Evaluating Semantic Browsers for Consuming Linked Data

Fahad Alahmari[1] **James A. Thom**[1] **Liam Magee**[2] **Wilson Wong**[1]

[1] School of Computer Science and Information Technology
[2] School of Global Studies, Social Science and Planning
RMIT University
Melbourne, Australia
Emails: `firstname.lastname@rmit.edu.au`

Abstract

The value of a single dataset is increased when it is linked to combinations of datasets to provide users with more information. Linked Data is a style of publishing data on the Web by using a structured machine-readable format, RDF, and semantically typed relations to connect related data. Its structured representation opens up new possibilities in the way these data can be accessed and queried, while posing new design challenges for human interactions such as overloading data, navigation style, or browsing mechanism. In this paper, we review 14 semantic browsers available for the consumption of structured Linked Data and evaluate them against our five criteria framework in order to establish how well these browsers bring the benefit of Linked Data to human users.

Keywords: Linked Data, Semantic Web, Semantic browsers, RDF browsers, Semantic interface, User interaction

1 Introduction

The Semantic Web (SW) has been famously defined as a "web of data that can be processed directly and indirectly by machines" [13]. Just as the value of World Wide Web *documents* increases dramatically when they are linked to other documents, the SW enhances the value of *data* by providing standardised mechanisms for describing and linking them to other datasets. The emergence of Linked Data (LD) sets using Uniform Resource Identifiers (URIs) and encoded in the Resource Description Framework (RDF) results in a powerful network of machine-processable information, making way for the next generation of the Web [3]. Distributed on a global scale, LD in turn can be used by machines to generate new information and knowledge.

Resources on the Web of LD are identified using Uniform Resource Identifiers (URIs), which are used to specify a globally unique name for a resource. Resources can be further distinguished between resources that describe real-world objects such as people, places and cars, and resources that describe other resources, including documents, images and video. In the past decade, many large providers of data, including the US (data.gov[1]) and UK (data.gov.uk[2]) governments have adopted LD for publishing their data. As evidence of the growth of LD [5], the Linked Open Data (LOD) community project (Figure 1) has grown to 256 datasets, which include nearly 30 billion triples interlinked by 471 million links (as of August 2011[3]). The LOD cloud contains data from a range of different domains, including media, government, the life sciences and geography. Significantly, these datasets are linked together by terms from the W3C base vocabularies — RDF, RDF Schema and OWL — in order to make LD machine processable. As a further sign that LD is moving towards "mainstream" adoption, Google, Yahoo and Microsoft have recently agreed to share a collection of schemas `http://schema.org/` that can be used to publish structured and LD.

While LD is designed for machine processing, human agents ultimately need to be able to navigate and query datasets. With the rapid growth of LD, users require powerful tools to browse and explore particular datasets of interest. However, there are considerable challenges for designing usable browsing and visualisation tools to explore the Web of LD as the requirements for browsing data are very different from the requirements for browsing documents. These challenges can be summarised as follows:

1. Exploratory challenges: the Web of LD connects a huge range of related data that includes both real-world and other web resources. How can a browser best present this wide range of data to the end users in a meaningful way?

2. Navigational challenges: the navigation techniques for LD differ from the navigation techniques in the Web of documents. In the Web of documents, the traditional browsers use untyped links to navigate between webpages, while the semantic browsers use typed links to navigate between entities. For instance, *foaf:knows*, *foaf:homepage* and *foaf:gender* are terms for linking people on the Web. How can a browser provide convenient mechanisms to allow for conventional forward and backward navigational techniques to users, while also making use of the linking context?

3. Interactive challenges: human users frequently need to interact and manipulate datasets in a number of ways. In the LD world, this is made more difficult due to the different languages used. While the formal languages of the SW — OWL, RDFS, RDF, RDFa, GRDDL and SPARQL —

[1] `http://data.gov`
[2] `http://data.gov.uk`
[3] `http://www4.wiwiss.fu-berlin.de/lodcloud/state/`

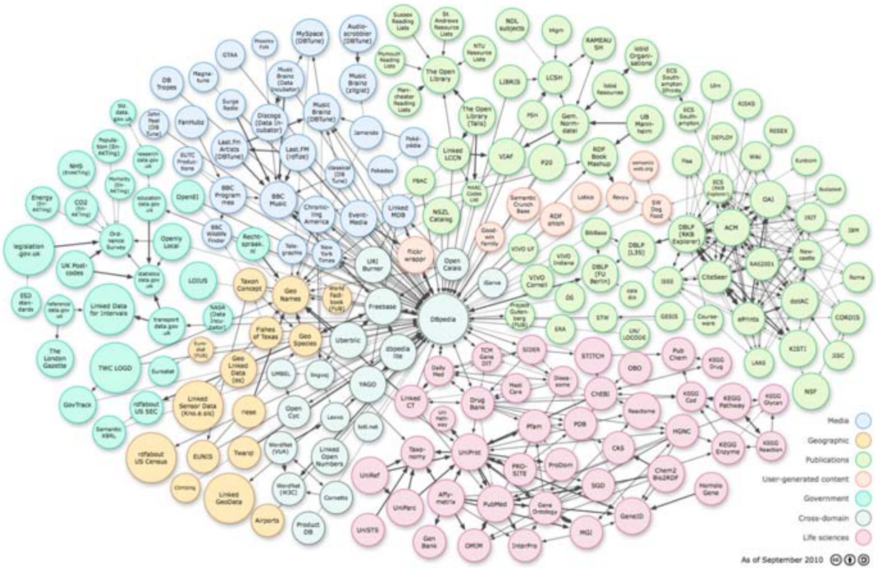

Figure 1: Linking Open Data cloud diagram, by Richard Cyganiak and Anja Jentzsch. http://lod-cloud.net/

are useful for defining resources for machine processing, they are often challenging to work with and understand, especially for human agents who are more familiar with document and scripting languages like HTML5, XHTML, CSS3 and JavaScript. How can a browser permit users to interact and manipulate LD?

