

Qsense

Learning Semantic Web Concepts by Querying DBpedia

Andrei Panu, Sabin C. Buraga and Lenuta Alboai

Faculty of Computer Science, Alexandru Ioan Cuza University, General Berthelot 16, Iasi, Romania

Keywords: Semantic Web, DBpedia, Linked Data, SPARQL, Knowledge, Learning Tool, Web Interaction.

Abstract: This paper describes *Qsense* – an educational and research Web application, developed to ease the interrogation of DBpedia for users intending to learn the most important concepts regarding the Semantic Web, especially to increase the knowledge about DBpedia in a pragmatic way. Additionally, by providing a compelling Web interface, *Qsense* offers the possibility to explore the DBpedia's ontology structure and to practice various SPARQL queries by expert and especially nonexpert users.

1 INTRODUCTION

Qsense is an educational and research Web project, offering an easy to use interface for querying DBpedia (DBpedia, 2013a) in order to learn to work with large amounts of semantic structured data. DBpedia is one of the most important thesauruses whose focus is “on the task of converting Wikipedia content into structured knowledge, such that Semantic Web techniques can be employed against it, asking sophisticated queries against Wikipedia, linking it to other datasets on the Web, creating new applications or mashups” (Auer et al., 2007).

Because DBpedia contains such a large amount of information, structured in a very complex manner, some people – especially the beginners like students enrolled at the courses concerning semantic Web, knowledge managements or related topics – may have trouble understanding and working with it. We decided to look for a common set of vocabularies, classes, and properties. The DBpedia project had at this time already developed the interlinking hub for the *Linking Open Data* projects, so it was the obvious choice to rely on DBpedia identifiers – according to the linked data initiative (Bizer et al., 2009).

DBpedia also provides structured data about concepts and the relations established between them. This aspect offers a strong foundation to automatically provide rich relationships between terms the user may need.

We initiated Qsense project with the following three main goals in mind:

- providing intuitive means for learning DBpedia

structure and, additionally, semantic Web technologies (data modeling via RDF, querying via SPARQL, knowledge modeling by using vocabularies, thesauri, and ontologies);

- getting the right information fast and easy without knowing any special query language;
- practicing certain SPARQL (W3C, 2013) queries for the ones who have the proper knowledge.

The paper continues with an overview regarding the semantic Web technologies and DBpedia. Section 3 details the *Qsense* requirements and features. The architecture of the project is presented by section 4, followed by several use cases. At the end, we provide conclusions and future directions of research.

2 DBpedia – A SUCCESSFUL SEMANTIC WEB PROJECT

The main purpose of Semantic Web is to enrich existing content on the Web with meaning in order for machines to understand it like humans to. According to (Bizer et al., 2009), “the first step is putting data on the Web in a form that machines can naturally understand, or converting it to that form”. This creates the *Semantic Web* – “a web of data that can be processed directly or indirectly by machines”.

To model available data on the Web, a standardized graph-based format is used: RDF (Resource Description Framework) (W3C, 2004). Each fact is denoted by a triple statement (subject, predicate, object), consisting of Web addresses (e.g., URLs). Also,

a query language was developed – SPARQL, which can be used to “express queries across diverse data sources, whether the data is stored natively as RDF or viewed as RDF via middleware” (W3C, 2013).

The huge amount of such linked data became one of the main showcases for successful community-driven adoption of Semantic Web technologies (Berners-Lee et al., 2001). It aims at developing best practices to opening up the data thesaurus on the Web, interlinking open data sets on the Web and permitting Web developers to make use of that rich source of information.

Additionally, the machine-understandable data made available and the existing practices and technologies provide benefits for both end-users and enterprises. Knowledge bases have an important role in enhancing the intelligence of Web and in supporting information integration (DBpedia, 2013a).

