-This is an electronic version of an Article published in:
36th International ACM SIGIR Conference, Dublin, Ireland, 2013
-The final publication is available at:
http://dl.acm.org/citation.cfm?doid=2484028.2484200

X-ENS: Semantic Enrichment of Web Search Results at Real-Time

Pavlos Fafalios and Yannis Tzitzikas
Institute of Computer Science, FORTH-ICS, GREECE, and
Computer Science Department, University of Crete, GREECE
{fafalios,tzitzik}@ics.forth.gr

ABSTRACT

While more and more semantic data are published on the Web, an important question is how typical web users can access and exploit this body of knowledge. Although, existing interaction paradigms in semantic search hide the complexity behind an easy-to-use interface, they have not managed to cover common search needs. In this paper, we present X-ENS (eXplore ENtities in Search), a web search application that enhances the classical, keyword-based, web searching with semantic information, as a means to combine the pros of both Semantic Web standards and common Web Searching. X-ENS identifies *entities of interest* in the snippets of the top search results which can be further exploited in a faceted interaction scheme, and thereby can help the user to limit the - often very large - search space to those hits that contain a particular piece of information. Moreover, X-ENS permits the exploration of the identified entities by exploiting semantic repositories.[1]

Categories and Subject Descriptors

H.4.0 [**Information Systems Applications**]: General

Keywords

semantic post processing of search results; entity mining; Linked Data

1. INTRODUCTION

While more and more semantic data are published on the Web, the question of how typical web users can access this body of knowledge becomes of crucial importance. There is a growing amount of research on interaction paradigms that allow end users to profit from the expressive power of Semantic Web standards. These paradigms range from keyword search (e.g. [2]), faceted browsing and exploration (e.g. [1]) to natural language question answering (e.g. [3]). Although most hide the complexity behind an easy-to-use interface, regular users do not take advantage of using them. Their usefulness in comparison to common web search engines (especially for regular users) has not been proved, mainly because they aim at providing the user with (linked) data and relationships between these data, instead of providing links to web pages.

[1] A deployment of X-ENS is available at http://139.91.183.72/x-ens/.

SIGIR'13, July 28–August 1, 2013, Dublin, Ireland.
ACM 978-1-4503-2034-4/13/07.

Figure 1: An indicative screen dump of X-ENS.

To address this issue we focus on *enriching* the classical web search engines with *Linked Open Data* (LOD), as a means to combine the pros of both semantic web standards and common web searching. X-ENS is a *web application* that identifies at *query-time* entities of interest (i.e. LOD of a particular class, e.g. Persons, Locations, Organizations, etc.) in the snippets of the top results returned by a search system and allows the user to explore at *real-time* their properties. The entities are exploited in a faceted and session-based interaction scheme, allowing the user to narrow the search space to those hits that contain a particular piece of information. Figure 1 depicts an indicative screen dump.

2. SYSTEM DESCRIPTION

The process. We focus on a *dynamic* approach where no pre-processing of the resources has been done. Specifically, the user a) submits a *keyword query* through X-ENS's web interface, b) X-ENS fetches the *top results* of the same query from the underlying search system, c) it applies *entity mining* in the snippets of the results, d) *ranks* the discovered entities, e) *groups* them according to the categories they belong to, f) *ranks* the categories, and finally g) *presents* the categorized entities in a *faceted* interface. At that point, the user has many (session-based) options for interacting with the search system that we analyze in Section 2.1.

Discovering entities in the results. We currently use GateAnnie[2] for entity mining. In our setting it takes as input a set of document snippets, specifically those of the

[2] http://gate.ac.uk/ie/annie.html

top-L hits of the query answer, and it returns as output a set of entity lists (one list for each category). We have automated the procedure of adding a new category of entities in GateAnnie, i.e. we can easily configure the entity names that are interesting for the application at hand (e.g. LOD of a particular type). As we will see later (Section 2.2), for defining the *entities of interest* we can exploit any semantic repository that is accessible via a SPARQL endpoint.

Ranking of entities. The user submits a keyword query q, and let A be the set of the top-L hits (e.g. $L = 200$) returned by the underlying search system. For an $a \in A$, let $rank(a)$ be its position in the answer (the first hit has rank equal to 1, the second 2, and so on). We apply entity mining in A, get a set of entities E, and rank E according to the formula: $Score(e) = \frac{\sum_{a \in docs(e)} ((|A|+1) - rank(a))}{\frac{|A|(|A|+1)}{2}}$, where $docs(e)$ denote the elements of A in which an entity e has been identified. We can see that the entities occurring in the top hits are promoted, i.e. we exploit the ranking of the documents. The rational behind this ranking formula is that the top hits in the ranked list probably contain more useful entities that the last hits since they are considered "better" results.

