Towards an evaluation of semantic searching in digital repositories: a DSpace case-study

Georgia Solomou and Dimitrios Koutsomitropoulos
High Performance Information Systems Laboratory (HPCLab), Computer Engineering and Informatics Department, School of Engineering, University of Patras, Patras-Rio, Greece

Received 4 July 2013
Revised 26 January 2014
Accepted 18 September 2014

Abstract

Purpose – Successful learning infrastructures and repositories often depend on well-organized content collections for effective dissemination, maintenance and preservation of resources. By combining semantic descriptions already lying or implicit within their descriptive metadata, reasoning-based or semantic searching of these collections can be enabled and produce novel possibilities for content browsing and retrieval. The specifics and necessities of such an approach, however, make it hard to assess and measure its effectiveness. The paper aims to discuss these issues.

Design/methodology/approach – Therefore in this paper the authors introduce a concrete methodology toward a pragmatic evaluation of semantic searching in such scenarios, which is exemplified through the semantic search plugin the authors have developed for the popular DSpace repository system.

Findings – The results reveal that this approach can be appealing to expert as well as novice users alike, improve the effectiveness of content discovery and enable new retrieval possibilities in comparison to traditional, keyword-based search.

Originality/value – This paper presents applied research efforts to employ semantic searching techniques on digital repositories and to construct a novel methodology for evaluating the outcomes against various perspectives. Although this is original in itself, value lies also within the concrete and measurable results presented, accompanied by an analysis, that would be helpful to assess similar (i.e. semantic query answering and searching) techniques in the particular scenario of digital repositories and libraries and to evaluate corresponding investments. To the knowledge there has been hardly any other evaluation effort in the literature for this particular case; that is, to assess the merit and usage of advanced semantic technologies in digital repositories.

Keywords Reasoning, Digital repositories, Information retrieval, Ontologies, Semantic query answering, Semantic search evaluation

Paper type Research paper

1. Introduction

Digital repository systems that exist today, offer an out-of-the-box and easily customizable solution to educational institutions and organizations for managing their intellectual outcome. Some of the most popular platforms include *DSpace*[1], with over a thousand instances worldwide[2], *EPrints*, *Digital Commons* and *CONTENTdm,* whereas *ETD-db* and *Fedora* have a notable presence as well (Bjork *et al.*, 2013).

In such a setting, semantic technologies can enable new dimensions in content discoverability and retrieval, by allowing the intelligent combination of existing metadata in order to enable semantic query answering among repository resources.

For such an approach to be appealing and thus useful to end-users, it has to withhold its inherent complexity, be as intuitive as possible, offer added-value and match user needs. In order to assess the extent to which these requirements are met, a careful evaluation has naturally to be conducted.

Program: electronic library and information systems
Vol. 49 No. 1, 2015
pp. 63-90

0033-0337
DOI 10.1108/PROG-07-2013-0037

However, not only the evaluation of semantic search systems is inherently difficult, but also the particular scenario of semantic query answering in digital repositories creates distinct requirements that have to be taken into account, thus deviating from standard approaches. This, combined with lack of existing relevant paradigms can potentially obscure the added-value semantic search has to offer for digital repositories.

To overcome these issues, in this paper we present a methodology and procedure for evaluating semantic query answering in digital repositories. To this end, we use the semantic search plugin that we have developed for the DSpace repository system and which is openly available to the community. After briefly introducing semantic search for DSpace, we review relevant literature and discuss the evaluation methodology we actually followed. In doing so, the specific evaluation criteria comprising our methodology as well as the distinct evaluation phases are identified and their association to the entailment-centric character of semantic searching in repositories is pointed out. Finally, we present the concrete evaluation procedure, discuss its results and outline our observations w.r.t. system behavior, user appeal and future improvements.

The rest of this paper is organized as follows: Section 2 gives an overview of semantic search for DSpace, a self-contained plugin for semantic query answering in the DSpace digital repository system, as well as in any web accessible ontology document. Section 3 describes our evaluation methodology and procedure, starting with the specific necessities for digital repositories, reviewing-related work and then documenting the evaluation criteria it is comprised of. Sections 4 and 5 present and discuss the process and results of applying this methodology for evaluating semantic search in DSpace in two phases, respectively. Finally Section 6 summarizes our conclusions and outlook.

2. Semantic search for DSpace

Semantic search v2 for DSpace is the outcome of an initiative to enable semantic searching capabilities for web-based digital repository systems.