DBpedia is one of the main projects of the semantic Web, together with FOAF (Friend Of A Friend) (Brickley and Miller, 2000), SIOC (Semantically Interconnected Online Communities) (Breslin and Bojars, 2004), GoPubMed (Doms and Schroeder, 2005) and NextBio (NextBio, 2013). It extracts structured information from Wikipedia and makes this information accessible on the Web under an open licence terms. Wikipedia is an important source of information nowadays and many Web sites tend to provide information extracted from it, in a semantic and meaningful way. We use DBpedia to provide a common knowledge model – expressed by a controlled vocabulary and various well-known ontologies – and a stable service (endpoint), in order to give useful information to our users.

The data contained in DBpedia is quite impressive. As stated on the Web site (DBpedia, 2013a), the English version of the DBpedia knowledge base currently describes 3.77 million things, out of which 2.35 million are classified in a consistent ontology, in different categories like places, creative works, organizations, species, and many others. It provides localization in more than 100 languages and all these versions together describe 20.8 million things, out of which 10.5 million overlap (are interlinked) with concepts from the English DBpedia. The full dataset consists of 1.89 billion pieces of information stored as RDF triples.

Unfortunately, there are few initiatives concerning the querying and exploring such as big amount of structured data with benefits for end-users or neophytes. Thus, it is a real challenge to work with all this data, to understand DBpedia inner structure and to reuse it in personal future semantic Web work. In most cases, the existing applications require various

knowledge regarding semantic Web key technologies.

These facts raised the idea of our *Qsense* project. The DBpedia data set enables quite astonishing query answering possibilities against Wikipedia data. There is a public SPARQL endpoint over the DBpedia data set (DBpedia, 2013b). The endpoint is provided using Virtuoso (OpenLink, 2013) as the back-end database engine.

For making all kind of queries against DBpedia, we need a strong approach with respect to the Linked Data principles (Bizer et al., 2009), which means we need a method of publishing RDF data on the Web and of interlinking data between different data sources. Linked Data on the Web can be accessed using semantic Web browsers (or hyperdata software), just as the traditional Web of documents is accessed using traditional browsers. By using semantic Web technologies, users could navigate, browse, and visualize – in an intelligent manner – different data sources by following self-described RDF links. Additionally, these technologies are used to model the human knowledge as ontologies, to be further automatically managed – stored, queried, filtered, transformed, reused, etc. (Allemang and Hendler, 2011).

The information in Semantic Web is structured in triples in the form of (subject, predicate, object) – the RDF model. Altogether, the DBpedia dataset consists of more than 100 million RDF triples and because of this it is provided for download as a set of smaller RDF files.

To understand and manually interrogate this large amount of data is a very difficult task, especially for the beginner students studying the semantic Web or related fields.

3 REQUIREMENTS AND FEATURES

In the Semantic Web community, a very important feature is what someone is looking for and how the RDF document is structured. Learning structured information reduces the amount of work that users may have to perform. Also, people who have knowledge of RDF want to perform specific queries by using SPARQL language.

One possibility is to download a specific software, like Twinkle (Dodds, 2007) – which needs Java to be installed –, or to use a SPARQL endpoint – a public Web service.

For such semantic Web developers, we are presenting *Qsense* – a solution that is tailored to every very specific use case, taking into account the limited information about the term and the language users

are looking for. We made available some of the specific characteristics of DBpedia, constructing queries based only on the subject of interest. The user can find out more about the subject with just one click. The subject is represented by a keyword and, after the user types it, *Qsense* is able to automatically suggest – via Ajax (Asynchronous JavaScript And XML) – the possible predicates (properties) and to give the object value, if it exists in DBpedia. This is a straightforward SPARQL query and the very first step in *Qsense*.

Another goal of our developed project is to ease the understanding of the internal structure of semantic data in DBpedia. This means that the user starts to deal with the DBpedia ontology. For this part, the user should be able to learn the main notions regarding the ontological model expressed by RDF Schema and, furthermore, OWL language: “classes”, “sub-classes”, “properties”, and “instances”.