Ranking of categories. For a category c, if $inst(c)$ denotes the entities that fall in c, we rank the categories according to the formula: $Score(c) = \sum_{e \in inst(c)} Score(e)$. We can see that the categories which contain the more highly scored entities are promoted.

2.1 Interaction Model

Faceted search-like exploration of the results. The results of entity mining (LOD grouped in categories) are visualized and exploited according to the *faceted exploration* interaction paradigm [4]: when the user clicks on an entity, the hits are restricted to those that contain that entity (Figure 1A). Specifically, the user is able to gradually select entities from one or more categories and refine the answer set accordingly (the mechanism is session-based). If such selections belong to the same category, they have disjunctive (`OR`) semantics and if they belong to separate categories they have conjunctive (`AND`) semantics.

On-click semantic exploration of LOD. There are already vast amounts of structured information published according to the principles of LOD. The availability of such datasets enables the *dynamic* enrichment of the identified entities with more information about them. In this way the user not only can get *useful information* about an entity without having to submit a new query, but he can also start *browsing* the entities that are linked to that entity. However, a question is which LOD datasets to use. We identify and specify an appropriate dataset for each category of entities. For example, *DBpedia*[3] is appropriate for multiple categories such as Organizations, Persons, Locations, etc.

Running one (SPARQL) query for each entity would be a very expensive task, especially if the system has discovered a lot of entities. For this reason, we offer this service *on demand*. Specifically when the user clicks on the small icon at the right of an entity, the system at that time collects more information about that entity which is visualized in a popup window (Figure 1B). As we will see in Section 2.2, the user is able to define a SPARQL endpoint and a SPARQL template query for each distinct category of entities.

[3]http://dbpedia.org/

2.2 Configurability

We give particular emphasis on the configurability of `X-ENS`. The administrator can specify various parameters of `X-ENS` through a configuration page (Figure 1C).

Specifying the underlying search system. We can define the *underlying search system* (e.g. Bing) by giving an *OpenSearch*[4] description document. *OpenSearch* is a collection of simple formats for the sharing of search results and the *OpenSearch description document* format can be used to describe a search engine. Nevertheless, we could plug any search system by creating the appropriate results parser.

Adding a new category of entities. We are able to add a new category of entities by giving a *category title* and a *list of words/phrases*. The list can be loaded by running a SPARQL query over a knowledge base that offers a SPARQL endpoint. For example, we can run a SPARQL query over DBpedia's SPARQL endpoint that returns a list of all objects of `rdf:type dbp-ont:TennisPlayer` and thereby offer the ability to explore Tennis Players in the search results.

Specifying the underlying knowledge bases. We are able to define how to *semantically explore* an identified entity by giving a *SPARQL template query* and a *SPARQL endpoint* for each category of entities that we want to offer entity exploration. The SPARQL template query must contain the character sequence <`ENTITY`> (including the < and >). When a user asks for more information about an entity, we read the template query of the category in which the selected entity belongs and replace each occurrence of <`ENTITY`> with the entity's label name.

3. CONCLUSION

We described `X-ENS`, a web search application that enhances at *real-time* the classical web searching with semantic information, as a means to combine the pros of both semantic web standards and common web searching. `X-ENS` is applicable to any search system that offers textual results, it does not require any pre-processing and it does not use any caching scheme. It exploits at *real-time* semantic repositories (i.e. the LOD cloud) for both configuring the *entities of interest* and further exploring their properties. The average response times of `X-ENS` highly depend on the underlying search system, the knowledge bases that we exploit (SPARQL endpoints) and the SPARQL queries that we run.

Acknowledgement. Work done in the context of *iMarine* (FP7 Research Infrastructures, 2011-2014) and *MUMIA* COST action (IC1002, 2010-2014).

4. REFERENCES

[1] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. Ives. Dbpedia: A nucleus for a web of open data. *The Semantic Web*, pages 722–735, 2007.

[2] A. Hogan, A. Harth, J. Umbrich, S. Kinsella, A. Polleres, and S. Decker. Searching and browsing linked data with SWSE: the semantic web search engine. *Web Semantics: Science, Services and Agents on the World Wide Web*, 2011.

[3] V. Lopez, M. Pasin, and E. Motta. Aqualog: An ontology-portable question answering system for the semantic web. *The Semantic Web: Research and Applications*, pages 135–166, 2005.

[4] G. Sacco and Y. Tzitzikas. *Dynamic taxonomies and faceted search*, volume 25. Springer, 2009.

[4]http://www.opensearch.org/