Semantic Wonder Cloud: Exploratory Search in DBpedia

Roberto Mirizzi[1], Azzurra Ragone[1,2], Tommaso Di Noia[1], and Eugenio Di Sciascio[1]

[1] Politecnico di Bari – Via Orabona, 4, 70125 Bari, Italy
mirizzi@deemail.poliba.it, {ragone,dinoia,disciacio}@poliba.it
[2] University of Trento – Via Sommarive, 14, 38100 Povo (Trento), Italy
ragone@disi.unitn.it

Abstract. Inspired by the Google Wonder Wheel[1], in this paper we present Semantic Wonder Cloud (SWOC): a tool that helps users in knowledge exploration within the DBpedia dataset by adopting a hybrid approach. We describe both the architecture and the user interface. The system exploits not only pure semantic connections in the underlying RDF graph but it mixes the meaning of such information with external non-semantic knowledge sources, such as web search engines and tagging systems. Semantic Wonder Cloud allows the user to explore the relations between resources of knowledge domain via a simple and intuitive graphical interface.

1 Introduction

"*The Web, they say, is leaving the era of search and entering one of discovery. What's the difference? Search is what you do when you're looking for something. Discovery is when something wonderful that you didn't know existed, or didn't know how to ask for, finds you.*" [11]. When Jeffrey O'Brien wrote this sentence in 2006, he was referring to recommender systems on the web [1]. Nevertheless, it can be also interpreted in a broader sense if we think that the recommended items are pieces of knowledge extracted from a knowledge base. In other words, that sentence summarizes in an excellent way the problem of information discovery in a knowledge repository. The main issue there is assisting a user to discover new knowledge in a repository and what might be the role played by the interconnected nature of the Semantic Web.

Thanks to its hyperlinked structure, the Web defined a new way of browsing documents and knowledge: selection, navigation and trial-and-error tactics [12] were and still are exploited by users to search for relevant information satisfying some initial requirements. In [12], the author distinguishes among three distinct categories of search strategies: lookup, learn and investigate. He also groups the last two under the umbrella of *exploratory search*. Lookup has database systems as background technology and it is used for finding exact information, i.e.,

[1] http://www.googlewonderwheel.com/

F. Daniel and F.M. Facca (Eds.): ICWE 2010 Workshops, LNCS 6385, pp. 138–149, 2010.

records in a database or documents containing a specific keyword. The above mentioned search strategy is by far the most used in the current "syntactic" web where the information sources are mainly based on textual documents. Nevertheless, it is applied in many of the current applications exploiting structured metadata available in the Semantic Web: records are RDF triples and keywords are URIs with a specific associated meaning (semantics). In the other two search strategies – learn and investigate – the final aim of the end user changes [13]. She is no more interested in a fact retrieval or query answering but her focus is on:

Learn: Knowledge acquisition, comprehension of new concepts and ideas, comparison of information;
Investigate: Analysis, synthesis, evaluation, knowledge discovery.

In this paper, inspired by *Google Wonder Wheel*, we present *Semantic Wonder Cloud* (SWOC): a tool for exploratory knowledge search for DBpedia [5]. Differently from other RDF explorers, SWOC allows the user to explore DBpedia not just via directed links in the RDF dataset but via newly computed associations between DBpedia nodes. Such new associations are computed exploiting extra-knowledge extracted from Web search engines and social tagging systems. Main contributions on this paper are:

- novel approach to *exploratory search* in the Semantic Web: in this work we focus on DBpedia and present a new way to explore a RDF dataset discovering new knowledge associations;
- visual tool to guide the user during the knowledge discovery process;
- hybrid approach to rank pairs of resources on DBpedia: our system mixes a *semantic-based* approach, which relies on the structure of the RDF graph, and a *text-based* IR approach, which computes the popularity of the resources by querying external information sources

The remainder of this paper is structured as follows. In Section 2 we present Linked Data and in particular DBpedia as the knowledge base we use in this work. Section 3 provides details on the implementation of SWOC. Section 4 overviews related work. An outline of the future work concludes the paper.