In the traditional approach regarding the ontology engineering, the experts add new data by carefully analyzing other existing ontologies and fitting their new concepts into the existing hierarchy. In the emerging Semantic Web approach, common users may inspect the DBpedia ontology before creating their own conceptual model – eventually, using a specific modeling tool and/or a (formal) specification. Once the user had learned a bit of DBpedia structure, (s)he will be able to build the desired ontology for a future semantic Web project. This is another important aim of the *Qsense* project.

By using *Qsense*, a user can also learn how to use Wikipedia information according to his/her needs and the application domain. We consider that DBpedia project is a very important open source semantic project and learning it by as many people as possible is a real gain for the semantic Web community.

Additionally, *Qsense* offers support for more experienced people who are used to build and work with RDF documents, too. The advanced users (e.g., data analysts, ontologists) can make any SPARQL query using our *Qsense* online service. If the user has his/her own RDF document and wants to make interrogations on it, (s)he has the possibility to directly upload it. This facilitates practicing SPARQL queries on local or remote RDF documents through various existing endpoints (i.e. RESTful Web services) – consult (Vandenbussche, 2013).

Other specific projects concerning DBpedia are:

- *DBpedia Mobile* – a location-aware (mobile) Web application designed for the use case of a tourist exploring a city; users are not able to perform their own queries (Becker and Bizer, 2008);
- *RelFinder* – an interactive tool for visualizing relationships established between the entities of a

knowledge base like DBpedia; unfortunately, by offering only a Flash UI, this application could not be used on mobile devices (Heim et al., 2009);

- *LodLive* – a visualization Web application able to explore DBpedia or Freebase; the users have access only to the English version of the knowledge base (Camarda et al., 2012).

4 Qsense ARCHITECTURE

Qsense consists of three main components that introduce the user into the world of Semantic Web and assists him/her in learning the basics by querying and exploring the DBpedia knowledge base. The system’s architecture is depicted in Figure 1.

4.1 DBpedia Interrogation Module

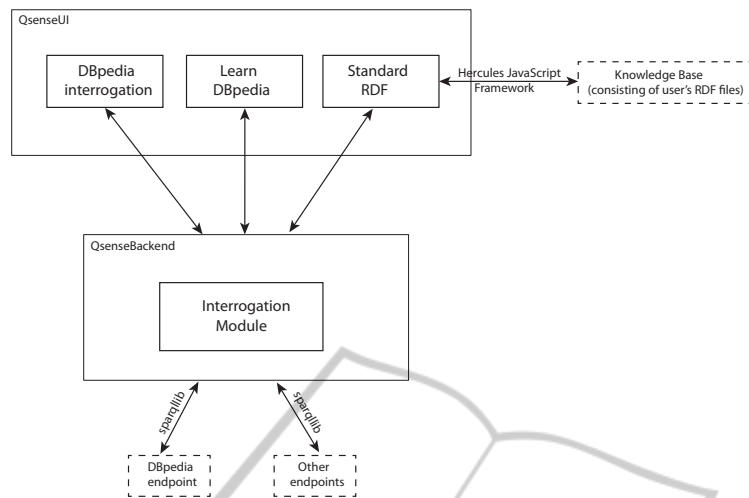
The first *Qsense* component performs DBpedia interrogations. The user has the ability to search for any keyword (e.g., the name of a concept and/or individual such as artefact, country, domain, person, etc.). The system automatically constructs a SPARQL statement which is sent to DBpedia, then the result is parsed and displayed in a friendly manner. The user does not see the interrogation, (s)he just enters the keyword and inspects the result, which represents, technically, the (abstract) information from the Wikipedia page related to the searched keyword.

To retrieve data from DBpedia, *Qsense* makes the following SPARQL query:

```
PREFIX res:http://dbpedia.org/resource/
PREFIX ont:http://dbpedia.org/ontology/
SELECT ?abstract WHERE {
  res:'. $keyword.' ont:abstract ?abstract.
FILTER
  langMatches
    (lang (?abstract), "' . $lang . '")
}
```

Beside extracting the basic information, for the first time, the system performs a SPARQL interrogation that retrieves all the keyword’s properties (the response is cached for further uses). Afterwards, using a JavaScript library, it automatically and asynchronously loads these properties in background. When the user starts typing any character in the specific field, several suggestions will appear – the user does not need to know in advance any property name.