The main idea in v2 is to break down the intended user query into a series of atoms and to assign them to different parts of a dynamic UI, enriched with auto-complete features, intuitive syntax-checking warnings and drop-down menus (Figure 1). In fact, these atoms correspond to the building blocks of a Manchester Syntax expression (Horridge and Patel-Schneider, 2008). Each such expression is an atomic or anonymous class in the Web Ontology Language (OWL) (Motik *et al.*, 2012) and its (both inferred and asserted) members are the answers to the query.

Search is conducted against the DSpace ontology, which is automatically populated through the Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH) interface (Lagoze *et al.*, 2002), and results are presented as a browsable list. Effective querying of the knowledge base is accomplished by interconnecting to an appropriate inference engine, capable of reasoning with OWL 2 ontologies.

Ontology handling, query processing and reasoner communication are managed by the *DSpace Semantic API*, which exhibits a pluggable reasoner design as well as proper handling of OWL 2 and is based on OWL API v3 (Horridge and Bechhofer, 2009). As a result, the new Semantic API features the ability to "hot-swap" between reasoners dynamically (see Figure 2). For the time being, any OWL API compliant reasoner can be supported, including out-of-the-box support for Pellet, FaCT++ and HermiT.

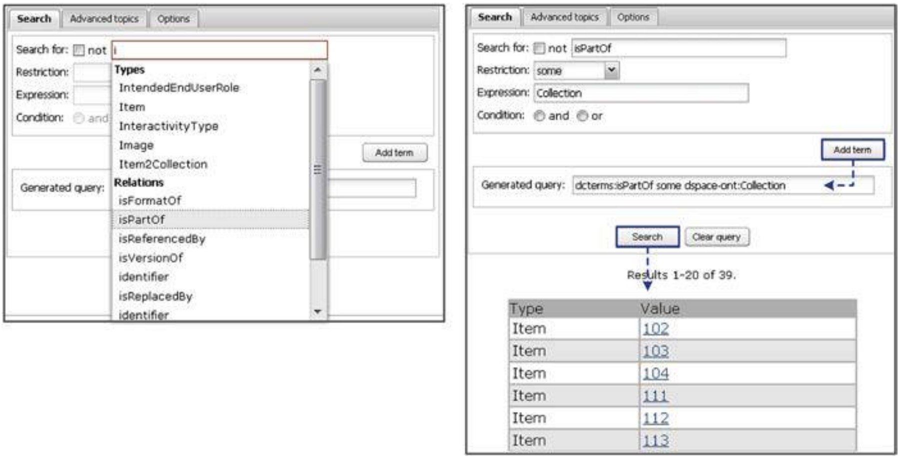


Figure 1.
The auto-complete and query construction mechanisms of the semantic search interface

Figure 2.
Changing reasoner through the options tab

Most importantly, this Semantic API is designed along the same principles as its predecessor, i.e. to remain independent of the DSpace business logic and to be agnostic to the rest of the user interface or even to the underlying ontology. In addition, it has undergone several improvements aiming at greater extensibility and performance, like, for example, ontology caching, inference precomputation and reasoner sharing and reuse. Finally, semantic search v2 includes several enhancements and/or fixes throughout the DSpace ontology creation process as well as in the involved ontology documents, aiming for better resource discovery, reasoner compatibility and extended inference capabilities.

Semantic search is hosted and maintained as a Google Code project[3] and is listed as an official DSpace add-on[4]. A public demo is also available[5]. A more thorough account of the architecture, methodology and design principles behind semantic search v2 can be found in (Koutsomitropoulos *et al.*, 2011).

3. Evaluation methodology

In this section we present the methodology for evaluating semantic query answering in digital repositories. We first note the added value of semantic searching especially within the context of a digital repository. We specify the requirements of evaluation for this specific scenario by identifying the determinant characteristics for this kind of applications. We then survey-related work in the field and point the major differences and incompatibilities with other related approaches in the literature. In total, two evaluation phases emerge, which both serve the purpose of assessing system qualities along different and often overlapping dimensions, based on a number of specific evaluation criteria.

3.1 Requirements for evaluating semantic search in digital repositories

Semantic searching in digital repositories can be considered as a complementary task that comes to embellish the suite of content discovery services for digital collections. However, it deviates from standard practice at least in two aspects: first, semantic queries require different mechanisms that cannot be put side-by-side with traditional search. Second, the evaluation of these mechanisms has to consider a set of distinct requirements, which are not typically met when implementing semantic search in general.