2 Linked Data and DBpedia

The idea behind Linked Data [4] is using the Web to allow linking data and simplifying the publication of new interconnected data on the Web. It proposes a new method of exposing, connecting and sharing data through dereferenceable URIs on the Web. The goal is to extend the Web by publishing various open datasets as RDF triples on the Web and by setting RDF links between data items from different data sources. According to this aim, URIs are fundamental to identify everything. Using HTTP URIs, things can be referred to and looked up both by people and by agents. Linked Data uses RDF to describe things in the world. In this paper we focus on DBpedia [5], that is one of the main clouds of the

Linked Data graph. It is the machine-understandable equivalent of Wikipedia project. It is possible to ask queries against DBpedia (through its SPARQL endpoint http://dbpedia.org/sparql), and link other data sets on the web to DBpedia data. Currently the DBpedia dataset (version 3.5.1[2]) contains almost three million and half resources, including more than three hundred thousand persons, over four hundred thousand places, thousands of films, companies, music albums, etc.. All this information is stored in RDF triples. The whole knowledge base consists of over one billion triples. DBpedia labels and abstracts of resources are stored in 92 different languages. The graph is highly connected to other RDF dataset of the Linked Data cloud. There are more than half million categories, inherited from Wikipedia, and almost one hundred thousand YAGO [21] categories. DBpedia has many strong points over existing knowledge bases: it is spread over many domains; being based on Wikipedia, it represents a real community agreement; it follows the changing in Wikipedia, so it is continuously updated; it is multilingual. Moreover DBpedia has a central role in the Linked Data community effort: it is one of the central interlinking-hubs of the emerging Web of Data, inducing data providers to link their RDF datasets to DBpedia. The DBpedia RDF dataset is hosted and published using *OpenLink Virtuoso*[3]. This infrastructure gives access to DBpedia's RDF data through a SPARQL endpoint, with HTTP support for any Web client's standard GETs for HTML or RDF representations of DBpedia resources. In DBpedia there is a special kind of resources called **categories**[4]. Since in Wikipedia they are used to classify and cluster sets of documents, in DBpedia they classify sets of resources. They might be seen as abstract concepts describing and clustering sets of resources. As a matter of fact, to stress this relation, every DBpedia category is also a skos:Concept. Moreover, since DBpedia categories have their own labels we may think of these labels as names for clusters of resources. Most categories can have a number of other categories listed as subcategories. Exploring from more general categories to more specific ones it is possible to cluster resources according to more fine-grained domains. Similarly, navigating towards more general categories the obtained clusters will be broader and less domain-specific. In order to avoid an explosion of articles belonging to a given Category, articles are not usually placed in every category which they logically belong to. In many cases they are placed just in the more specific subcategories. For this reason, in Wikipedia it may be necessary to dig up to find all the categories an article belongs to. DBpedia maintains these hierarchical relations using the SKOS vocabulary. In particular the property skos:subject of the SKOS vocabulary relates a Wikipedia article to its Category. Actually the skos:subject property has been deprecated from the SKOS vocabulary, although it has not yet been replaced by any other property[5].

[2] http://wiki.dbpedia.org/Downloads351

[3] http://virtuoso.openlinksw.com/

[4] http://en.wikipedia.org/wiki/Help:Category

[5] A natural candidate for the replacement could be the dc:subject property, belonging to *Dublin Core* specification, as proposed in http://www.w3.org/2006/07/SWD/wiki/SkosDesign/Indexing

Similarly in DBpedia a Category is linked to another Category (specifically a subcategory to its category) through the skos:broader property.