After completing the field with as many words as the user wants, the system displays in a very simple and elegant way the value of those properties. The

Figure 1: *Qsense* General Architecture.

construct used to get all the properties is the following:

```
PREFIX res:<http://dbpedia.org/resource/>
SELECT DISTINCT ?property WHERE {
  { res:". $keyword." ?property ?obj . }
  UNION
  { ?obj ?property res:". $keyword." . }
}
```

4.2 Standard RDF Queries

Another *Qsense* component could be used by the persons knowing the SPARQL basics. It provides a simple editor that allows them to write their own queries and execute them on an existing SPARQL endpoint, a RDF resource identified by an URL or a local RDF file. This module uses the Hercules JavaScript Semantic Web Framework (Arielworks, 2008) that transforms the given RDF data into a NTriple format – via Apache Anything To Triples (any23) online service – and then parse it in order to get the SPARQL result. To minimize the response time, data is cached and asynchronously loaded.

4.3 Learn DBpedia Module

The users need to explore DBpedia without any knowledge of how it works or about SPARQL. This *Qsense* module provides an easy way to query DBpedia and also helps users to understand how it is structured. First, it automatically loads all the basic categories in a list, from where the user can choose any category (s)he wants.

When choosing a category, its subcategories are automatically discovered and loaded in an asyn-

chronous manner. The SPARQL statement that loads the *SubCategory* list is:

```
SELECT DISTINCT ?subject WHERE {
  ?subject
  rdfs:subClassOf
  <http://dbpedia.org/ontology/" .
  $scat . "> .
  ?instance a ?subject .
}
```

Using this approach, the tool can be easily adapted to use another knowledge base.

Now, the user can click on the View Category's Properties button in order to see all the properties existing in DBpedia for that category. *Qsense* performs the following parameterized query that searches all properties for the chosen category:

```
PREFIX ont:<http://dbpedia.org/ontology/>
SELECT DISTINCT ?property WHERE {
  ?instance a ont:" . $category ." .
  ?instance ?property ?obj .
}
```

Moreover, the user can obtain all the properties/instances for that particular subcategory. There is also possible to search for a keyword and see all its properties specified by DBpedia.

Qsense is implemented by using PHP as the back-end technology and HTML, CSS and JavaScript for the front-end. In order to perform SPARQL queries, we choose SPARQLLib (Gutteridge, 2012). By adopting responsive Web design principles, the application could be also used in the mobile context – a significant advantage for an e-learning tool.

5 USE CASES

Qsense is a Web instrument that gives the user the opportunity to interrogate DBpedia in a very simple and intuitive way, without knowledge about semantic Web technologies and DBpedia architecture.

First, the user can find out basic information about a certain topic. He/she just enters the keyword and the system automatically does the interrogation and provides the result which represents, technically, the abstract information from the Wikipedia page related to the searched keyword. The user can also select the language in which (s)he wants to receive the information. Our system supports all the languages in which the given information is found on Wikipedia.

Keyword:

Language: Done

Search keyword property:

- product
- primaryTopic
- isPrimaryTopicOf

Results: Semantic Web

abstract

The Semantic Web is a collaborative movement led by the World

Figure 2: Obtaining the values of certain properties regarding a specific concept (e.g., “Semantic Web”).

Moreover, when the user searches information about a certain topic, the system automatically retrieves all its properties and allows the user to complete a new field with as many desired properties, in order to find out more about the topic. The user input is assisted with an autocomplete feature, so the user does not need to know in advance the name of the properties – see Figure 2. The user searches for Semantic Web and general information about this topic are displayed. Afterwards, by using the properties of the keyword, the user can find out specific knowledge about a given concept.