By traditional search we mean keyword-based search, i.e. search for the occurrences of particular text strings within available metadata. On the other hand, semantic search operates on additional knowledge that is implied by the combination of these metadata and their semantic interpretations in an ontology. This combination is made possible by inference engines that implement automated reasoning techniques based on formal logics (Horrocks, 2008), which are the essential underpinnings of the Semantic Web.

Keyword-based queries can often be ambiguous, due to the polysemy of terms used in the query, thus leading to low precision; they can also be affected by synonymity between query terms and the contents of the search corpus, thus leading to low recall. Semantic search as metadata-based search defined according to an ontology, enables overcoming both issues because ontology assertions are unambiguous and do not suffer from synonymity (Bhagdev *et al.*, 2008). In addition, semantic search enables a new set of queries that are based on the power of inference engines and are not possible with traditional keyword-based search.

The difficulty in implementing and evaluating semantic-enabled search has its roots in the requirement for expert syntaxes, the complexity of ontological queries and the need to have at least a rough knowledge of the ontologies involved beforehand (Koutsomitropoulos *et al.*, 2011). Beyond these general preconditions, a number of additional requirements need also to be fulfilled in case we search a digital repository. In essence, querying the Semantic Web is one thing; however, applying semantic search in the setting of digital repositories is quite another, since the latter have a special scope, a more specific user-base and the nature and status of information maintained in them is different. This set of requirements is described below. Each requirement triggers specific evaluation criteria, which are explained in more detail in Section 3.3:

- R1. User experience/expertise/background. Searching the contents of a repository is more than just browsing among hundreds or thousands of digital resources. The effectiveness of a semantic search service has to strike a balance between assisting end-users with the composition of complex queries and giving them freedom to express semantically rich concepts. In the end, digital repository users need not always be Semantic Web experts and they do not necessarily have specific expectations out of a semantic query mechanism, which could justify the additional effort to put up with it. Investigating user experiences and background (criterion C1-Demographics) is therefore necessary in order to be able to comprehend their reaction to system behavior (C3-Friendliness/intuitiveness) and to assess the impact of the new service modalities (C4-Perception of semantic search and reasoning).

- R2. Structure and quality of metadata. In a digital repository, the responsibility of documenting resources lies among domain experts, which are typically assigned with this task, rather than arbitrary users. Consequently, well-defined metadata records are produced, which contain rich and high quality descriptions. Because of this, it may be difficult to overlay the full semantic interpretation and complexity of stored metadata, during the querying process, as this depends on the quality of the produced transformation into an ontology. In addition, there is a semantic gap among a flat metadata record and its corresponding translation into a semantic description. On the other hand, general semantic search systems start already with a proper ontology. The most frequently encountered causes of failure (C2-Causes of failure), as well as a sufficiently refined result set (C7-Number of results) are indicative of how well the system can perform in terms of knowledge interpretation. Additionally, the types of semantic queries (C9-Query types), compared against their complexity and result productivity, may reveal system's potential to handle implicit and interconnected knowledge.

R3. Need for reasoning. Implied relations that lie among a repository's well-documented metadata descriptions can come to surface only when they are appropriately processed and reasoned about. To evaluate the actual merits of reasoning, we first need to figure out how users perceive entailment-based query answering (C4-Perception of semantic search and reasoning) and how they think such a semantic search facility could be further improved (C5-Improvements/suggestions). Moreover, the time needed for a request to be processed (C6-Response times) and the types of queries users tend to submit (C9-Query types), could give us a hint about the best possible trade-off among knowledge discovery and performance, on behalf of the end-users.

R4. Added value. By default, a common metadata-based search facility is provided to digital repository users, with the latter being accustomed to its functionality and performance. On the other hand, the paradigm-shift to a new, semantic-enabled search interface, with additional complexity in use, may raise some hesitation. Therefore, users need to be able to find out the actual gains of intelligent searching through their own experience. To this end, we investigate both users' perception about semantic search and reasoning (C4-Perception of semantic search and reasoning) and their consideration on how these constructs can be improved (C5-Improvements/suggestions). Nonetheless, added value is also captured in an implicit manner, through the query analysis process. The average query length (C8-Query length) reflects how much the underlying interface may facilitate composite and probably more expressive requests, something which is also indicated by measuring query success rates and causes of failure (C2-Causes of failure).

