.edu/wn31/cat-n#CanonicalForm> in the address bar, the <http://wordnet-rdf.princeton.edu/wn31/cat-n> resource will be displayed instead. For this reason, in the future we will have to change the structure of the web page that displays the resource content and include related resources with different fragment IDs.

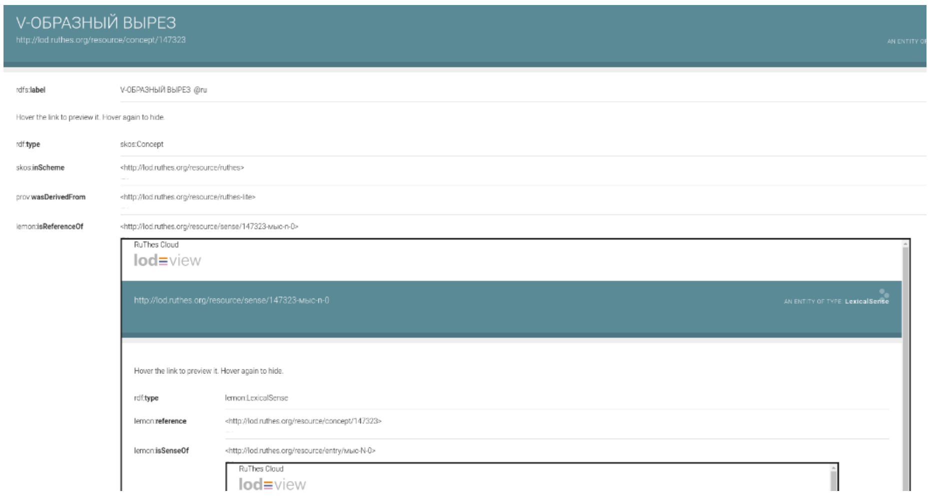
Problem with human-readable URLS (improvement №5).

To solve this problem, we added support for such URIs by making changes to the ResourceController class.

The problem with the deployment of the embedded resources (improvement №6).

This task is completely solved and consists of two changes. For easy viewing of embedded resources, we added the “Expand" button, which displays an iframe containing a page with a web view of the embedded resource (Fig. 7).

Displaying fields in blank nodes was implemented using an existing block with blank nodes. Part of its content associated with a resource that is present in the main block of the web page is moved to the parent element of this resource. Thus, detailed information about the contents of the blank node is displayed directly below it (Fig. 6).



V-ОБРАЗНЫЙ ВЫРЕЗ
http://lod.ruthes.org/resource/concept/147323

AN ENTITY OF

rdfs:**label** V-ОБРАЗНЫЙ ВЫРЕЗ @ru

Hover the link to preview it. Hover again to hide.

rdf:**type** skos:Concept

skos:**inScheme** <http://lod.ruthes.org/resource/ruthes>

prov:**wasDerivedFrom** <http://lod.ruthes.org/resource/ruthes-lite>

lemon:**isReferenceOf** <http://lod.ruthes.org/resource/sense/147323-мыс-n-0>

RuThes Cloud
lod≡view

http://lod.ruthes.org/resource/sense/147323-мыс-n-0

AN ENTITY OF TYPE: **LexicalSense**

Hover the link to preview it. Hover again to hide.

rdf:**type** lemon:LexicalSense

lemon:**reference** <http://lod.ruthes.org/resource/concept/147323>

lemon:**isSenseOf** <http://lod.ruthes.org/resource/entry/мыс-N-0>

RuThes Cloud
lod≡view

Fig. 7. Example of deploying an embedded resource using an iframe

5 Conclusion

This article provides an overview of the main principles, methods, and existing solutions for LD visualization, as well as ways to improve the web application for visualizing resources from triplet stores, aimed at correctly displaying Russian-language resources (or any other ones that use non-Latin characters).

These improvements include: 1) Resolution of Cyrillic URIs; 2) Cyrillic URIs in Turtle representations of resources; 3) Support for Cyrillic literals; 4) support for URIs with IDs of fragments; 5) Human-readable URLs for RDF representations of resources; 6) Deployment of embedded resources.

At the moment, we have fully implemented revisions #1-3 and #5-6, and partially implemented revision #4. The implementation of these improvements is the main contribution of this article.

These improvements will be offered as a fork to the LodView project and will be used as the main resource for viewing the contents of triplet repositories within our project.

Acknowledgements. The work was initiated for navigation over LLOD cloud by K. Nikolaev and A. Kirillovich at KFU, where was funded by Russian Science Foundation according to the research project no. 19-71-10056, and then adapted for navigation over specialized LaTeX-based mathematical collections by A. Kirillovich at JSC RAS.