3 SWOC: Semantic Wonder Cloud

The graph-based structure of datasets in the Semantic Web allows a natural visualization and browsing of the formalized information simplifying the implementation of learning and investigating strategies for knowledge acquisition and discovery. Using an *Orienteering* [22] behavior, a user starts from an initial vague idea of what she is looking for and navigates through the information space. It has been shown that *Orienteering* navigation decreases the cognitive load, maintains a sense of location and gives a better feeling for context [10].

Effective user interfaces play a crucial role in order to provide a satisfactory user experience during an exploratory search. There are two main trends in visualizing and navigating RDF datasets [6]: via browsing a labeled oriented graph or displaying RDF properties as browsable facets of a node. The main issue of both approaches is to filter information which is not relevant for the explorative task. If we visualize all the triples together with their connections we have a huge unreadable representation of the underlying knowledge [18]. In fact, RDF triples are conceived to represent information for machine-to-machine interaction. A lot of information triplified in datasets is very useful in many automated computational tasks (e.g., web service interaction) but it is completely useless from a human perspective during an exploratory search [7].

In this section we introduce *Semantic Wonder Cloud* (SWOC), a tool that helps users in exploratory knowledge search in DBpedia. The front-end of the system, inspired by *Google Wonder Wheel*, is available at http://sisinflab.poliba.it/semantic-wonder-cloud/index/. A sketch of the functional architecture is represented in Figure 1. SWOC consists of two main subsystems. The first subsystem, the SWOC explorer, is a Graphical User Interface (GUI) designed to allow exploration of DBpedia. The second subsystem – *DBpediaRanker* – is the back-end of the whole system whose main task is to compute similarity between pairs of DBpedia nodes. In order to compute a *similarity value* between two resources res_1 and res_2, *DBpediaRanker* performs three main tasks:

- analysis of the underlying RDF **graph structure**;
- **textual analysis** on the abstract of the resources;
- exploitation of **external information sources**, such as search engines and social tagging systems.

More details are provided in Section 3.1.

We stress that, differently from common RDF visualization tools[6], SWOC does not visualize the DBpedia RDF graph as it is. There are some key differences w.r.t. other RDF explorers.

[6] By way of example refer to http://timerollson.wordpress.com/2007/09/30/links-for-2007-09-30/ and http://www.mkbergman.com/414/large-scale-rdf-graph-visualization-tools/

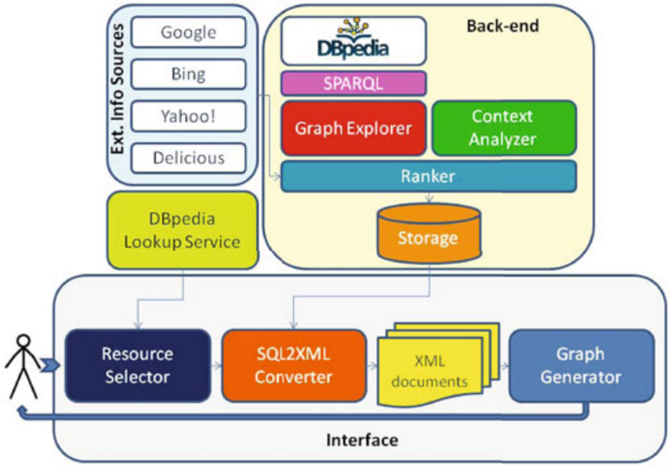

Fig. 1. *Semantic Wonder Cloud* functional architecture

- The graph displayed by SWOC represents possible new associations between resources computed by mixing the *semantic* knowledge formalized in the DBpedia dataset and the *non-semantic* statistical knowledge available from web search engines and social tagging systems. Hence, the navigation is driven by knowledge associations between resources rather than guided by the structure of the underlying dataset. Displayed associations may even not exist in the original DBpedia graph.
- Each node in the displayed graph has a different size representing how relevant it is with respect to the resource represented in the center of the graph itself.
- There are two main classes of nodes in SWOC: one representing DBpedia categories, the other representing instances. This reflects in some way the different nature of the knowledge they represent. Categories are used to group and classify resources. Hence, if the user clicks on a category node then she sees the most relevant (popular) instances of that category together with its most relevant categories.