Our project provides also a module where the user can easily learn how a part of DBpedia is structured, within a very simple, user-friendly interface. In the first place, the system automatically loads and shows all the basic categories of objects from DBpedia.

The user can choose a category from the list and see its (first level) subcategories. Using this simple approach, the beginners do not have to know the entire complex graph structure of the knowledge model. Instead of this, they could incrementally explore certain topics of interest.

For example, by selecting *Animal*, all its existing subcategories – provided by DBpedia – are automatically loaded in an asynchronous manner (Figure 3).

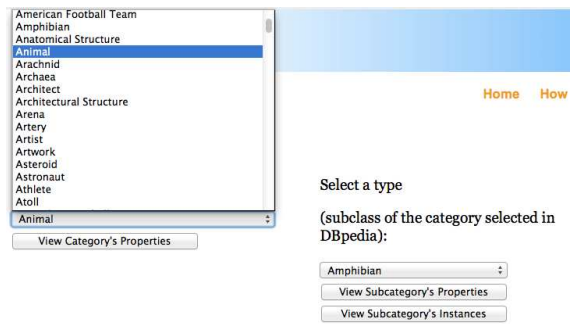


Figure 3: Browsing DBpedia (sub)categories.

Furthermore, by simply clicking one button, the user sees all the properties existing in DBpedia for a category. Adopting the “learning by example” approach, a student could learn about the existing list of properties (e.g., expressed by RDF Schema or OWL) attached to a given class (concept). For example, the *Bird* concept has a relatively long list of common/specific properties such as *type*, *synonym*, *comment*, *classis*, *genus*, etc.

There is also possible to see all the instances of a (sub)category, just by clicking the *View Subcategory's Instances* button. In the case of *Brain* category, there are shown *Cerebellum*, *Neocortex* and so on – see Figure 4.

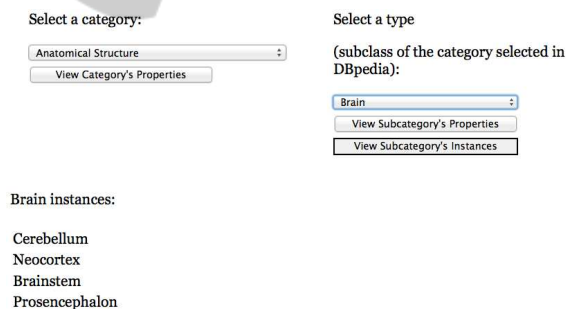


Figure 4: DBpedia instances for a chosen subcategory.

Finally, the user has the possibility to type a keyword denoting an item of interest and browse all its properties. In this way, we encourage him/her to try to learn and understand better how DBpedia works, without any previous knowledge about SPARQL or other semantic Web technologies.

For example, if the user enters *Tim Berners-Lee*, then (s)he will obtain a list of properties to be further explored (e.g., *sameAs*, *founder*, *inventor*, *leaderName*, *KeyPeople*, and many others).

Additionally, beginners could access a help section that includes a video demo (Cristea and Timofciuc, 2013) showing the most important features.

6 CONCLUSIONS AND FUTURE WORK

We introduced a straightforward, intuitive and easy to use (mobile) Web application which allows users to (transparently) make queries on DBpedia and learn about its structure. It does not require an account or additional software installed besides a Web browser.

The key difference that we see between other Web sites/applications that are using DBpedia information and our approach is that expert and especially non-expert users can perform queries against such a large amount of structured data with or without any knowledge of semantic Web languages/technologies, such as RDF, SPARQL, linked data, vocabularies, etc.

Various *Qsense* versions were tested by a number of about 30 students during the lab classes regarding semantic Web discipline. Currently, we are preparing to launch the first public beta version of our developed application.

Qsense is designed to be a learning tool that can be easily extended. We intend to add support for more complex queries and to link multiple DBpedia resources through common properties. This feature will allow users, by only selecting items and filling some fields, to link data and to obtain deduced results via a specific reasoner.