The challenges above show that there are substantial gaps in bringing the benefit of LD to human users. Considerable research has been devoted to the design and development of "semantic browsers" that aim to overcome these gaps by providing mechanisms for navigating and visualising LD sets. While Dadzie and Rowe [8] provide a comprehensive review of current state-of-the-art semantic browsing tools, their emphasis is on the first two of the three challenges identified above. Here we present a further review, closely following their methodology, but with greater attention to the third of these challenges. Specifically, we examine the facilities and features that SW browsers provide for incorporating new datasets as well as interacting with existing LD sets.

The main contributions of this study are as follows:

1. We propose five criteria for evaluating semantic browsers.

2. We briefly review 14 semantic browsers that are currently available for the consumption and manipulation of LD.

3. We evaluate the semantic browsers according to our five criteria in order to determine the ones that are best suited for LD manipulation. Again following Dadzie and Rowe [8], we include in our evaluation a discussion of the pros and cons of the browsers for interacting with LD.

Section 2 of the paper gives overview on recent related work, including a brief review of semantic browsers. Section 3 proposes our five criteria and methodology for the browser evaluation study presented in Section 4. Section 5 provides a discussion and the conclusion.

2 Related Work

In recent years, there has been an increasing amount of literature on LD. While building upon the foundations of the SW architecture, LD focuses more specifically on the linking and usage of datasets. Recent studies point to considerable research attention on publishing LD [4], using LD [9], searching LD [19], sharing LD across a community [6] and browsing LD [8].

In 2006, Berners-Lee articulated several key rules for publishing data on the Web in order to expose it as LD [3]. These rules are:

1. Use of URIs as names for things. On the SW, URIs indicate entities such as people or places, in addition to content such as text or images.

2. Use of hypertext transfer protocol (HTTP) for URIs, as the key protocol for agents to resolve URIs.

3. Use of RDF and SPARQL [4], for representing and querying SW data.

4. Links to other URIs must be embedded within RDF datasets, in order to discover additional information.

In 2010, Berners-Lee has also developed a 5-star rate scheme, to encourage data owners in various domain areas, such as government, healthcare and multimedia, to expose their datasets as part of the Linked Open Data (LOD) cloud. According to [11], the 5-star system is as follows:

1 Star: "data is available on the Web (whatever format), but with an open license"

2 Stars: "data is available as machine-readable structured data (e.g., Microsoft Excel instead of a scanned image of a table)"

3 Stars: "data is available as (2) but in a non-proprietary format (e.g., CSV instead of Excel)"

[4] http://www.w3.org/TR/rdf-sparql-query

4 Stars: "data is available according to all the above, plus the use of open standards from the W3C (RDF and SPARQL) to identify things, so that people can link to it"

5 Stars: "data is available according to all the above, plus outgoing links to other peoples data to provide context"

Since this scheme has important implications for how data is subsequently manipulated, we adopt it here as one of our evaluation criteria for semantic browsers, as discussed further below.

Dadzie and Rowe [8] recently carried out a survey on current approaches to visualising LD. They classified current semantic browsers into the following two overall types:

Text-based Browsers: these browsers use textual structures such as tables and lists to present LD entities, properties and relationships. Some also use advanced features such as faceted browsing to allow for more intuitive rendering and navigation of data. Examples of such browsers include: Dipper[5], Disco[6], Marbles[7][2], Piggy Bank[8][12], Sig.ma[9][18], URIburner[10] and Zitgist[11].

Browsers with visualisation options: these browsers use primarily visual or graphic structures, such as images, maps, graphs and timelines (individually and in combinations) to represent LD. Examples of such browsers include: DBpedia Mobile[12][2, 7], IsaViz[13], OpenLink Data Explorer (ODE)[14], RDF Gravity[15], TheRelationshipFinder[16][15, 14] and the Tabulator[17][17].

In addition to using 14 of the 15 browsers they reviewed, we also adopt this distinction between text and graphical browsers.

Dadzie and Rowe [8] also define the types of LD users and their different requirements for the consumption and production of LD. Their analysis follows Shneiderman [16], who classifies users into three types:

Lay users with little or no understanding of the underlying semantic technologies. Such users might use semantic browsers for exploring large data sets or finding particular facts of general interest (on DBpedia for example),

Technical users with expertise in the SW and LD. Such users might use semantic browsers for data retrieval, integration and analysis (so-called "mash-ups"), using advanced filtering and querying services, and

Domain experts with expertise in a specific domain, but who may not be familiar with particular SW and LD technologies. Such users, such as medical researchers, might use semantic browsers for advanced domain-specific queries and ontology reasoning.

[5] http://api.talis.com/stores/iand-dev1/items/dipper.html
[6] http://www4.wiwiss.fu-berlin.de/bizer/ng4j/disco/
[7] http://www5.wiwiss.fu-berlin.de/marbles
[8] http://simile.mit.edu/wiki/Piggy_Bank
[9] http://sig.ma/
[10] http://linkeddata.uriburner.com/fct
[11] http://zitgist.com/products/zlinks/zlinks.html
[12] http://wiki.Dbpedia.org/DbpediaMobile
[13] http://www.w3.org/2001/11/IsaViz/
[14] http://ode.openlinksw.com/
[15] http://semweb.salzburgresearch.at/apps/rdf-gravity/
[16] http://www.visualdataweb.org/relfinder.php
[17] http://www.w3.org/2005/ajar/tab

3 Criteria Framework and Methodology

We first propose five criteria to evaluate the facilities of the browsers surveyed by Dadzie and Rowe [8], with a specific emphasis on their data navigation, triage and manipulation capabilities. We then describe our methodology for evaluating these browsers using these criteria.

3.1 Evaluation criteria

Our proposed criteria for evaluating browsers from a LD perspective are as follows:

1. **Data conversion:** How easy is it to convert non-LD into LD using the browser?

