QwwwQ:
Querying Wikipedia without writing queries

Massimiliano Battan[1] and Marco Ronchetti[1]

[1] DISI, Università degli Studi di Trento
38123 Povo di Trento, Italy
massimiliano.battan@gmail.com,marco.ronchetti@unitn.it
http://latemar.science.unitn.it

Abstract. Wikipedia contains a wealth of data, some of which somes in a structured form. There have been initiatives to extract such structured knowledge, incorporating it in RDF triples. This allows running queries against the body of knowledge. Unfortunately, writing such queries is an unfeasible task for non-technical people, and even those who are familiar with the SPARQL language face the difficulty of not knowing the logical data schema. The problem has been attacked in many ways, mostly by attempting to provide user interfaces which make it possible to graphically navigate the see of RDF triples. We present an alternative user interface, which allows users to start from a Wikipedia page, and to simply express queries by saying "find me something like this, but with these properties having a value in the [A-B] range".

1 Introduction

The Sematic Web effort has opened the way to the Web of Data initiative. Based on the notion of RDF triplets, every single chunk of data is connected to something else. These connections transform simple data into precious information, and today Linked Data offers a wealth of such. Sometimes however too much information is equal to no information: we need to be able to find and harness information that is relevant to us and to our goals. Having a gigantic database is of no use if we do not know its schema and what the relations mean, but if we know, we can write queries and extract knowledge that is relevant to us. Likewise, making sense of Linked Data without some base knowledge is extremely difficult. This problem has spawned a whole research field: exploration and visualization of Web of Big Linked Data. A review of it has been recently presented by Bikakis and Sellis [1]. Most responses are given in terms of graph visualization and exploration, with no less than 20 different systems having been proposed. Although such approach is certainly useful and valuable, bringing the data to the final, generic user requires something more. Creating a generic user interface that makes it possible to easily find and extract relevant information for non-technical and inexperienced users is an extremely difficult task. Restricting to a subset of data may help, even though is does not respond to the more general problem. A particularly relevant subset of Linked Data is the one collected by the DBPe-

The present is a self-archived post-print version of the paper:
Battan, M. and Ronchetti, M. QwwwQ: Querying Wikipedia Without Writing Queries, in "*Web Engineering: 16th International Conference, ICWE 2016, Lugano, Switzerland, June 6-9, 2016. Proceedings Web Engineering" - Volume 9671 of the series Lecture Notes in Computer Science* pp 389-396, 2016
The published source is available at:
http://link.springer.com/chapter/10.1007%2F978-3-319-38791-8_24
DOI: http://dx.doi.org/10.1007/978-3-319-38791-8_24

dia by exploiting the structured part of information, which is collected by the Wikipedia project. An example of such an approach is the Spacetime visualizer [2], where the final user can implicitly write queries about those DBPedia data, which have geographic and temporal attributes. They can be queried, and responses are returned in a context-rich interface. For instance, the system allows finding all the poets born in France in the XVII century, or all the battles fought during World War II, visualizing their geographic location in a temporal sequence. Along with this line, we present a novel approach, which deploys the user context to allow her/him to express queries against DBPedia [3] in a simple way and without limitation to specific data types. The interface is presented as a Chrome plug-in that can be activated on any Wikipedia page. The page define the context, and the user can start from the structured data present in it to express queries by analogy, in the form "find me something like this, but with such and such different parameters". No need of knowing any query language is needed, empowering hence generic, non-skilled users.

In this paper we first set the ground, by recalling the DBPedia project and its relations with Wikipedia, then we present our approach, and finally discuss and conclude.

2 From Wikipedia to DBPedia

DBPedia project offers a structured and machine-readable representation of a subset of Wikipedia knowledge. Started in 2007 with its first release, the dataset is currently at the base of the Linked Data effort, connecting and linking to almost any available dataset over the web such as Freebase, Linked MDB1, Linked Geo Data2, Proside and many others. It allows performing semantic queries over entities and relationship, which are directly connected to Wikipedia itself and other dataset.

Information is written in Wikipedia in natural language, so how can DBPedia extract it and put it in a machine-readable form?

The main trick is that a part of the information present in Wikipedia pages is actually coded into the so-called infoboxes. An infobox is a template used to collect and present a subset of information about its subject. The template contains a list of attributes: for instance, the "settlement" infobox, which is used to describe populated areas, lists properties such as "Name", "image_skyline", "governing_body", "area_total_sq_mi", "elevation_ft" etc. A Wikipedia author, who wants to create a page about a city, will include this template and add all the known values for the attributes. In this way content-uniformity is improved, as well as content-presentation. In fact, originally infoboxes were mostly devised for page layout purposes, the original intent being to improve the appearance of Wikipedia articles. When the MediaWiki software parses the document, infobox(es) are processed by a template engine, which applies to it a web document and a style sheet. Such design allows separating content from presentation, so the template rendering can be modified without affecting the information. Usually, infoboxes appear in the top-right corner of a Wikipedia article in the desktop view, or at the top in the mobile view.