Additionally, a visualization module is planned to be included, eventually using the VOWL notation (Negru and Lohmann, 2013).

Another useful feature seems to be a cloud-based recommending system about things the user may be interested, such as books, geographical information, movies, activities and so on, following the directions exposed by (Luca and Buraga, 2009).

ACKNOWLEDGEMENTS

We are grateful to *Ana Maria Timofciuc* and *Alexandra Cristina Cristea* for their important support regarding the project development and testing.

REFERENCES

- Allemand, D. and Hendler, J. (2011). *Semantic Web for the Working Ontologist: Effective Modeling in RDFS and OWL*. Morgan Kaufmann.
- Arielworks (2008). Hercules Semantic Web Framework. <http://hercules.arielworks.net/>.
- Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., and Ives, Z. (2007). Dbpedia: A Nucleus for a Web of Open Data. *The Semantic Web*, pages 722–735.
- Becker, C. and Bizer, C. (2008). DBpedia Mobile: a Location-aware Semantic Web Client. *Proceedings of the Semantic Web Challenge*, pages 13–16.
- Berners-Lee, T., Hendler, J., Lassila, O., et al. (2001). The Semantic Web. *Scientific american*, 284(5):28–37.
- Bizer, C., Heath, T., and Berners-Lee, T. (2009). Linked Data - The Story So Far. *International Journal on Semantic Web and Information Systems (IJSWIS)*, 5(3):1–22.
- Breslin, J. and Bojars, U. (2004). Semantically-Interlinked Online Communities (SIOC) Initiative. <http://sioc-project.org/>.
- Brickley, D. and Miller, L. (2000). Friend of a Friend (FOAF) Project. <http://www.foaf-project.org/>.
- Camarda, D. V., Mazzini, S., and Antonuccio, A. (2012). LodLive, Exploring the Web of Data. In *Proceedings of the 8th International Conference on Semantic Systems*, pages 197–200. ACM.
- Cristea, A. and Timofciuc, A.-M. (2013). Qsense Video Demo. https://www.youtube.com/watch?v=FBGÜqs_DmQ.
- DBpedia (2013a). DBpedia Knowledge Base. <http://wiki.dbpedia.org/About>.
- DBpedia (2013b). DBpedia SPARQL Endpoint. <http://dbpedia.org/sparql>.
- Dodds, L. (2007). Twinkle: A SPARQL Query Tool. <http://www.ldodds.com/projects/twinkle/>.
- Doms, A. and Schroeder, M. (2005). GoPubMed: Exploring PubMed with the Gene Ontology. *Nucleic Acids Research*, 33(Supplement 2):W783–W786.
- Gutteridge, C. (2012). RDF Library to Query SPARQL from PHP. <http://graphite.ecs.soton.ac.uk/sparqllib/>.
- Heim, P., Hellmann, S., Lehmann, J., Lohmann, S., and Stegemann, T. (2009). Relfinder: Revealing Relationships in RDF Knowledge Bases. In *Semantic Multimedia*, pages 182–187. Springer.
- Luca, A.-P. and Buraga, S. C. (2009). Enhancing User Experience on the Web via Microformats-Based Recommendations. In *Enterprise Information Systems*, pages 321–333. Springer.
- Negru, S. and Lohmann, S. (2013). A Visual Notation for the Integrated Representation of OWL Ontologies. In *Proceedings of the 9th International Conference on Web Information Systems and Technologies*. SciTePress.
- NextBio (2013). NextBio Platform. <http://www.nextbio.com/>.
- OpenLink (2013). Virtuoso Universal Server. <http://virtuoso.openlinksw.com/>.
- Vandenbussche, P.-Y. (2013). SPARQL Endpoint Status. <http://labs.mondeca.com/sparqlEndpointsStatus/>.
- W3C (2004). Resource Description Framework (RDF). <http://www.w3.org/RDF/>.
- W3C (2013). SPARQL Query Language for RDF. <http://www.w3.org/TR/sparql11-overview/>.