2. **Creating links to other URIs:** How easy is it for a user to find new LD and add their own links to data? How easy is it for a user to navigate from a current dataset to a new dataset? How easy is it to navigate forward and backward using the browser? Are the navigation links generic or specific ("context-aware" in some sense)?

3. **Data triage:** What tools does the browser have to answer queries or questions? How effective is the tool (if any) to sort data based on the users' needs? Does the query facility support text-search or SPARQL queries or both?

4. **Browsing mechanism:** What is the browsing mechanism? Does it support faceted, pivoting and "link-sliding" browsing mechanisms?

5. **5-star data:** Berners-Lee suggested a 5-star deployment scheme for Linked Open Data[18]. We consider what level of support is offered for data consumption according to Berners-Lee's scheme. If the browser also supports the production of data, where data is actively aggregated, augmented or reinterpreted by the browser, what is the level of this support?

Common formats for data include: Excel, CSV, XML, relational database files and RDF. A key aspect of working with structured data is understanding and manipulating the formats they may be represented in. Further, data schemas can vary from one dataset to another. Hence we introduce the **Data Conversion** criterion to evaluate how well semantic browsers handle conversions between different formats.

Once data is in an appropriate representation, it needs to be linked to other data. For example, when a user wants to look up a URI, this URI has to have some links to other URIs to provide more information about things. Users also often want to be able to create links of their own between heterogeneous datasets. We include the criterion of **Creating links to other URIs** to describe how well browsers facilitate both navigation of existing links, and the creation of new links between datasets.

We also consider facilities for retrieving, manipulating and reorienting LD sets under the criterion of **Data triage** and **Browsing mechanism**. Here we evaluate browsers support for "advanced" features such as SPARQL queries, sorting, faceted and pivoted views, and any other facilities for visualising datasets.

Finally, we include Berners-Lee's 5-star scheme, for considering how well browsers support the consumption and production of LD.

[18] http://inkdroid.org/journal/2010/06/04/the-5-stars-of-open-linked-data/

While our evaluation follows that of Dadzie and Rowe [8], our criteria focus more directly on the levels of user interactivity with LD. Hence we evaluate features of greater relevance to technical users and domain experts.

3.2 Methodology

To enable the evaluation of these browsers, we installed all the browsers currently available for testing on our machine as a first step. In some cases, where the "browser" is a web service, this simply involved visiting the relevant URL. In other cases the process involved the installation of a browser extension or desktop application. We performed some initial tests to make sure all the browsers functioned correctly. There is that could not be installed, or returned critical errors, were excluded from our criteria-based evaluation, as noted below.

The tests also allowed us to familiarise ourselves with how these browsers work and how they can be used, as there are different ways to enable the use of these browsers to explore SW data. During this process, we noted three methods of operation: some worked as independent applications which can be downloaded (IsaViz and RDF Gravity), while some operate as browser extensions (ODE, Zitgist, Marbles, Disco and Tabulator). Another group run within the browser itself (Dipper, Piggy Bank, URIburner and Sig.ma). This in turn has implications for levels of interactivity, as we describe further below.

In this study, two types of evaluation were conducted to provide different methods for examining approaches of existing browsers. The first type of evaluation is a general review of the 14 browsers, to describe our experience and impression of the usability and capability of these browsers from a technical user's perspective. The second evaluated the browsers according to our five criteria. Both types of evaluation assist in identifying the advantages and disadvantages of the existing browsers for interacting with LD by technical users.

To evaluate the first criterion, "Data conversion", we used the following three steps to test the conversion process:

1. Find out whether the browser provides a service of converting any of these types: CSV, RDF, HTML or URL into LD style.
2. If so, identify the conversion procedures and whether the process happened on-the-fly or by the user.
3. Then, determine which kind of users can perform the conversion process (lay users or technical users).

The above steps allow us to undertake standard evaluation of the conversion process for non-LD into LD for all semantic browsers.

For evaluating the second criterion, "Creating links to other URIs", we followed these steps:

1. Start from a URI, which is maintained according to W3C standard for SW URIs 'use so-called "Cool URIs" [19]'.
2. Once the results are retrieved, we look at the URIs to check whether they "Cool URIs" or not; if so, we use the "click" action to navigate to another URI that can fetch up a different datasets.
3. We repeat step 2 for three more times until it completed five datasets; with each URI, we check whether the URI provides useful information for the user.

For example, we use the URI "http://Dbpedia.org/resource/Berlin" to explore data about Berlin starting from the DBpedia dataset (a LD version of Wikipedia) and then navigate to the other datasets using the URIs that appear in the results.

The third and fourth criteria are evaluated by observing the browsers features that are related to both. In criterion 3, if a browser provides support for a SPARQL endpoint, we run an SPO ("Subject Predicate Object") SPARQL query and test for the endpoints. The text search provided by some browsers are also evaluated by typing a simple query. In criterion 4, the evaluation was conducted by observing the mechanism view of browsers.

The fifth criterion evaluates the level of data support within the browsers. Since all the browsers are semantic-based, they all support data in RDF format. However, for those that did not directly support LD facilities, they were scored 4 rather than 5 stars. A 5 star rating would be applied only when data is linked to another URIs. In addition, some browsers actively convert RDF data, producing integrated views of more than one data set. We distinguish this capability in terms of "producing" rather than simply "consuming" LD, and also rated this feature accordingly. Our method to evaluate links for this criterion extended the approach in criterion 2 by testing links between multiple datasets to identify whether the data are discoverable or not. The test included both inward and outward bound links.

4 Evaluation Study

Our evaluation is in two parts. First we review the 14 semantic browsers to highlight their general usability, and specific functionality that supports user interaction. We then conduct the evaluation against our five criteria.

4.1 Review of SW browsers from a technical perspective

We follow the classification structure as well as the actual browsers analysed by Dadzie and Rowe [8], as described in Section 2. We have, however, excluded one browser, namely, Zitgist that seem to be a defunct project at present. Here we focus on the level of user interaction supported, in particular on the functions provided for technical users to explore heterogeneous data.