3.2 Background

Semantic search tools, although rapidly evolving in the last decade, lack a common evaluation practice which would contribute to their performance improvement and their broader acceptance by the world wide web community. As identified in Fernandez *et al.* (2009), even though some well-established techniques have been adopted for the evaluation of IR search systems – like those used in the annual TREC (Text REtrieval Conference) competitions[6] – semantic technologies are still a long way from defining standard evaluation benchmarks.

On the other hand, the use of standard IR metrics, like recall and precision, are not always sufficient for rating semantic search systems' performance and efficiency.

As discussed in Section 3.1, measuring precision and recall makes sense only when synonymity and ambiguity are determining factors in the quality of search results. On the contrary, an ontology-based semantic search system that employs sound and complete algorithms will always exhibit perfect precision and recall, as is typically the case in Pan *et al.* (2006). What seem as the most widely accepted techniques for these systems' evaluation are those based on user-centered studies (McCool *et al.*, 2005; Schandl and Todorov, 2008).

The state of the art in the domain of the semantic search evaluation appears very limited, either missing comprehensive judgments or producing very system-specific results that are inappropriate for making comparisons. Reasons for this include the inherently complex and diverse structure of semantic technologies, the lack of common evaluation algorithms (Fernandez *et al.*, 2008; Hildebrand *et al.*, 2007) or other deficiencies like the inability to identify suitable metrics and representative test corpora for the Semantic Web (Uren *et al.*, 2011). Among the evaluation frameworks found in the literature, those presented below can be considered as closest to our work, although they come with significant differences, mainly in that: they often ignore the capabilities for inference-based knowledge acquisition and reasoning and they do not consider the specific requirements applying for digital repositories (Section 3.1).

Within the Semantic Evaluation At Large Scale (SEALS) project (Nixon *et al.*, 2011; Wrigley *et al.*, 2010), a systematic approach for benchmarking semantic search tools is proposed, divided into two phases. The adopted methodology, unlike to ours, disregards the reasoning perspective which is a key aspect for extracting implicit knowledge (requirement R3). What is more, it is not well suited for knowledge-based retrieval in the setting of digital repositories, where more specific requirements apply. For example, although query expressiveness is taken into account, no particular provision is made for cases where the required information already lies among rich and well-documented metadata records (R2). Also, the generic scope of the SEALS campaign does not take into account the specific background and/or expectations that a digital repository user may have (R1).

Similarly, work presented in Blanco *et al.* (2011) proposes the evaluation of keyword-based searching over RDF using IR techniques. Therefore, this methodology focusses on assessing what is actually traditional search over structured text corpora (RDF documents) and the inference mechanisms and capabilities related to semantic search do not find their way in the evaluation.

The Question Answering over Linked Data (QALD) (Cimiano *et al.*, 2013) series of evaluation campaigns form an attempt to evaluate QALD. Its actual aim is to establish a standard evaluation benchmark and not a methodology in itself; therefore, it examines hardly any of the requirements posed in the previous section, like, for example, users' background and expectations (R1). Most importantly, QALD has a focus on systems which are about semantically assisted natural language question interpretation into graphs (SPARQL query language; Harris and Seaborne, 2012). What gets mostly evaluated by the QALD challenge is the translation of natural language questions into a formal query language, rather than the process and impact of constructing and submitting queries directly to a knowledge base, be it structured, semantically replicated or otherwise. What is more, the effect of inference-based query answering (R3) is not accounted for.

Finally, the framework introduced in Ekaputra *et al.* (2013) evaluates semantic querying systems from a more user-centric perspective. Its main concern is to guide users among the different types of ontology querying tools, rather than evaluating the actual capabilities and performance of the underlying retrieval mechanisms.

Our evaluation methodology consists of an end-user survey aiming to capture user satisfaction as well as user reaction to system-specific behavior. The latter is also measured by analyzing actual queries posed through a demo installation. Besides, as concluded in Strasunskas and Tomassen (2010), when evaluating semantic search systems the ultimate success factor is end-users' satisfaction. While reasoning effects play a considerable role in the evaluation process, our evaluation approach has still many in common with similar efforts in the literature: for once, survey questions match and refine the usability-oriented philosophy of the standard SUS questionnaire (Brooke, 1996), while the two-stage approach of survey- and query-analysis is also shared by the SEALS evaluation campaign ("user-in-the-loop" and "automated" phases (Wrigley *et al.*, 2010)). The logging analysis method has also been proposed in Uren *et al.* (2011).

