LENA\textsuperscript{TR}: Browsing Linked Open Data Along Knowledge-Aspects

Thomas Franz, Jörg Koch, Renata Dividino, Steffen Staab
ISWeb - Information Systems and Semantic Web
University of Koblenz-Landau
\{franz,jmkoch,dividino,staab\}@uni-koblenz.de

Abstract
Browsing linked open data (LOD) is a promising, yet, often unsatisfactory experience today. User-support for the identification of relevant information within the fast-growing cloud of LOD is limited. This paper presents LENA\textsuperscript{TR}, a browser for LOD that highlights relevant information with respect to different knowledge aspects hidden in linked data. Its interpretation of faceted navigation facilitates the sense-making and browsing of LOD, solving many of the shortcomings experienced in LOD browsing today.

Introduction
On the Web of linked open data (LOD) (Bizer, Heath, and Berners-Lee 2009), information is represented uniformly by the resource description framework (RDF) (Antoniou and Harmelen 2008) and interlinked by semantic relations. Instead of application-centric task support, data-centric task support can be implemented upon this infrastructure, i.e. single browsing tools can support the exploration of information originally provided by a variety of different information systems. Several user studies have revealed that this strategy of manual search by browsing through available information is the preferred strategy of humans (Teevan et al. 2004; Blanc-Brude and Scapin 2007; Boardman and Sasse 2004; Ravasio, Schär, and Krueger 2004).

Consequently, the idea of browsing LOD has been taken up in several research works and resulted in the development of several browser applications for LOD such as the research prototypes Tabulator (Berners-Lee et al. 2007), LENA (Koch, Franz, and Staab 2008), and Marbles\textsuperscript{1}.

Exploring interlinked data with these browsers, however, is difficult. None of them supports users in deciding which paths to follow during their exploration of the available data. Users are required to investigate displayed information manually, virtually scanning presented information from top to bottom, in order to spot links that are most relevant, meaningful, and interesting. While technologies like the Fresnel display vocabulary (Pietriga et al. 2006) have been developed to encode statically for a specific information domain and a known vocabulary, which properties and links are relevant, a generic approach that can be applied dynamically is missing today. However, on the web of linked open data, it cannot be assumed to know the vocabularies utilized to describe existing data in advance. The problem of selecting relevant links becomes even harder as users cannot easily investigate the information available behind a link, e.g. investigate the information type, its quantity, and its relevance.

In this paper, we propose LENA\textsuperscript{TR}, a linked data browser that analyses LOD \textit{dynamically} in order to provide a more profound browsing experience for \textit{arbitrary} data available\textsuperscript{2}. LENA\textsuperscript{TR} features facets that describe the data in the vicinity of a currently browsed resource. Each facet corresponds to a particular knowledge-aspect and lists relevant resources for this aspect. Resources shown within facets are in the vicinity of the currently viewed resource, i.e. not necessarily connected by a single link-hop, so that users benefit from the analysis of data that is ahead of their next browsing steps. The analysis employed by LENA\textsuperscript{TR} is TripleRank (Franz et al. 2009), a method for ranking RDF data.

In this paper, we illustrate LOD browsing by a scenario followed by a discussion of its shortcomings. We then present the LOD browser LENA\textsuperscript{TR} and show how it solves the shortcomings identified before. We continue with an overview of the architecture of LENA\textsuperscript{TR} and the integration of the TripleRank method, before we give a detailed description of TripleRank. We summarize the related work done in the field of LOD browsers and finish the paper with a conclusion and outlook of ongoing and future work.

Scenario
In the web of linked data, resources are retrieved by following data level links into various other data sources. Existing linked data browsers such as Tabulator, Lena, Marbles, Fenfire\textsuperscript{3}, razorbase\textsuperscript{4}, and VisiNav\textsuperscript{5} enable users to visualize the available information and navigate along its in/outcoming links.