4.1.1 Text-based browsers

Text-based browsers rely predominantly upon textual representation to present LD resources.

Dipper is a text browser that allows for the exploration of RDF data stores. In our initial review, the aims of Dipper appears ambiguous; supporting documentation is sparse, and does not describe the way the software works, or even what it was designed for. Dipper's user interface is not particularly intuitive, as it is hard for even advanced users to determine where to source and consume data they need. Dipper does however provide links to some of the public RDF stores, such as Openlibrary, NASA,

[19] http://www.w3.org/TR/cooluris/

BBC-Backstage and data.gov.uk. Most datasets are stored on repositories provided by the developer of the browser, Talis[20].

We observed that there did not seem to be an organising principle on how datasets were collected together. For instance, the only airport we found in the airport dataset is Birmingham International Airport. The data provider has not given any explanation as to why these datasets were chosen. This means queries provide ad hoc and partial results, without obvious means for loading further datasets and extending these results. In addition, during testing we faced error messages such as "Could not retrieve data: error", without providing any explanation on why that error has occurred.

In terms of LD, the browser does not provide a mechanism to go beyond the current data store to find other data. Navigational control is limited, without backward and forward options, and data retrieval also seems slow.

Disco is described as a simple browser for navigating the SW as an unbound set of data sources. It renders all information it finds on a particular resource as HTML. It allows navigation between SW resources by dereferencing HTTP URIs and by following *rdfs:seeAlso* links.

Marbles is an application that resides on the server formatting SW resources for HTML clients, such as HTML browsers, through using Fresnel lenses and formats[21]. Its user interface is sparse, which in certain contexts may be a strength as this would not overwhelm an average user exploring the SW. When provided with a URI to display, it tries to dereference it, as well as querying Sindice and Falcons for a data source that may contain information about that resource. It also uses *owl:sameAs* and *rdfs:seeAlso* to retrieve more data about the resource in question. For example, providing a URI such as `http://openlibrary.org/b/OL649M` to display, it retrieves and format the resource for suitable HTML viewing.

Piggy Bank facilitates users to convert standard HTML Web content into SW content. It was developed earlier on during the advent of the SW to provide a way to easily obtain SW content from Web content that users might encounter in regular browsing. It uses a series of customised screenscrapers to turn HTML Web content into RDF. Although this approach has merit, it clearly requires some development effort to build a custom screenscraper in order to recognise the particular characteristics of the website being analysed.

Sig.ma is an application and a browser that integrates LD from multiple sources allowing data navigation (see Figure 2). The initial interaction is driven by the user with a free text search, which is a useful way to begin data exploration especially when compared with the other browsers which prompt for a URI to begin data exploration. The ability to use a URI as an entry point is also still available for users. Sig.ma is built on top of Sindice, a semantic search engine, that provides a search service allowing technical users to find resource descriptions [19].

[20] `http://www.talis.com/`

[21] Fresnel `http://www.w3.org/2005/04/fresnel-info/` is a browser-independent vocabulary for specifying how RDF graphs are presented.

Figure 2: Sig.ma text-based browser highlights data sources with associated facts

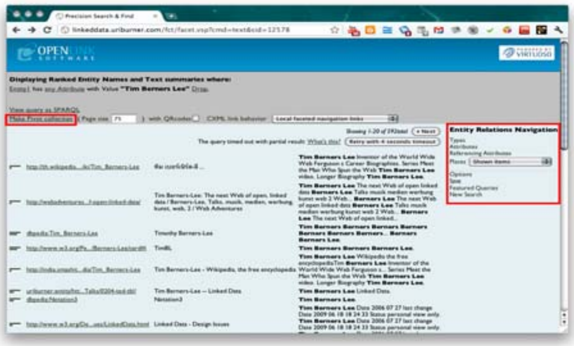

Figure 3: URIburner, a text-based browser that uses the canonical RDF form of subject-predicate-object to present information

URIburner is a software service that delivers structured data about Web resources, generating an RDF graph of the resource's metadata using existing well-known ontologies as well as site-specific knowledge. It takes a Web URI, and can represent the dereferenced resource as HTML or RDF. It also provides a Firefox extension which can be used to bookmark the URI of interest.

4.1.2 Browsers with visualisation options

In this section, we discuss browsers and applications that use graphic visualisations to represent SW resources.

DBpedia Mobile is a client application, designed for mobile phone use, that allows users to access information about objects stored in DBpedia. For example, real-world entities such as cities, streets and landmarks that have been described and asserted in the DBpedia knowledge base can be queried automatically based on a user's current geolocation. DBpedia augments these resource descriptions so that users can explore other resources on the SW. DBpedia Mobile uses Marbles to render Fresnel-based views of these Web resources that have information in DBpedia. Users can moreover use these data as "jumping-off" points to other SW datastores such as Geonames, Eurostast and Revyu.

Fenfire is a service that enables users to explore LD by dereferencing URIs, and following available *rdfs:seeAlso* properties to retrieve related

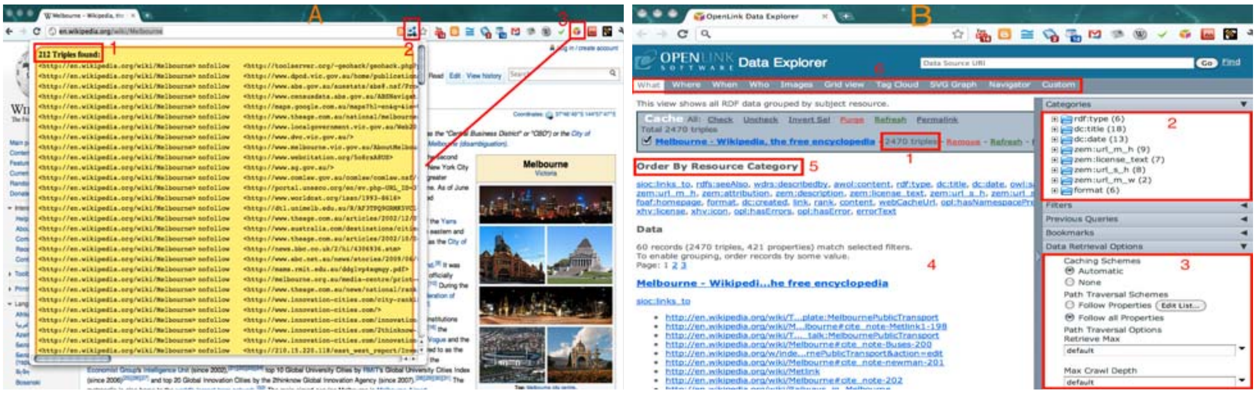

Figure 4: Example of converting Wikipedia page to Linked Data (high resolution on the electronic version)

datasets [10]. The user can enter the starting URI to begin navigating LD.