Referring to the previous characteristics, our tool has been designed to be a user centered explorer of RDF knowledge bases rather than a common RDF graphic visualizer. Actually, SWOC can not be considered an RDF visualizer in the classical sense. It is a tool useful to help end-users in exploratory knowledge search. Usually end-users do not care about obtaining a sketch of the whole result set (think to results returned by a search engine: rarely you would go beyond the second page), but are more interested in finding something *relevant* among the first results. Thanks to SWOC, end-users can navigate from a resource to another one exploring and discovering new knowledge. Following the exploratory search paradigm, SWOC differentiates also from traditional search engines. The latter allow to find what you are looking for and already know, SWOC allows to explore what you probably did not know to exist.

3.1 Backend

In this section we describe the back-end of the system, based on our hybrid ranking algorithm *DBpediaRanker* [14], used to rank resources in DBpedia. This algorithm computes the *relevance* of DBpedia resources w.r.t. a starting node. In a nutshell, *DBpediaRanker* explores the DBpedia graph and queries external information sources in order to compute a *similarity value* for each pair of resources reached during the exploration.

The graph browsing, and the consequent ranking of resources, is performed *offline* and, at the end, the result is a weighted graph where nodes are DBpedia resources and weights represent the similarity value between any pair of nodes. The graph so obtained will then be used at *runtime* in the *exploratory search*, to look for resources that are semantically related to the initial one. The similarity value between any pair of resources in the DBpedia graph is computed querying search engines and social bookmarking systems and exploits *textual* and *link analysis* in DBpedia. For each pair of resources in the explored graph, we perform a query to each external information source: we search for the number of returned web pages containing the labels of each node individually and then for the two labels together. Moreover, we look at, respectively, **abstracts** in Wikipedia and **wikilinks**, i.e., links between Wikipedia pages. Specifically, given two resource nodes res_1 and res_2, we check if the label of node res_1 is contained in the abstract of node res_2, and vice versa. The rationale behind this check is that if a DBpedia resource name appears in the abstract of another DBpedia resource it is reasonable to think that these resources are related with each other. For the same reason, we also check if the Wikipedia page of resource res_1 has a link to the Wikipedia page of resource res_2, and vice versa.

The back-end of SWOC is composed by the following components:

- **Graph Explorer:** it explores the DBpedia graph, through SPARQL queries, starting from a set of initial seeds. Each node is explored within a predefined number of hops and following a predefined set of properties. If you want to explore only a domain-specific subset of the DBpedia graph (see below), the initial seeds have to belong to that domain (i.e. to the same context). During our experiments we set the maximum explored depth equal 2, and we explored the graph following two properties belonging to the SKOS vocabulary, i.e., `skos:subject` and `skos:broader` (see Section 2). A deep explanation of these choices is provided in [14].
- **Context Analyzer:** it allows to limit the exploration of the graph to a specific context. In this paper the exploration has been limited to the *IT* domain (specifically *databases* and *programming languages*). It uses the *Ranker* (see below) to determine if a resource belongs to the context. The context is represented by the most popular DBpedia categories (see Section 2), i.e., the categories that are reached more often during the exploration of the graph.

For example, in the selected domain, one of the most popular categories is *Programming languages*[7].