In our scenario, the user Claudia is browsing for information about the music band \textit{The Beatles}. Much information can be found about this band on the web of LOD, e.g. the re-

\textsuperscript{1}http://marbles.sourceforge.net/
\textsuperscript{2}http://west.uni-koblenz.de/Research/systeme/lena
\textsuperscript{3}http://fenfire.org/
\textsuperscript{4}http://www.razorbase.com/
\textsuperscript{5}http://visinav.deri.org/
source http://dbpedia.org/resource/The_Beatles is linked to music labels that have published albums of the band, it lists people whom they have worked with, countries they have visited, the birthday of each of the band members, birth town and so on.

Figure 1 shows how data about the music band is presented by different LOD browsers. The resource title is displayed at the top associated with corresponding property names and property values. A property value again describes either a dereferenceable resource or a literal. The user can select these property value resources in order to visualize the associated data. For instance, if Claudia would like to get more information about the current members of the band the Beatles, she looks up the property currentMembers and follows one of the links associated to its resource property values.

**Shortcomings**

Browsing data is about adopting an exploratory search approach over a dataspace. Usually, the user is trying to find some information which could not be formulated as a precise query or be completely retrieved by a search engine. By browsing available information, the user learns about and investigates the dataspace by navigating along in/outcoming links. Exploratory search in LOD, however, can be very complex for users. The magnitude of information available today, i.e. at the advent of LOD, makes the identification and quantification of relevant links difficult.

Furthermore, LOD browsing is realized by following data links corresponding to a one-step-at-a-time approach. The user is able to select one available resource at a time; from there he goes on to the next navigation step. This approach, however, limits exploring the sophisticated interrelations of the web of LOD.

At last, even though the linked data principles are being adopted by an increasing number of data providers over the last years, the portion of links across data sets is often low, i.e. the available information is poor linked or inadequate. In such cases, the user is not able to explore a qualitative number of links and consequently he does not learn much more about the information he was seeking by browsing than he would when using other retrieval methods.

Back to our scenario, Claudia is seeking for information about the band The Beatles because she would like to buy one of the band’s albums. For Claudia it is important that the album has many songs that she has already heard. Since Claudia usually cannot remember a song’s name, she learns about the band and albums by reading their discography, by knowing its members, music style, and so on.

Due to the huge amount of information (e.g. the resource
is associated with more than 1000 others re-

writer

in the description of

is a linked data browser that features faceted navi-

that is the top re-

summarize multiple types of links,

in LOD interrupt the browsing process as they

have very different meanings. Some are used to label

associatedActs, associatedBand, associatedMusi-

labels, i.e. publishers of songs by The Beatles. Claudia has

dotted several of such similar ending properties, e.g. dbp-

data.org/ontology/writer and dbpedia.org/property/writer.

However, unlike for properties ending with label, the links

ending with writer seem to have identical meanings as they

all seem to link to people who have written songs of The

Beatles.

Furthermore, Claudia experiences links leading to state-

ments with few further outcoming links, such as http://

www.w3.org/2002/07/owl#Class, or even to dead ends,

thus stopping further navigation and requiring her to go back

and forth several times. The following list enumerates the

shortcomings experienced by Claudia:

1. The magnitude of information makes the identification

and quantification of relevant links difficult.

2. The one-step browsing approach implemented by current

browsers limits the exploration of the sophisticated inter-

relations in the web of linked data.

3. Redundant and ambiguous links reduce the effectiveness

and efficiency of browsing.

4. Dead-ends in LOD interrupt the browsing process as they

require users to browse backwards manually in order to

find further relevant information.

These shortcomings of browsing LOD are related to the fact

that none of the current LOD browsers provide sophisticated

functionalities to support and guide users to explore inter-

relations in the web of LOD, such as, to present a list of

relevant resources for a resource, which are not necessarily

connected by a single link-hop, or to present the relevant

resources organized by different aspects.

LENA

TR

LENA

TR

is a linked data browser that features faceted naviga-

tion (cf. Fig. 2) to overcome the shortcomings listed in the

section above. A facet is a named group of resources where

resources of a group have a similar – not necessarily exactly

the same – relation to the resource currently displayed by

LENA

TR

. For instance, Figure 2 shows the facets label,

associatedActs, currentMembers and so on for the resource

The Beatles.