IsaViz is a visual application for browsing and editing RDFs as graphical objects. The software helps to see the graph structure of the RDF. Clicking on a vertex node of the RDF graph brings up the textual descriptions of the vertex. Users can further edit names of RDF nodes, as well as add and delete both nodes and links between nodes.

LESS enables users to create their own templates to view LD [1]. It uses the LESS Template Language (LeTL) which define a text-representation to view the output from URI dereferencing or SPARQL queries.

OpenLink Data Explorer (ODE) is a browser extension that allows Web users to explore raw data and relationships of a Web page. Once installed, when a user is browsing a Web page which they would like to obtain LD for, they simply click on the ODE extension, and that Web page is turned into a suitable LD format viewable in a browser.

RDF Gravity is another application for visualising RDFs as graphs and the ontologies. It allows a user to customise the view of results by using local filters.

TheRelationshipFinder (Relfinder) is a Web application for exploring connections between objects in a SW knowledge base. It offers a new way to get insights on how instance data in a SW knowledge base relate together. The knowledge base used in RelFinder is the DBpedia dataset.

Tabulator is an RDF browser, designed both for users and developers of RDF content. A key feature is it allows RDF data providers see how their data interacts with the rest of the SW. The browser is designed to be as easy as possible for a new user to pick up, and easy for developers to extend with their own ideas. Our experience of trying Tabulator came up with a lot of networking errors due to some technical limitations in the tool. None of the available versions worked particularly well, posing difficulties for more detailed evaluation. This evaluation is assisted therefore from supplementary resources: a movie demonstrating the browser (`http://dig.csail.mit.edu/2007/tab/tutorial/editing.mov`) and an associated paper [17].

4.2 Evaluating browsers using our criteria

In this section, 14 semantic browsers are evaluated according to our five criteria described in Section 3.

4.2.1 Data conversion

Some browsers, such as Dipper and Sig.ma, do not support any data conversion capabilities. Dipper assumes that the resource in question is already in LD format. Sig.ma handles a mixture of LD and non-LD but does not facilitate the conversion of non-LD to LD. The Marbles browser supports data conversion of HTML; users can view any Web page as LD through the click of a button. Piggy Bank has similarly impressive function that uses a screenscraping approach to turn the HTML content into RDF content. However, the available screenscraping is limited to a number of websites such as the ACM Portal Scraper and the Flickr Photo Scraper. Additional screenscrapers need to be developed for other websites in Javascript or through use of an associated tool, Solvent, based on an awareness of the underlying document structure of the targeted website.

URIburner has extended facilities that make it fairly trivial to convert non-LD into LD, so long as the original sources are interpreted via an associated Virtuoso middleware platform. Together with Virtuoso, URIburner provides the following URL patterns for browsing the description of URIs:

HTML: `http://linkeddata.uriburner.com/about/html/[URLscheme]/[hostname]/[localpart]`

RDF: `http://linkeddata.uriburner.com/about/rdf/[URLscheme]/[hostname]/[localpart]`

where rdf can be replaced with any other format such as: xml, n3, nt or ttl.

For example: `http://linkeddata.uriburner.com/about/rdf/http://zeus.seg.rmit.edu.au:8080/diseasome/page/resource/diseases/100` provides LD of `http://zeus.seg.rmit.edu.au:8080/diseasome/resource/diseases/100`.

Although DBpedia Mobile uses some of the capabilities of Marbles to explore data, it does not provide a direct facility for data conversion. The client browser starts the exploration from DBpedia. IsaViz, however, does not provide this feature for users; it is only useful for browsing LD that are already in RDF.

In ODE, a browser extension[22], it is easy to convert any HTML page into LD. Figure 4 shows an

[22] `http://ode.openlinksw.com/`

Table 1: Evaluation of textual LD browsers from an interactivity perspective

		Dipper	**Disco**	**Marbles**	**Piggy Bank**	**Sig.ma**	**URIburner**
Data conversion	Unstructured data to RDF		✓		✓		✓
	Structured data to LD			✓		✓	✓
	Export RDF/XML, JSON format	✓				✓	✓
Creating links to other URIs	Forward navigation		✓	✓	✓		✓
	Backward navigation		✓	✓	✓		✓
	RDF model navigation					✓	
	Links filtering					✓	✓
	Navigating Local LD	✓		✓	✓	✓	✓
	Navigating Global LD			✓		✓	✓
Data triage	Simple (keyword search or filter)				✓	✓	✓
	Complex (SPARQL)	✓		✓			✓
	Use of search engine			✓		✓	
	Caching results		✓			✓	✓
	Aggregated results			✓		✓	
	Editable results					✓	✓
	Tagging results				✓	✓	
Browsing mechanism	Entity-Attributes-value view	✓	✓	✓	✓	✓	✓
	HTML view	✓	✓	✓		✓	✓
	Facet view				✓	✓	
	Map view						
	Visual view					✓	
	Grid view						✓
	Layout control					✓	
	“Order by” option					✓	✓
5-star data schema	Producer 5 stars					5	5
	Consumer 5 stars	5	4	5	4	5	5