- **Ranker:** this is the core component of the system. It determines a similarity value between any pairs of nodes (res_1 and res_2) discovered by the *Graph Explorer*; this similarity value is the weight associated to the edge between the two resources. The score is computed by querying the external information sources with the following formula:

$$sim(res_1, res_2, info\_source) = \frac{p_{res_1,res_2}}{p_{res_1}} + \frac{p_{res_1,res_2}}{p_{res_2}} \quad (1)$$

  where p_{res_1,res_2} is the number of documents that contain (or have been tagged with) both the label associated to res_1 and the one associated to res_2, and p_{res_1} and p_{res_2} are the number of documents that contain (or have been tagged by) the label associated to respectively res_1 and res_2. Furthermore *Ranker* exploits additional information from `DBpedia`. In fact it checks if there is a *dbpprop:wikilink* between res_1 and res_2 and vice versa, and in positive case it assumes a stronger relation between the two resources. Finally, the component checks if the *label* of res_1 is contained in the *abstract* of res_2 and vice versa.
- **Storage:** all the similarity values between the pairs of resources, together with the “popularity” of each resource (calculated as the number of times that each node is reached in the exploration) are stored in a DBMS for an efficient retrieval at runtime.

3.2 Interface

The interface of `SWOC` is a Flash-based web application that presents the knowledge stored in the back-end (see Section 3.1). A screenshot of the web interface is presented in Figure 2.

The first step for the user is the choice of a node to start the exploration. This is done by keying in some characters, concerning a subject to explore, in the text field marked as *(a)* in Figure 2. The system responds with an auto-complete list containing a set of labels which refer to `DBpedia` resources where each of them is described by a unique *URI* which identifies it with no ambiguity. This list is obtained by querying the `DBpedia` *URI* lookup web service[8]. Note that, thanks to the uniqueness of `DBpedia` *URI* identifying suggested resources, the system does not suffer from problems such as synonymy or polysemy as for simple keywords-based queries to search engines.

After the user selects a resource res_1 from the drop down list, the system generates a graph, centered in res_1 (that has the maximum circle radius), and whose constellation is constituted by the ten most similar resources res_i, $(i = 1, ..., 10)$, w.r.t. res_1. If a `DBpedia` category is highly relevant w.r.t. res_n, it is drawn in the constellation with a different color (the radius of the circle follows the same rules as seen before).

[7] `http://dbpedia.org/resource/Category:Programming_languages`

[8] `http://lookup.dbpedia.org/api/search.asmx`

Fig. 2. *Semantic Wonder Cloud* web-interface

As seen in Section 3.1, the similarity value between res_1 and its constellation has been calculated off-line by the *Ranker*. In the interface the similarity value is normalized on a scale in the range $[1, 10]$ and rounded to the nearest integer, where 10 means that res_i is highly relevant w.r.t. res_1 (according to our *Ranker*), and 1 means that res_1 and res_i are not very similar. These values influence the size of the circle radii: the bigger the constellation's circles are, the more important the referring resources are w.r.t. res_1.

In order to generate the graph, `SWOC` uses a `XML` file containing information on: relation types (instance, category), the types of node, the nodes to be plotted and the relations among them. An example of such file is:

```
<?xml version="1.0" encoding="UTF-8"?>
<RelationViewerData>
    <Settings appTitle="Semantic Wonder Cloud" startID="7" defaultRadius="150" maxRadius="180">
        <RelationTypes>
            <UndirectedRelation color="0xAAAAAA" lineSize="4"/>
        </RelationTypes>
        <NodeTypes>
            <Node1/>
            <Node2/>
            ...
            <Node10/>
        </NodeTypes>
    </Settings>

    <Nodes>
        <Node10 id="7" name="PHP" URI="http://dbpedia.org/resource/PHP" dataURL="moreNodes.php?id=7">
            <![CDATA[
                 PHP: Hypertext Preprocessor is a widely used, general-purpose scripting language that was originally
                 designed for web development to produce dynamic web pages. For this purpose, PHP code is embedded ...
            ]]>
        </Node10>
        ...
    </Nodes>

    <Relations>
        <UndirectedRelation fromID="7" toID="69"/>
        ...
    </Relations>
</RelationViewerData>
```

Each node corresponds to a `DBpedia` resource, and it can be uniquely identified by its *URI*. It has also an abstract which goes in the *CDATA* section of the Node content.