References

1. Yi, J., Kang, Y., Stasko, J., and A. Jacko, J.: Toward a deeper understanding of the role of interaction in information visualization. IEEE Transactions on Visualization and Computer Graphics (TVCG), 13(6):1224–1231 (2007).
2. Moritz, D., Fisher, D., Ding, B., and Wang, C.: Trust, but verify: Optimistic visualizations of approximate queries for exploring big data. In Conference on Human Factors in Computing Systems (CHI) (2017).
3. Park, Y., Cafarella, M., and Mozafari, B.: Visualization-aware sampling for very large databases. In IEEE International Conference on Data Engineering (ICDE) (2016).
4. Bikakis, N., Papastefanatos, G., Skourla, M., and Sellis, T.: A hierarchical aggregation framework for efficient multilevel visual exploration and analysis. Semantic Web Journal, 8(1), (2017).
5. Godfrey, P., Gryz, J., Lasek, P., and Razavi, N.: Visualization through inductive aggregation. In International Conference on Extending Database Technology (EDBT) (2016).
6. Lins, L., Klosowski, J., and Scheidegger, C.: Nanocubes for realtime exploration of spatiotemporal datasets. IEEE Transactions on Visualization and Computer Graphics (TVCG), 19(1):2 (2013).
7. Angelini, M., Santucci, G., Schumann, H., and Schulz, H.: A review and characterization of progressive visual analytics. Informatics, 5(3):31 (2018).
8. Fekete, J., Fisher, D., Nandi, A., and Sedlmair, M.: Progressive data analysis and visualization. Dagstuhl Seminar 18411, Dagstuhl Reports, 8(10) (2018).
9. Zgraggen, E., Galakatos, A., Crotty, A., Fekete, J., and Kraska, T.: How progressive visualizations affect exploratory analysis. IEEE Transactions on Visualization and Computer Graphics (TVCG), 23(8) (2017).
10. El-Hindi, M., Zhao, Z., Binnig, C., and Kraska, T.: Vistrees: Fast indexes for interactive data exploration. In HILDA (2016).
11. Augusto de Lara Pahins, C., Stephens, S., Scheidegger, C., and Luiz Dihl Comba, J.: Hashedcubes: Simple, low memory, real-time visual exploration of big data. IEEE Transactions on Visualization and Computer Graphics (TVCG), 23(1) (2017).
12. Bikakis, N., Maroulis, S., Papastefanatos, G., and Vassiliadis, P.: RawVis: Visual exploration over raw data. In 22nd European Conf. on Advances in Databases & Information Systems (ADBIS) (2018).
13. Quan D., A., Karger, R.: How to make a semantic web browser. In International World Wide Web Conference (WWW), pp. 255–265 (2004).
14. Vega-Gorgojo, G., Slaughter, L., Giese, M., Heggestøyl, S., W. Klüwer, J., Waaler, A.: Pepesearch: Easy to use and easy to install semantic data search. In Extended 138 BIBLIOGRAPHY Semantic Web Conference (ESWC), pp. 146–150 (2016).
15. Hildebrand, M., van Ossenbruggen, J., and Hardman, L.: Facet: A browser for heterogeneous semantic web repositories. In International Semantic Web Conference (ISWC), pp. 272–285 (2006).
16. F. C. Araújo, S., Schwabe, D., D. J. Barbosa, S.: Experimenting with explorator: A direct manipulation generic RDF browser and querying tool. In Visual Interfaces to the Social and the Semantic Web (2009).
17. Maria Brunetti, J., Auer, S., García González, R., Klímek, J., Necaský, M.: Formal linked data visualization model. In International Conference on Information Integration and Web-based Applications and Services, IIWAS, p. 309 (2013).

18. Tschinkel, G., E. Veas, E., Mutlu, B., Sabol, V.: Using semantics for interactive visual analysis of linked open data. In International Semantic Web Conference (ISWC), pp. 133–136 (2014).
19. Valerio Camarda, D., Mazzini, S., Antonuccio, A.: LodLive, Exploring the Web of data. In International Conference on Semantic Systems, I-SEMANTICS, pp. 197–200 (2012).
20. Micsik, A., Tóth, Z., Turbucz, S.: LODmilla: Shared visualization of linked open data. In Theory and Practice of Digital Libraries—TPDL 2013 Selected Workshops, pp. 89–100 (2014).
21. Bosca, A., Bonino, D., Pellegrino, P.: OntoSphere: More than a 3D ontology visualization tool. In Semantic Web Applications and Perspectives, SWAP (2005).
22. Suigen Guo, S., W. Chan, C.: A tool for ontology visualizaiton in 3D graphics: Onto3DViz. In Canadian Conference on Electrical and Computer Engineering, CCECE (2010).
23. Dmitrieva, J., Verbeek, F.: Node-link and containment methods in ontology visualization. In International Workshop on OWL: Experiences and Directions (OWLED) (2009).
24. LodView homepage, https://lodview.it, last accessed 2020/05/03