Table 2: Evaluation of visual LD browsers from an interactivity perspective

		DBpedia Mobile	**Fenfire**	**IsaViz**	**LESS**	**ODE**	**RDF Gravity**	**RelFinder**	**Tabulator**
Data Conversion	Non-structured data to RDF					✓			
	Structured data to LD					✓			✓
	Export RDF/XML, JSON format					✓			
Creating links to other URIs	Forward navigation	✓				✓			✓
	Backward navigation	✓				✓			✓
	RDF model navigation			✓		✓	✓	✓	✓
	Links filtering					✓	✓	✓	
	Navigating Local LD	✓	✓		✓	✓	✓	✓	✓
	Navigating Global LD	✓	✓		✓	✓		✓	✓
Data triage	Simple (keyword search or filter)	✓	✓			✓	✓	✓	
	Complex (SPARQL)	✓		✓	✓	✓	✓	✓	✓
	Use of search engine	✓							
	Caching results					✓			
	Aggregate search results	✓						✓	✓
	Editable results	✓		✓	✓	✓	✓	✓	
	Tagging results					✓			
Browsing mechanism	Entity-Attributes-value view					✓			✓
	HTML view				✓				✓
	Facet views					✓			
	Map view	✓							✓
	Visual view	✓	✓	✓	✓	✓	✓	✓	
	Grid view					✓			
	Layout control	✓		✓			✓	✓	
	“Order by” option					✓			
5-star data schema	Producer 5 stars	5				5			5
	Consumer 5 stars	5	4	4	5	5	4	5	5

example of converting a Wikipedia page to LD. As shown on screen A, the RDF detective service (A-2) allows for the extraction of RDF triples embedded in HTML, which in this case yields 212 triples (A-1). To convert the contents of the Wikipedia page from hypertext to LD, ODE (A-3) extracts the raw data and displays it as LD. The output from the data conversion process also shows related data resources, here resulting in the 2470 triples shown in (B-1). However, we noted that commonly used structured data formats, such as CSV, cannot be converted by this browser.

RDF Gravity, RelationshipFinder and Tabulator do not facilitate converting data. The RelationshipFinder does however work on RDF datastores that contain LD in RDF form.

Generally, browsers can be discriminated in terms of facilities for accessing and combining resources that have different formats, schemas and ontologies. More capable browsers in this regard will allow users to consume, explore and interact with LD more efficiently.

4.2.2 Creating links to other URIs

Dipper is a closed system that does not permit exploration of arbitrary URIs, RDF data stores or other sources. Users are only given the opportunity to explore LD that is stored inside Dipper datastores. Although there are labels for Lookup, Store and Starting Points in Dipper, however, it is hard for the user to realise where to start from to consume the data they need. The user has to explore and click the links to get started but in terms of what should the user click, the browser does not facilitate this for users. Also, the predicates given to explore datasets are not very visible and obvious, e.g. http://ckan.net/ontology/downloadURL is a predicate given for freebase.

The Marbles browser allows the user to navigate to new links. It also gives the user details of the provenance of the data, i.e., where the data was retrieved from. Piggy Bank has a notion of Semantic Bank where multiple users can store and share the Piggy Bank-converted SW contents. This allows new LD to be found and linked to other datasets. However, it does not allow global discovery of new LD resources. Marbles and Piggy Bank are available as Firefox extensions, therefore users can conveniently navigate backward and forward using the browser's in-built navigational features.

Sig.ma facilitates the user to find new LD from the results of queries, through navigation of the links provided in the search results. URIburner allows users to navigate from one domain of LD to a different one linked from the current dataset. It supports both forward and backward navigation.

DBpedia Mobile has good highlighting of links between datasets. However users can only navigate to selected datasets on the SW, such as Geonames, Revyu, Eurostat and Flickr.

IsaViz service does not support navigating links into another datasets. It only supports browsing a single RDF graph.

ODE provides the user with the facility of navigation between datasets. So the user can click on other LD from the given results to navigate to another source. In ODE, users also can view data based on previously selected predicates; links will then be traversed based on their custom properties as shown in Figure 4 (B-2).

RDF Gravity does not provide a facility for highlighting links between different data resources.

TheRelationshipFinder gives users the relationships, if any, between different RDF nodes. However, users do not have the opportunity to navigate to new LD using this application. They can find the relationships between existing LD in an RDF database using this application. One can save the results of the search using a URI and this URI can be dereferenced.

LESS and Tabulator provide support for exploring new LD by following links. However, the user can not custom the browsing data based on the data type.

4.2.3 Data triage

In Dipper, there is no specific support for sorting data or constructing queries on new data. Some datasets with public SPARQL endpoints can be queried and explored, but users need to know about SPARQL query language to write a query. This feature can be leveraged by developing a client user interface for these SPARQL endpoints. However, not all datasets provide a SPARQL endpoint service for accessing their data using the SPARQL query language. Also, some of the default datasets, such as `http://lists.broadminsteruniversity.org/lists/demo`, need official authorisation of a user's credentials to explore the data, which imposes challenges during initial evaluation of the tool.

The Marbles browser offers a SPARQL endpoint, that is a useful facility for technical users, though less so for lay users and domain experts. Marbles uses the search engines Sindice and Falcons for querying a user-entered URI once it has been dereferenced. It can then combine data from different resources into a single view. Marbles does however have limitations; it does not support advanced interaction features such as exporting data or tagging result sets.

Piggy Bank also provides users some facilities for data triage, such as text search and tagging. However, there are no facilities for SPARQL queries or for result aggregation, and domain specific support is also limited.

In Sig.ma, the search terms are text-based, and users can search based on RDF properties. As shown in Figure 2, Sig.ma uses a mixture of query planning, word disambiguation, distributed data source selection, and parallel data gathering to return a list of results for the search query. The confidence or quality of the search results is also provided, a strength of Sig.ma. This allows the user to validate and verify the result even after search results have been shown. For example, if the user is not satisfied with any of the results, they can ask for those results to be changed, resulting in improved results in future searches.