By moving the mouse over the name of a facet, the pre-

dicates contributing to a facet are shown as tooltip. Figure 2

shows such a tooltip for the facet associatedActs. It lists the

complete URIs of the contributing properties and the score

used for ranking them. In this example the top-three pre-

dicates are associatedActs, associatedBand, associatedMus-

icalArtist, all with a score of more than 0.4. These descrip-

tions of facets enable users to investigate on the knowledge-

aspect expressed by a facet.

The facets themselves are ordered according to their rel-

evance with respect to the resource currently viewed in

LENA

TR

. The relevance of a facet is expressed by the weight

that is displayed below the name of each facet, e.g. Figure 2

highlights the weight of the facet currentMembers.

Each facet lists several resources. The list is ordered so

that the most relevant resources are at the top of the list. This

ordering is based on scores that can also be investigated by

users. The tooltip for resources of a facet shows the com-

plete URI of the resource and its score. Figure 2 shows such

a tooltip for the resource Paul McCartney that is the top re-

source for the facet currentMembers.

The facets provided by LENA

TR

build upon rankings for

resources and properties in the vicinity of the resource cur-

rently displayed by LENA

TR

. These rankings are computed

at runtime whenever a new resource is browsed. As rank-

ning method, we integrated the TripleRank approach. Its out-

puts are exploited by the user interface features presented.

To highlight how these features of LENA

TR

help to over-

come the shortcomings in LOD browsing identified above,

we confront LENA

TR

with each shortcoming in the follow-

ing.

Shortcoming 1: The magnitude of information makes the

identification and quantification of relevant links dif-

ficult. As illustrated in Figure 1, LOD browsing can easily

lead to information overload. LENA

TR

implements faceted

navigation as a counter measure to information overload.

Facets assist users in browsing LOD by guiding them along

most promising knowledge aspects, giving them a means
to explore the correlations between resources and available

LOD. They enable to identify relevant resources and proper-

ties more easily than by scanning manually through all of

the available information and thus mitigate the negative effect

of feeling overloaded.

Shortcoming 2: The one-step approach limits exploring the

sophisticated interrelations of the web of data. By

combining LOD browsing and faceted navigation, a user

is not just able to browse LOD along RDF links based on

the one-step navigation approach but also to browse along

facets. This combination enables users to go beyond explicit

links by exploring along facets. As explained before, a facet

corresponds to a set of similar types of links and thus repres-

ents an implicit link. Moreover, resources shown within the

facets offered by LENA

TR

are not necessarily connected by

a single link-hop. They are suggested based on the TripleR-

ank analysis that considers resources in the vicinity of the re-

source currently viewed. Accordingly, selecting a resource of

a facet in LENA

TR

can correspond to multiple browsing

steps.

Shortcoming 3: Redundant and ambiguous links. The

scenario has revealed the existence of ambiguous and re-

dundant links as a cause for dissatisfaction and irritation. Facets

provided by LENA

TR

summarize multiple types of links,

e.g. as shown by Figure 2. They also provide a simple means to

distinguish among groups of links. Thus, they effectively
hide redundancy and ambiguity of the underlying data and thus mitigate irritations caused by it.

**Shortcoming 4: Dead-ends in LOD interrupt the browsing process.** The web of LOD connects well defined datasets to a huge cloud of interlinked data. However, many resources have only few outgoing links and can be considered as dead ends that interrupt the browsing process. By faceted LOD browsing as implemented by LENA\textsuperscript{TR}, users are always enabled to navigate to resources that are not directly connected by a single-link-hop. Consequently, the browsing process does not end because no links can be followed anymore.

**Architecture**

LENA\textsuperscript{TR} runs as Java servlet application accessible from common web browsers. It utilizes an implementation of the TripleRank method (Franz et al. 2009) in order to support LOD faceted browsing. In the following, we describe the work-process of LENA\textsuperscript{TR} and review its architecture (see Fig. 3).