Being the whole DBpedia dataset constituted by more than three millions resources, it was not feasible to have a single XML file containing all the nodes. Even in cases where the *Context Analyzer* (see Section 3.1) had limited the graph exploration, the number of nodes would still be large, and the relations among nodes would be a really huge set[9]. For this reason we opted a dynamic creation of the required XML file. For each selected resource (i.e., a node that appears in the center of the graph), the corresponding XML file is dynamically created at run-time by querying the DB populated by the *Storage* module in the off-line computation. The *SQL2XML Converter* component of the system (see Figure 1) takes as input the *ID* of the resource res_n to explore, then it queries the underlying database and generates the XML file containing the top-10 relevant nodes w.r.t. res_n. Finally the *Graph Generator* accepts as input the XML file previously created and plots the corresponding graph on the screen.

During the exploration, a local list (marked as *(b)* in Figure 2) is populated with discovered nodes up to that moment: it is possible to browse a node from that list in order to directly explore it. Moreover the system keeps track of the exploration path (marked as *(c)* in Figure 2): the user can move backward and forward through resources that have previously been explored.

On the right side of the interface, an area is displayed containing the DBpedia abstract of the selected resource res_n, together with its *URI* and the set of the most relevant DBpedia categories res_n belongs to.

4 Related Work

The rapid growth of semantic metadata and related ontologies in the Semantic Web of Data asks for new tools and approaches to data exploration and visualization. Nowadays, RDF datasets with their formalized knowledge are not only developed for machine-to-machine interaction but also for knowledge discovery and navigation. One of the main issues to be faced is how to manipulate these huge repositories of information in a "*overview first, zoom and filter, then details-on-demand*" fashion as pointed out in [19]. Here we discuss only some approaches representative of the graph-based approach and of the faceted one even though alternative solutions have been adopted for specific tasks [17]. We do not focus on issues typical of clustering and faceted categories for information exploration (see [9]) but we concentrate on specific problems related to the semantic nature of the underlying data. Many approaches to semantic data visualization adopt a graph-based approach exploiting concept relations, entities aggregation and number of nodes instantiating a given class. In [15] the authors focus on discovering disconnections in the ontology graph for a visualization of smaller graphs. Although the smaller dimension of the visualized graphs the approach does not scale well for huge dataset as DBpedia. Moreover, it exploits only topological information to suggest the navigation of the graph. In [20], a cluster-based visualization is adopted. Also in this case only relations explicitly stated in the

[9] Each node in the DBpedia graph has on the average more than half a thousand nearby nodes within two hops.

dataset are displayed. A slightly different approach is adopted in [10] where the author focuses on a visual rendering of information stored in Freebase[10] and Wikipedia. In the latter case they use `SemanticProxy`[11] to extract structured information from textual pages. In this case the graph displays only some aggregated and filtered information related to a given resource. Faceted browsing [23] has been widely adopted for many `RDF` dataset spanning from `DBpedia` to `DBLP`. A lot of work has been produced in this area. Faceted browsing improves usability over current interfaces and `RDF` visualizers as it provides a better information lookup w.r.t. to keyword searches. Nevertheless faceted interfaces are domain-dependent, do not fully support graph-based navigation, and they become difficult to use for the end user as the number of presented choices grows. An implementation for faceted navigation of arbitrary `RDF` data is presented in [16]. They identify important facets by ranking the predicates that best represent and most efficiently navigate the dataset. Differently from our approach, their ranking metrics take into account only the structure of the dataset while we also exploit external information sources to improve the results. Hahn et al. [8] present a faceted browser for Wikipedia. The system is based on the `DBpedia` data extraction framework and `Neofonie`[12] search technology. Faceted views are also available in `Virtuoso`[13], but here ranking is obtained only from text and entity frequency while semantics associated with links is not explored. This aspect leads to a poor "semantic" ranking in most cases. *"Context-aware semantic association ranking"* [2] is useful because it allows to find and present to the end-user the top relevant information. A common problem with faceted browsing is the impossibility to navigate through relations different from the ones *explicitly* represented in the dataset. On the contrary `SWOC` allows to discover new knowledge by exploring also *implicit* relations that do not existed explicitly in the dataset.