The URIburner service makes use of a public SPARQL endpoint. There is also a Search and Find browser endpoint at `http://linkeddata.uriburner.com/fct/`. Both free text search as well as URI-based searches are supported.

DBpedia Mobile provides for both simple filters that allow simple text and list based querying facilities, and complex SPARQL queries for complex manipulation of linked datasets. DBpedia Mobile also generates a summary view of selected resources by building on the Marbles browser, using semantic search and integrating the results.

IsaViz does not provide any support for data triage, while ODE supports basic sorting of data based on user preferences. As shown in Figure 4 (B-3), ODE users can retrieve and view the requirements based on either an automatic or editable data retrieval schema, and the depth of the link traversal can be adjusted by the user. It provides support for cached results as well.

RDF Gravity has three simple query facilities: support for RDF query language (RDQL — a precursor to SPARQL) so a technical user can write queries based on their needs; support for text search over concepts, properties and instances specified in a RDF data set; and enabling global and local filters to specify the view of results.

RelationshipFinder provides free text search of objects or elements in a plain text view. The user enters two objects of interest to them, which are preferably described by articles in the English Wikipedia. These selected elements are first semi-automatically mapped to unique objects of SW datasets in the underlying RDF database. These datasets are then crawled for relationships to present to the user.

Finally, in Tabulator, the user can select fields to use as simple query parameters. Due to current technical limitations, query and manipulation options could not be further explored.

To conclude this part, some of the more functional browsers demonstrate impressive capabilities for data triage, retrieval and extraction. In particular, they show strong support for SPARQL, providing a flexible range of options for sorting, retrieving and finding new data. In cases where datasets can not be queried via SPARQL, such as when an endpoint is not exposed, some of the browsers also utilise text-based search for data retrieval. In the case of Sig.ma, result sets can also be reviewed, approved and rejected, providing useful feedback to the search heuristics and algorithms used.

4.2.4 Browsing mechanism

Browsing mechanisms for visual browsers differs considerably. At one end of the spectrum, some, such as Dipper and Marbles, use basic HTML text representation, making them closer in approach to pure text-based browsers. Marbles also includes a text summary feature that describes resources.

Other browsers use more advanced visual approaches. Sig.ma, for example, enables users to browse search results by dividing the screen into two panels: the left panel is used to view the aggregated data sources, and the right panel is used to present the facts between the resources. Although it has a search engine, Sindice, whose features and APIs can be leveraged to perform faceted browsing, Sig.ma only supports “flat” one dimensional browsing.

In contrast, other browsers offer different data views. Piggy Bank has a faceted view that provides a summary about an item with different aspects. URIburner presents data in the form of “Entity-attributes-value”, which uses textual representation for viewing information. It also provides an extra feature that enables uses to pivot collections around the current data view.

The browsing mechanism in DBpedia Mobile is a map view complementing HTML information, as well as, where available, some visualisation of specific objects themselves. It also gives the user the opportunity to have view control over the results by using zoom-in and zoom-out options.

In IsaViz, RDF data is presented as graphs, which can navigated along graph vertices and edges. ODE supports multiple ways of viewing data, including faceted view, grid view, and visual view, and also includes data ordering options as shown in Figure 4(B-5)(B-6). RDF Gravity also supports graph-based views, while RelationshipFinder supports both graph- and tree-based views. Finally, in Tabulator, there are different views, such as map view, timeline view and graph view. Due to difficulties mentioned earlier in Section 4.1.2, we were unable to effectively evaluate Tabulator.

4.2.5 5-star data

Dipper scored a five star rating since it supports LD objects — using URIs to identify entities and RDF to represent data. As discussed earlier, there were however some problems with actually following LD in this tool. Marbles also scored five stars, although it only consumes and cannot produce LD output.

Much of the data in Piggy Bank is a mixture of 2 star, 3 star and 4 star — that is, it is represented in more or less structured formats. Increased usage of the SW could raise some of this data to 5 stars, as more of it becomes linked — however, in some cases the browser does not provide a full context to users, as there are no outgoing links to follow in the datasets currently used.

The data support in Sig.ma and URIburner are both 5-star, but in the case of URIburner some links needs to be further curated, in order to support further resource discovery. DBpedia Mobile is a 5 star system as it produces RDF data, and also provides links to other external datasets such as GeoNames, Revyu, EuroStat and Flickr.

ODE is also rated at 5 stars as it produces LD in RDF, and also allows navigation to other external data linked to and from the produced LD. Tabulator similarly produces new data that is structured, and provides links to other datasets, so this is also rated 5 stars.

Finally, LESS, IsaViz, RDFGravity and RelFinder solely consume 4 star and 5 star data, but they do not produce LD output.

5 Conclusion and Future Work

This study set out to determine the criteria to evaluate the interactive capabilities of semantic browsers. The main focus here is on user interaction features, which are particularly useful to more technical users, for exploring and consuming LD. An extension of this study would investigate more deeply organisational factors, HCI relevant issues and query implementation issues.

For the LD world to be useful to the end users, powerful semantic browsers need to be developed that can perform well against the kinds of criteria we presented here. First, browsers should provide support for *data conversion*, for users to convert their data from non-LD into LD. Second, browsers should allow for the *creation of links to other URIs* to connect more information. Third, browsers should provide *query capabilities for data triage* along with result sets that can be viewed in different ways by users. More importantly, SPARQL support is crucial as it allows results to be extracted readily from very large datasets. Fourth, users will interact with LD if the *browsing mechanism* is easy and flexible to use, and presents data in an intuitive way. Fifth, browsers should support highly structured and LD, in accordance to Berners-Lee's *5 star* criteria.