LENA\textsuperscript{TR} implements the Model-View-Controller (MVC) pattern. The controller manages incoming requests from the browser, queries data from the model, and serves the view with data for the final rendering output. The model allows for requesting LOD, e.g. by means of SPARQL queries. It serves a triple cache of local repositories to reduce network overhead. The view consists of XHTML templates used to generate the final rendering output with help of the provided controller information. Since LENA\textsuperscript{TR} relies on the Fresnel display vocabulary for rendering requested datasets, the controller utilizes the JFresnel Java library in order to apply Fresnel (Pietriga et al. 2006) definitions.
Upon incoming HTTP requests from web browsers, the requested RDF resource is rendered according to the standard LENA rendering process (Koch, Franz, and Staab 2008). In parallel, the TripleRank components are invoked and requested by means of an AJAX call. Thus, the user can immediately inspect the resource, while assisting facets are provided as soon as the TripleRank processing is complete. The three TripleRank components (crawler, pre-processor, analyser) are processed consecutively, whereas the resulting facets are injected into the user interface by means of AJAX.

**TripleRank**

The TripleRank implementation is separated into three core components; namely, the crawler, pre-processor and analysis component (cf. Fig. 3). The crawler component performs a breadth-first search. It emanates from the requested resource and recursively aggregates related LOD up to a certain depth, link-, and statement limit. Technically, RDF data is requested from LOD data providers on the semantic web via SPARQL endpoints. The output of the crawling-process is a RDF graph (e.g. as shown by Figure 4(a)), which is transformed into a tensor representation (cf. Figure 4(b)), that is input for the pre-processor.

The pre-processor component is responsible for i) reducing the amount of the crawled data and for ii) increasing its quality. E.g. properties linking a majority of resources are discarded, since they only marginally contribute to the overall information. Secondly, statements are strengthened according to their property frequency, whereas less frequent properties are weighted stronger, since their significance appears to be more eminent (for details see (Franz et al. 2009)).

The TripleRank analysis computes a list of weighted facets for a given RDF graph. Each facet is described by multiple weighted RDF properties that contribute to it. As illustrated by Figure 2, the user interface of LENA<sup>TR</sup> enables to view the contributing properties and their ranking scores. Additionally, TripleRank also returns for each facet, a ranked list of RDF resources that are displayed by LENA<sup>TR</sup> to ease the browsing experience as discussed above. Table 4(c) illustrates the results produced by TripleRank for the RDF graph shown in Figure 4(a).

**Tensor Model**

By means of the crawler and pre-processing components, an RDF graph is created that is transformed to a tensor representation. Figure 4(b) illustrates the tensor representation resulting from the transformation of the sample graph shown in Fig. 4(a). The first adjacency matrix \( T(:,1) \) models linkage by the property loves. An entry > 0 corresponds to the existence of a link by this property, empty entries are considered as zeroes. The second matrix \( T(:,2) \) models links by the property hates. For instance, the graph expresses that Alex hates Bob, which corresponds to \( T(1,2,2) = 1 \).

---

<sup>6</sup>Throughout this paper we use the common Matlab-notation for addressing entries in tensors and vectors

**PARAFAC Analysis**

As analysis step, a decomposition of the tensor is performed to detect hidden dependencies in the data. The TripleRank implementation applies the PARAFAC method for this purpose. It is regarded as a higher-order equivalent to a matrix decomposition like the singular value decomposition (SVD). The PARAFAC tensor decomposition has the advantage of robustness and computational efficiency. These advantages are due to its uniqueness property with respect to the scaling and permutation of produced component matrices (Harshman and Lundy 1994). The analysis step transforms the input tensor into a Kruskal tensor that corresponds to authority and hub scores (Kleimberg 1999) for particular latent aspects (topics) of the analyzed data.