5 Conclusions and Future Work

Ten years after the seminal paper by Tim Berners-Lee et al. [3], the Semantic Web is finally becoming a well established reality. New technologies have been developed able to manipulate large sets of semantic metadata available in on-line `RDF` datasets. The Semantic Web, as a web of data, can be considered nearly done. Together with the availability of new tools to manipulate semantic metadata, we see the flourishing of mash-ups able to aggregate metadata coming from heterogeneous repositories and new semantic search engines able to provide more precise results to the user, as well as new opportunities and new means of manipulating semantic information appeared. Thanks to the highly interconnected nature of semantic metadata the user may browse knowledge repositories

---

[10] `http://www.freebase.com`
[11] `http://www.semanticproxy.com`
[12] `http://www.neofonie.de/index.jsp`
[13] `http://www.openlinksw.com/dataspace/dav/wiki/Main/VirtuosoFacetsViewsLinkedData`

discovering new relations and information. In this paper we presented SWOC: a tool for exploratory knowledge search in DBpedia. The tool exploits both knowledge encoded in DBpedia graph and statistical knowledge disseminated in web search engines and social tagging systems in order to find and display new associations between pair of nodes based on their *semantic similarity*. Thanks to the graphical interface of SWOC, the end-user is guided through the exploration of the knowledge space represented by the DBpedia dataset. The associations generated by the system allow the user to discover new knowledge via a simple and intuitive point-and-click interaction with the graphical interface. We plan to use SWOC has a new way to navigate through tags in social web sites like Delicious in order to mix exploratory search and lookup search for a better user experience. The idea is that once the user finds a new knowledge item represented by the corresponding DBpedia *URI* then she is redirected to its corresponding web resources/documents via a lookup in tagging systems or even in web search engines.

Acknowledgment

We are very grateful to Diego Guario and Moritz Stefaner for the implementation of SWOC and to Enrico Motta for fruitful discussion. This research has been supported by Apulia Strategic projects PS_092, PS_121, PS_025.

References

1. Adomavicius, G., Tuzhilin, A.: Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. IEEE Transactions on Knowledge and Data Engineering 17(6), 734–749 (2005)
2. Aleman-Meza, B., Halaschek-Wiener, C., Arpinar, I.B., Sheth, A.P.: Context-Aware Semantic Association Ranking. In: Proceedings of SWDB 2003, The first International Workshop on Semantic Web and Databases, Co-located with VLDB 2003, Humboldt-Universität, Berlin, Germany, September 7-8, pp. 33–50 (2003)
3. Berners-Lee, T., Hendler, J., Lassila, O.: The Semantic Web - A new form of Web content that is meaningful to computers will unleash a revolution of new possibilities. Scientific American 284(5), 34–43 (2001)
4. Bizer, C., Heath, T., Berners-Lee, T.: Linked data - the story so far. International Journal on Semantic Web and Information Systems 5(3), 1–22 (2009)
5. Bizer, C., Lehmann, J., Kobilarov, G., Auer, S., Becker, C., Cyganiak, R., Hellmann, S.: Dbpedia - a crystallization point for the web of data. In: Web Semantics: Science, Services and Agents on the World Wide Web (July 2009)
6. Deligiannidis, L., Kochut, K.J., Sheth, A.P.: Rdf data exploration and visualization. In: CIMS 2007: Proceedings of the ACM first workshop on CyberInfrastructure: information management in eScience, pp. 39–46. ACM, New York (2007)
7. Geroimenko, V., Chen, C.: Visualizing the Semantic Web: XML-Based Internet and Information Visualization, 2nd edn. Springer, London (2006)
8. Hahn, R., Bizer, C., Sahnwaldt, C., Herta, C., Robinson, S., Brgle, M., Dwiger, H., Scheel, U.: Faceted wikipedia search. In: 13th International Conference on Business Information Systems, BIS 2010 (2010)