Although the infobox idea was mainly related to presentation issues, it had the effect of including some structured data into a relevant portion of Wikipedia pages: about 44.2% of Wikipedia articles contained an infobox in 2008 [4]. The infobox is usually

compiled by the page authors, but there have been attempts to populate them by automatically extracting knowledge from the natural text contained in the page [5].
DBPedia uses machine-learning algorithms to extract structured data from infoboxes. It creates triples consisting of a subject (the article topic), predicate (the parameter name), and object (the value). Each type of infobox is mapped to an ontology class, and each property (parameter) within an infobox is mapped to an ontology property.

DBpedia dataset is regularly (but not frequently) updated, syncing it with the Wikipedia source. There are localized versions of it (so as there are versions of Wikipedia in many languages). Any entity contained in DBpedia (say "the thing"), is associated to an unique URI reference in the form http://dbpedia.org/resource/thingName, where things must always coincide with the URL of the relative Wikipedia article which has the form http://en.wikipedia.org/wiki/thingName; in this way it is possible to have a simple mapping between DBpedia and Wikipedia. Any thing can have various properties, some of which are mandatory. Every DBPedia resource is always described by a label, a link to the relative Wikipedia page, a link to an image representing it (if any) and a short and long abstract. If the resource is available for different Wikipedia languages, corresponding abstracts and labels are added accordingly to the language.

The knowledge contained in DBPedia is far from being complete and fully correct, as it is derived from infoboxes, which are far from being perfect: for instance there is no control on the inserted values. It may happen that pages sharing the same infobox structure show strings, which include numeric values together with a symbol expressing the measure units (and sometimes different units are used), so that it may be difficult to (automatically) make sense of the value. In other cases, different textual expression are used to indicate the same entity, as for instance when the "ruling party" is associated with values such as "Democratic Party", "Democrats" or "D.P.": although for humans the three expression may carry the same meaning, the same cannot hold for automatic processing. It may also happen that a field is left blank by the authors, either because they do not know, or because the forget filling it. In property value sometimes there are even comments! Lacks of standards, typing errors etc. restrict the usefulness of infoboxes as source of machine-readable data.
In spite of this problem, which limits the completeness and usefulness of the harvested data, DBPedia offers the possibility to navigate the data and to express queries, which can extract "hidden" information from the Wikipedia body. For instance, it is possible to find all settlements in a give region, and to select the subset having a population within certain bounds. Further, from the result set we can select only those location being ruled by a certain political party: "find all the towns in Tuscany having more than 10.000 inhabitants, which are ruled by the Democratic Party". Such information is contained in Wikipedia, but extracting it by hand is extremely difficult and time-consuming, while asking that to DBPedia can be "easily" done. Easily, provided that one is familiar with SPARQL, and that one knows the underlying data structure! Hence, not only unskilled, non-technical people are excluded because they are unable to use query languages, but also SPARQL expert may not find the task easy, unless they are familiar with the (gigantic) schema. In this case, rather than a database schema, one needs to be familiar with the ontology, knowing its classes and properties. The focus on this last problem has driven the research towards "exploration tools", which in most cases are graph explorers. LODlive [6] is one such tool: In order to use

the system, user have to insert a keyword or a URI to be searched on a selected DBPedia endpoint; if the resource is present, a graph will be drawn with a single node representing the found DBPedia concept. The node is surrounded by small circles, which represent the properties associated to it. For each concept it is then possible to open different types of panels showing a summary of all the properties, abstracts, external links for the concept and its ontology classes. Many similar instruments have been presented in literature: discussing them here would be too long, and hence we refer the reader to the comprehensive survey recently written by Bikakis and Sellis [1].

3 QwwwQ (pron.: "qiuck")

In this scenario, we attempted to devise a quick method for allowing generic users to perform queries on Wikipedia without knowing query languages and ontology structure. It is somewhat similar to the "Query by example" paradigm suggested by Zloof over 40 years ago [7] and later refined by the same author [8]. Zloof's goal was remarkably similar to ours: the idea was to enable the non-professional user who has little or virtually no computer or mathematical background to extract information from the (at that time young) relational data model. Users could provide an example (giving one value per column) where some of the data were considered invariant, and other were variables. Hence, in a table having "TYPE", "ITEM", "COLOR" and "SIZE" columns the user could specify ITEM=P.PEN and COLOR=RED. The underline values meant "variable", and the not-underlined ones meant "constant", so that the query would return a set of items, having the property of being "RED": e.g. {LIPSTICK, PENCIL}. Specifying ITEM=PEN and COLOR=P.RED would return the set {GREEN, BLUE}, i.e. all the available colors for pens. The attributes "TYPE" and "SIZE" are considered irrelevant in the query. A psychological study demonstrated experimentally the ease and accuracy with which nonprogrammers learned and used this Query By Example language [9].