A further finding of our study suggests that the Dadzie and Rowe [8] classification between text and visual semantic browsers can be further distinguished into 4 kinds: RDF model browsers, domain-specific browsers, generic-domain browsers and links-relationship browsers. Selection and ranking of these browsers depends upon end user requirements. If those requirements are limited to the use of RDF format only, then IsaViz or RDF Gravity rate highest in terms of capabilities. For domain-specific purposes, where the data set is predefined, then the Dipper, Piggy Bank or RDF Gravity browser would be the most appropriate according to our evaluation. For exploration of the global SW, browsers such as Marbles, Sig.ma and URIburner provide text-representation of data, while browsers such as ODE and Tabulator provide more options for data visualisation. For the understanding of the relationships between linked datasets, RelFinder presents the relationships in an intuitive manner.

The outcome of this study indicates that the Sig.ma, URIburner and ODE are clearly the most powerful browsers currently available for exploring the Web of LD. They provide sophisticated functions that manipulate, integrate and explore different data resources, and this is of particular benefit to more technical users.

We also observe that the Marbles, Sig.ma, Uriburner and ODE browsers have shown more capabilities for exploring and manipulating data, and therefore were evaluated more highly in this study than their corresponding evaluation according to the criteria presented by Dadzie and Rowe [8]. These browsers substantially reduce the challenge of user interaction with LD. Our evaluations of Dipper, Disco, the Tabulator, DBpedia Mobile, and RelFinder arrive at the same outcome as Dadzie and Rowe [8], which suggests that these browsers are also useful when applied to interacting with LD. However, in our study, Piggy Bank, Fenfire, IsaViz, LESS and RDFGravity were evaluated more poorly, which suggests that their strengths lie more on the visualisation of RDF data than in its exploration and manipulation.

The future development of interactive browser capabilities based on the suggestions put forward in this study will further bridge the gap between the world of LD and human users. Improved LD visualisation and interaction tools will enable users to more readily comprehend LD.

Acknowledgment: This research was supported under Australian Research Council's Linkage Projects funding scheme (project number LP0990509).

References

[1] S. Auer, R. Doehring, and S. Dietzold. Less - template-based syndication and presentation of linked data. In *Proceedings of ESWC (2)*, pages 211–224, 2010.

[2] C. Becker and C. Bizer. Exploring the Geospatial Semantic Web with DBpedia Mobile. *Web Semantics: Science, Services and Agents on the World Wide Web*, 7(4):278 – 286, 2009. Semantic Web challenge 2008.

[3] T. Berners-Lee. Linked Data. *http://www.w3.org/DesignIssues/LinkedData.html*, 2006. Access Date: 2009-10-08.

[4] C. Bizer, R. Cyganiak, and T. Heath. How to Publish Linked Data on the Web. *Publish*, 20(October):43, 2007.

[5] C. Bizer, T. Heath, and T. Berners-Lee. Linked data - the story so far. *International Journal on Semantic Web and Information Systems (IJSWIS)*, 5:1–22, 2009.

[6] C. Bizer, T. Heath, K. Idehen, and T. Berners-Lee. Linked Data on the Web (LDOW2008). In *Proceeding of WWW '08*, pages 1265–1266, New York, NY, USA, 2008. ACM.

[7] C. Bizer, J. Lehmann, G. Kobilarov, S. Auer, C. Becker, R.Cyganiak, and S. Hellmann. DBpedia - A crystallization point for the Web of Data. *Web Semantics: Science, Services and Agents on the World Wide Web*, 7(3):154 – 165, 2009. The Web of Data.

[8] A. Dadzie and M. Rowe. Approaches to Visualising Linked Data: A Survey. *Semantic Web*, 2(2):89–124, 2011.

[9] O. Hartig, J. Sequeda, J. Taylor, and P. Sinclair. How to Consume Linked Data on the Web: Tutorial Description. In *Proceedings of WWW '10*, pages 1347–1348, New York, NY, USA, 2010. ACM.

[10] T. Hastrup, R. Cyganiak, and U. Bojars. Browsing Linked Data with Fenfire. In *Proceedings of LDOW2008*, volume 369, 2008.

[11] T. Heath and C. Bizer. *Linked Data: Evolving the Web into a Global Data Space*. Synthesis Lectures on the Semantic Web: Theory and Technology. Morgan and Claypool, 2011.

[12] D. Huynh, S. Mazzocchi, and D. Karger. Piggy Bank: Experience the Semantic Web Inside Your Web Browser. In *ISWC 2005*, volume 3729 of *LNCS*, pages 413–430. Springer Berlin / Heidelberg, 2005.

[13] T.B. Lee, J. Hendler, and O. Lassila. The semantic web. *Scientific American*, 284(5):34–43, 2001.

[14] S. Lohmann, P. Heim, T. Stegemann, and J. Ziegler. The RelFinder User Interface: Interactive Exploration of Relationships between Objects of Interest. In *Proceedings of IUI 2010*, pages 421–422, New York, NY, USA, 2010. ACM.

[15] S. Lohmann P. Heim and T. Stegemann. Interactive Relationship Discovery via the Semantic Web. In *Proceedings of ESWC 2010*, volume 6088 of *LNCS*, pages 303–317, Berlin/Heidelberg, 2010. Springer.

[16] B. Shneiderman, C. Plaisant, M. Cohen, and S. Jacobs. *Designing the User Interface: Strategies for Effective Human-Computer Interaction.* Addison Wesley, 5th edition, March 2009.

[17] L. Chilton D. Connolly R. Dhanaraj J. Hollenbach A. Lerer T. Berners-Lee, Y. Chen and D. Sheets. Tabulator: Exploring and analyzing linked data on the semantic web. In *In Proceedings of SWUI 2006*, 2006.

[18] G. Tummarello, R. Cyganiak, M. Catasta, S. Danielczyk, R. Delbru, and S. Decker. Sig.ma: Live views on the Web of Data. *Web Semantics: Science, Services and Agents on the World Wide Web*, 8(4):355 – 364, 2010.

[19] G. Tummarello, R. Delbru, and E. Oren. Sindice.com: Weaving the Open Linked Data. In *The Semantic Web*, volume 4825, pages 552–565. Springer Berlin / Heidelberg, 2007.