9. Hearst, M.A.: Clustering versus faceted categories for information exploration. ACM Commun. 49(4), 59–61 (2006)
10. Hirsch, C., Hosking, J., Grundy, J.: Interactive visualization tools for exploring the semantic graph of large knowledge spaces. In: Workshop on Visual Interfaces to the Social and the Semantic Web (VISSW 2009), IUI 2009 (2009)
11. O'Brien, J.M.: The race to create a 'smart' Google (2006), http://money.cnn.com/magazines/fortune/fortune_archive/2006/11/27/8394347/index.htm?section=money_latest
12. Marchionini, G.: Exploratory search: from finding to understanding. ACM Commun. 49(4), 41–46 (2006)
13. Marchionini, G., Shneiderman, B.: Finding facts vs. browsing knowledge in hypertext systems. Computer 21(1), 70–80 (1988)
14. Mirizzi, R., Ragone, A., Di Noia, T., Di Sciascio, E.: Ranking the linked data: the case of dbpedia. In: Benatallah, B., Casati, F., Kappel, G., Rossi, G. (eds.) ICWE 2010. LNCS, vol. 6189, pp. 337–354. Springer, Heidelberg (2010)
15. Mutton, P., Golbeck, J.: Visualization of semantic metadata and ontologies. In: IV 2003: Proceedings of the Seventh International Conference on Information Visualization, Washington, DC, USA, p. 300. IEEE Computer Society, Los Alamitos (2003)
16. Oren, E., Delbru, R., Decker, S.: Extending faceted navigation for rdf data. In: Cruz, I., Decker, S., Allemang, D., Preist, C., Schwabe, D., Mika, P., Uschold, M., Aroyo, L.M. (eds.) ISWC 2006. LNCS, vol. 4273, pp. 559–572. Springer, Heidelberg (2006)
17. Petrelli, D., Mazumdar, S., Dadzie, A.-S., Ciravegna, F.: Multi visualization and dynamic query for effective exploration of semantic data. In: Bernstein, A., Karger, D.R., Heath, T., Feigenbaum, L., Maynard, D., Motta, E., Thirunarayan, K. (eds.) ISWC 2009. LNCS, vol. 5823, pp. 505–520. Springer, Heidelberg (2009)
18. Schraefel, M., Karger, D.: The pathetic fallacy of rdf. In: International Workshop on the Semantic Web and User Interaction, SWUI 2006 (2006)
19. Shneiderman, B.: The eyes have it: a task by data type taxonomy of information visualizations. In: The Craft of Information Visualization: Readings and Reflections. Morgan Kaufmann Publishers Inc., San Francisco (2003)
20. Stuckenschmidt, H., Van Harmelen, F., De Waard, A., Scerri, T., Bhogal, R., Van Buel, J., Crowlesmith, I., Fluit, C., Kampman, A., Broekstra, J., Van Mulligen, E.: Exploring large document repositories with rdf technology: The dope project
21. Suchanek, F., Kasneci, G., Weikum, G.: YAGO: A core of semantic knowledge - unifying WordNet and Wikipedia. In: 16th International World Wide Web Conference (WWW 2007), pp. 697–706 (2007)
22. Teevan, J., Alvarado, C., Ackerman, M.S., Karger, D.R.: The perfect search engine is not enough: a study of orienteering behavior in directed search. In: CHI 2004: Proceedings of the SIGCHI conference on Human factors in computing systems, pp. 415–422. ACM, New York (2004)
23. Yee, K.-P., Swearingen, K., Li, K., Hearst, M.: Faceted metadata for image search and browsing. In: CHI 2003: Proceedings of the SIGCHI conference on Human factors in computing systems, pp. 401–408. ACM, New York (2003)