In our case, the user starts from an example, instantiated in a Wikipedia page, which defines the context: for instance s/he could start from the "Berlin" page. The unexpressed statement is "I am interested in (German ?) towns – or in (European ?) state capitals – or in German states (Länder)", as Berlin is both a city and a state, and also an European Capital. Next, the user makes the intention explicit of finding "something like this": s/he does so by invoking QwwwQ, which comes in the form of a Chrome plug-in. QwwwQ presents an interface, where all the known properties of Berlin are shown, in the form of a table. The properties are what DBPedia knows about Berlin, which in turn derives from the Wikipedia page infobox.

Fig. 1 shows such example. The right hand side shows a part of the infobox of the "Berlin" Wikipedia page. On the upper left side, a portion of pop-up called by the Chrome extension shows some of the fields present in the infobox, in the form of attribute-value pairs. The user can select the ones s/he deems relevant: for instance "population" and "leader_party". The selection is done by ticking the row. A condition is expressed by selecting an operator: for numbers =, >, ≥, ≤, <; for strings "starts

with", "ends with", "contains", "equal". For numbers it is possible to express also ranges, and operators are provided also for dates.

The example in Fig.1 expresses the query: we're looking for items similar to the current one (Berlin), having a population larger than five millions, and lead by the SPD party. We do not put any restriction on the other (non-selected) fields.

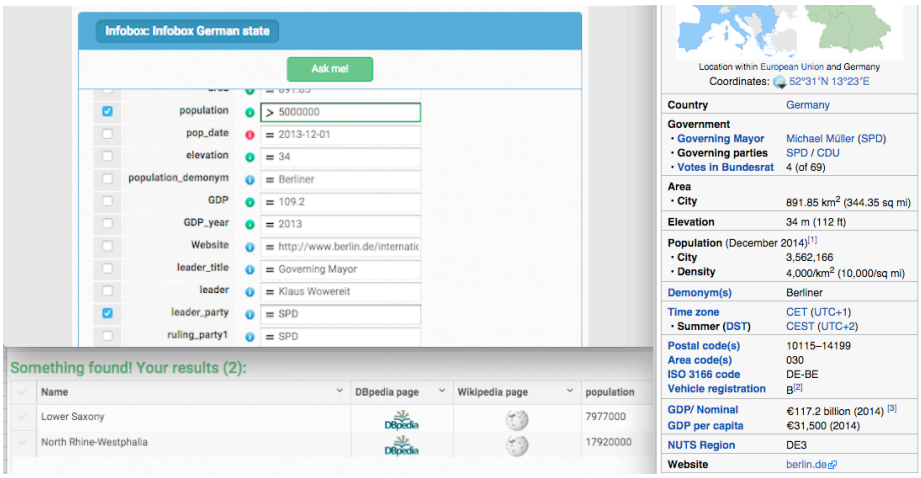

Fig. 1. On the right hand side, a part of the infobox of the "Berlin" Wikipedia page. On the upper left hand side, the pop-up for expressing the query. On the lower left hand side, the results.

On the lower right hand side, the found results are shown: it's Lower Saxony and North Rhine-Westfalia. The variable(s) properties (in this case only the population, as the party is fixed) are shown in a table along with the results, and the table can be ordered on any of the fields. Links also allow opening the corresponding Wikipedia or DBPedia page for the found results (e.g. for Lower Saxony in this case).