Formally, a tensor \( T \in \mathbb{R}^{k \times l \times m} \) is decomposed by an n-Rank-PARAFAC into component matrices \( U_1 \in \mathbb{R}^{k \times n} \), \( U_2 \in \mathbb{R}^{l \times n} \), \( U_3 \in \mathbb{R}^{m \times n} \) and \( n \) principal factors \((p,f)\) \( \lambda_k \) in descending order. Via these, \( T \) can be written as a Kruskal tensor by \( T \approx \sum_{k=1}^{n} \lambda_k \cdot U_1^k \odot U_2^k \odot U_3^k \) where \( \lambda_k \) denotes the \( k \)-th principal factor, \( U_1^k \) the \( k \)-th column of \( U_1 \) and \( \odot \) the outer product (Kolda and Bader 2009). \( U_1 \) yields the ratio of the \( i \)-th dimension to the principal factors. So, similar to SVD, PARAFAC derives hidden dependencies related to the principal factors and expresses the dimensions of the tensor by relations to the principal factors. Depending on the number of the principal factors, the PARAFAC decomposition can be loss-free. For a third-mode-tensor \( T \in \mathbb{R}^{k \times l \times m} \) a weak upper bound for this rank is known: \( \text{rank}(T) \leq \min\{kl,lm,km\} \) (Kolda and Bader 2009). While there is no proper way for estimating the optimal number of principal factors for an appropriate decomposition, several indicators like residue analysis and core consistency analysis exist for its estimation (Andersson and Bro 2000).

The output of a PARAFAC decomposition can be interpreted as authority and hub scores plus additional scores for the relevance of link types, i.e. RDF properties. Table 4(c) shows the scores of RDF properties and authority scores for resources. At the time of this writing, hub scores are not utilized by LENA<sup>TR</sup>.

**Related Work**

A few research works also deal with the challenge of facilitating users with more effective, efficient, and satisfying browsing support for linked open data. The Fresnel framework (Pietriga et al. 2006) consists of a vocabulary to express how certain information shall be presented, what the ordering for displayed information is and which information to leave out. Interpreters are provided that select, filter, and present data according to such specifications. While this approach requires to encode statically how information is displayed, with LENA<sup>TR</sup> we pursue an approach to identify the best links and resources dynamically.

Unlike LENA<sup>TR</sup>, Parallax (Huynh and Karger 2009) is a browser for RDF data that enables users to formulate join conditions and type restrictions in an intuitive manner that does not require users to be acquainted with a query language. The Tabulator browser (Berners-Lee et al. 2007)
also implements support for the formulation of queries on LOD. Queries are triggered on available LOD by selecting predicates and thereby creating a query pattern. The results of those queries can be visualized with respect to different information aspects, e.g. a map view that indicates spatial relations. Sig.ma (Tummarello et al. 2009) is a search engine for the Semantic Web that offers versatile utilizations of search results including their selection and filtering (e.g. by data source), export to several formats, and different display options. Unlike LENA TR R, sig.ma does not provide a ranking for the data retrieved. The Marbles browser also implements a means to ease the exploration and sense-making of LOD by indicating the provenance of resources. The metaphor of marbles is employed to distinguish between different sources of information. The Openlink Dataexplorer7 supports the filtering and ordering of information. Furthermore, various domain-specific views are provided, e.g. for data described by the FOAF vocabulary.

Conclusion and Outlook

This paper has presented a faceted navigation approach that integrates a novel ranking method with linked data principles. Its implementation by LENA TR R enables users to browse along hidden and diverse knowledge aspects and has been addressed in several ways: First, a user interface for faceted navigation has been presented. Second, the exploitation of a novel method for relevance ranking of linked open data, namely TripleRank has been illustrated. Third, the integration of this method into an architecture that facilitates the user interface has been described.

We are currently investigating on several optimizations of the components of LENA TR R, focusing particularly on the components for crawling and caching the LOD used for the analysis by TripleRank. We also aim for an extensive evaluation that considers implicit feedback gathered by users of LENA TR R.

Acknowledgements This research was supported by the European Commission under contract IST-FP6-026978, X-Media. The expressed content is the view of the authors but not necessarily the view of the X-Media project.

7http://demo.openlinksw.com/ode/

References