Had we have started from "Elizabeth I of England" and asked for House= "House of Tudor", and Religion=anything, we would had found the result set shown in the upper part of Fig.2 (5 results). These results can be filtered on a pop-up, using two different options: we can use DBPedia similarity class or Wikipedia categories. The applicable classes, given the starting example we have chosen, would be: "Royalty", "Person" and "Agent". In the example the available categories are many (it depends on how the Wikipedia page has been catalogued by its authors), and one possible choice would be "16-century women". Such choice and its results are shown in the lower part of Fig.2, where also the results are shown: they are now reduced to only 3 values.
Other examples of questions, which can be easily asked to the system include:

1. *Starting from "Avatar (Movie)" page, give me all the films directed by James Cameron where James Horner is the music producer.*
2. *Starting from the "NCSOFT" page, give me all the organization in Computer and Video Games industry founded after January 2000 with Headquarters located in South Korea.*

3. *Starting from the Italian town "Trento" page, give me all the Italian cities that falls in the 4th seismic zone, have an elevation greater than 200 m, a total population larger than 100.000, and whose patron day falls in the first half of the year.*

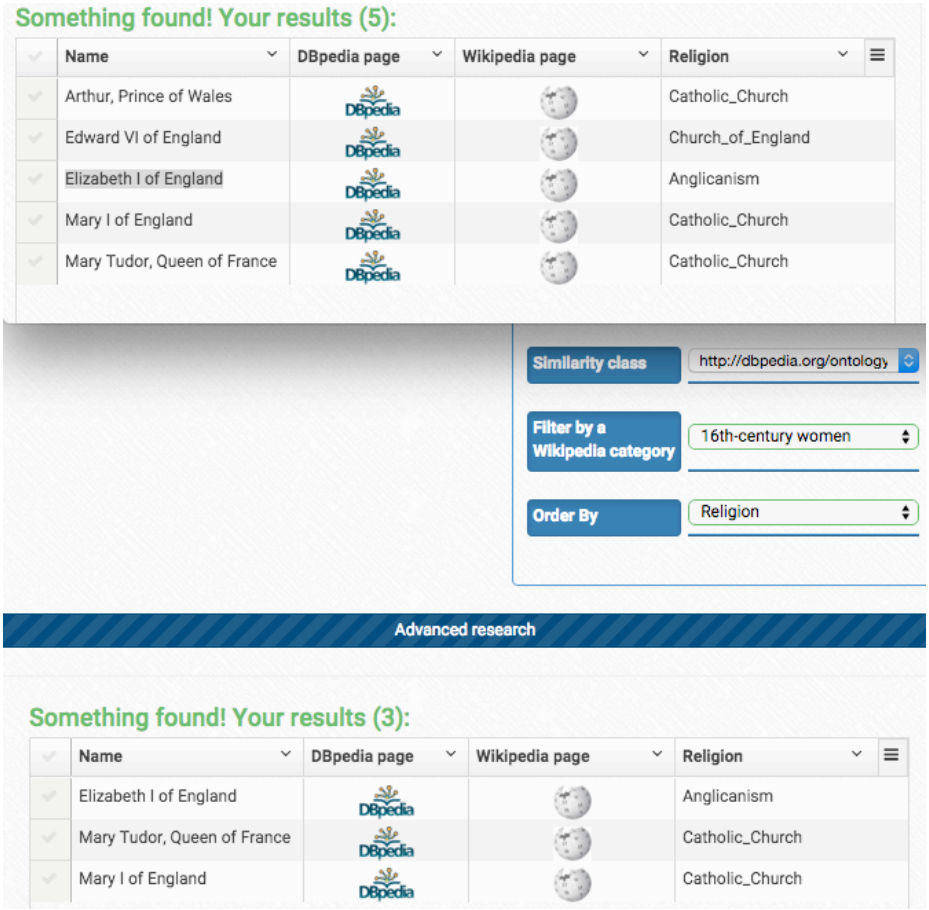

Fig. 2. On the upper part, the result set of the unfiltered query. In the middle, the filtering by Wikipedia category. In the lower part, the filtered results are shown.

3.1 Implementation

The system is composed by four components:
1) *The core library*: a library written in Java which is responsible of implementing all the features of the system. It is composed of a **Wikipedia Parser module** and a **SPARQL query builder** module, along with several utility classes. The library is responsible to provide the interconnection between DBpedia and the system and be-

tween Wikipedia and the system. All the other components never contact directly these services but they rely on the library in order to access either DBpedia or Wikipedia. Wikipedia is replicated on the server for efficiency reasons (first of all to have access to the wikicode, from which the templates can be easily extracted), and to make sure that we refer to the same version of Wikipedia, on which DBPedia is based.
2) RestFUL web services. A component that exposes the core library services, so that they can be called via Ajax from the client component.
3) AngularJS application: implements the client side component. In order to develop a Chrome Extension it was mandatory to use Javascript. The GUI had to to meet Chrome plugin requirements and usability. The application has been developed using the Google AngularJS framework.
4) Chrome extension: to integrate the Javascript application in the Chrome Browser, a a Chrome plugin was developed, with the duty of detecting user navigation and bootstrapping the JS application with the correct parameters.

A more detailed account of the implementation will be given elsewhere, and can be found a Master Thesis [10].

4 Discussion and conclusions

We proposed a system that allows inexperienced, non-technical users to query DBPedia, starting from a generic Wikipedia page, expressing interrogations of the type "find something like this, constrained by these parameters". The interface is fully integrated, as it presents itself as a pop-up on a Wikipedia page, and hence enhances the Wikipedia initiative with a set of functionalities, which were missing.

The system is not without problems, which mostly stem from Wikipedia shortcomings. We already mentioned the problems that infoboxes exhibit, and which in good part derive from the poor quality of data control. Another problems comes from Wikipedia categories, which we use for optional filtering of the results. They present several issues: for instance they are not an acyclic graph. Also, not all pages correctly include the suitable categories. For instance, Hanover belongs to the "University towns in Germany", while Berlin does not (which is obviously wrong). Not all pages include the suitable infoboxes: for instance Berlin is both a town and a state (Land), and its page presents only the "State of Germany" infobox, while Hanover presents four infoboxes: "German location", "Historical population", "Weather box" and "Geographic location". This makes the two "things" incomparable. The situation is similar if we examine the DBPedia ontology classes: Hanover is "Town", "Place", "Populated place", "Settlement", while Berlin is "Administrative Region", "Place", "Populated place" and "Region" (so here we have a partial overlap).

Although these shortcomings are general, and affect the whole DBPedia initiative, QwwwQ certainly suffers from them. Improving the QwwwQ Parser Module, e.g. by automatically recognizing measure units so as to convert to a standard would not help, as the data would not match the ones contained in DBPedia. The same happens for specific types of property values such as lists, because user is not able to really make

use of them in a query. A possible solutions would be recognizing unusable properties and remove them from the parsing result, so that the user then would never use them when building a query: it is better to have less options having the guarantee that they work rather than a wider spectrum of uncertain possibilities. Of course, the silver bullet would be a better quality control in the Wikipedia initiative.

QwwwQ has not yet undergone a validation with a large enough number of users, which we plan to run soon. The system has been tested with English and Italian versions of DBpedia and Wikipedia and it has been implemented to support multi-language, however parsing issues may arise when dealing with languages other than the ones supported at the moment. Another improvement, which we plan to tackle in future, is the possibility to express the equivalent of a join operation, i.e. to transvers DBPedia relations.

References

1. Bikakis, N., Sellis, T.: Exploration and Visualization of Web of Big Linked Data: A Survey of the State of the Art. Workshop Proceedings of the EBTD/ICDT 2016 Joint Conference, Bordeaux, France, arXiv:1601.08059v1 [cs.HC] (2016)
2. Valsecchi, F., Ronchetti, M.: Spacetime: a two dimensions visualization engine based on Linked Data, in Advances in Semantic Processing,. 8th International Conference on, (SEMAPRO 2014), Red Hook, NY 12571: IARIA, 2014, p. 8-12. - ISBN: 9781634392631.
3. Christian Bizer. Dbpedia - a crystallization point for the web of data. Web Semantics: Science, Services and Agents on the World Wide, 7(3):154–165 (2007).
4. Baeza-Yates, Ricardo; King, Irwin, eds.: Weaving services and people on the World Wide Web. Springer. ISBN 9783642005695. LCCN 2009926100 (2009)
5. Lange, Dustin; Böhm, Christoph; Naumann, Felix: Extracting Structured Information from Wikipedia Articles to Populate Infoboxes. Technische Berichte des Hasso-Plattner-Instituts für Softwaresystemtechnik an der Universität Potsdam, Hasso-Plattner-Institut für Softwaresystemtechnik Potsdam (Universitätsverlag Potsdam). ISBN 9783869560816 (2010)
6. Mazzini, S., Camarda, D.V., Antonuccio, A.: Lodlive, exploring the web of data. In Proceedings of the 8th International Conference on Semantic Systems., pages 197–200 (2010).
7. Zloof, M.M.: Query by example. In Proceedings of the May 19-22, 1975, national computer conference and exposition (AFIPS '75). ACM, New York, NY, USA, 431-438. DOI=http://dx.doi.org/10.1145/1499949.1500034 (1975)
8. Zloof, M. M.: Query-by-example: A data base language. IBM systems Journal 16.4 324-343 (1977)
9. Thomas J.C., Gould J.D: 1975. A psychological study of query by example. In Proceedings of the May 19-22, 1975, national computer conference and exposition (AFIPS '75). ACM, New York, NY, USA, 439-445. DOI=http://dx.doi.org/10.1145/1499949.1500035 (1975)
10. Battan, M.: Querying Wikipedia: A Linked Data System for exploring it. Università di Trento, Italy, Master Thesis (2015)