There are different interpretations of semantic searching in the literature (e.g. see Strasunskas and Tomassen, 2010). The methodology proposed in this paper is clearly applied in entailment-based querying answering through structured queries in digital repositories. Different querying approaches, like NITELIGHT and PowerAqua (see Koutsomitropoulos *et al.*, 2011) have yet to appear in repositories. However, the core methodology criteria would remain the same as they: take into account the specific problem space (repositories), emphasize a pragmatic end-user perception stance, focus on reasoning as the primary source of added-value of such queries. To our knowledge, this is the first effort to propose a framework and usage-case for the evaluation of semantic query answering in digital repositories.

3.3 Evaluation phases and criteria

In terms of the above, two distinct evaluation phases have been identified and implemented in this work:

(1) A user survey based on anonymous multi-part questionnaires aiming to capture users' impression and satisfaction about the system. Beyond the standard SUS questionnaire, this survey includes further refinements that allow us to correlate between user demographics and/or particular interface characteristics (Section 4).

(2) The calculation of metrics and statistics about actual queries coming from various logged data. In this way, not only the pragmatic evaluation requirements are accommodated, that would be impossible to capture otherwise, but this phase also helps validate and double-check the outcomes of the former (Section 5).

Both these phases are designed in order to evaluate semantic search across different criteria, taking specifically into account the capabilities for knowledge-based acquisition and entailment. These criteria are often found to intersect both phases, thus helping verify the findings of both and produce correlations. The evaluation criteria and their rationale are analyzed:

C1. Demographics: this information is used as the basis for investigating potential demographic correlations with user behavior and performance with the system. In particular, users are classified into groups according to their expertise in Semantic Web and Manchester Syntax. These groups are then taken into account when considering the actual usage of the provided interface and the successful outcomes, like the percentage of successful queries.

C2. Causes of failure: a system's failure to process and answer a query can degrade its final acceptance by users. Knowing what actually caused this failure, be it a systemic flaw or user misconception, can help in improving the system. Causes of failure are captured both by analyzing logged system's exceptions, as well as by explicitly asking user opinion from a list of potential reasons.

C3. Friendliness/intuitiveness: expressing a semantic query is quite a tricky task and requires both familiarity with the query language as well as a minimum ability in comprehending and handling the underlying ontology's concepts. A user-friendly and intuitive interface should be capable of smoothly guiding users through the query construction process. The extent to which this is achieved can be captured by evaluating users' perception about provided structures and facilities (auto-complete, help options, overall intuitiveness). It is also important to assess the overall effort it takes for a user to construct a query, to identify the potential difficulties met during this effort and to survey the time it took to assemble the query using the interface.

C4. Perception of semantic search and reasoning: entailment-based queries manage to retrieve non-explicit knowledge and this is a major asset of any semantic search system. To have an idea of how users comprehend and feel with such a feature, answers to questions that compare semantic with traditional search are analyzed.

C5. Improvements/suggestions: the success of a semantic search service mostly lies in the potential of its underlying semantic technologies and the way these are perceived by users. To further enhance such a service, users are asked to identify possible improvements, either from the semantic or usability perspective.

C6. Response times: given that accuracy is guaranteed by sound and complete algorithms, the total time needed to answer a query constitutes a core criterion for evaluating semantic search. Improved response times imply increased satisfaction for users. However, the extremely fast response times that traditional search engines come with, raise the standards for semantic search also. The system's actual response time is derived from logged data and its perception by the users can be reflected in survey responses concerning interface usability.

C7. Number of results: skewing the meaning of traditional recall, the productivity of ontological queries can be captured by the number of individuals obtained relative to the total number of individuals. This is useful in comparing groups of queries depending on their types, the ontology they were performed on or the reasoner involved and gives some indication about how refined results actually are.

C8. Query length: queries consisting of only a few characters are generally expected to produce fast responses and simple results. On the other hand, longer queries are more likely to be complex enough so as to produce refined results. In addition, how well auto-complete features and suggestion mechanisms assisted in query formulation can be effectively measured by average query length.

C9. Query types: queries intended to be evaluated upon semantic concepts and constructs can differ significantly in terms of complexity, both from a computational, as well as a logical and structural point of view. By partitioning queries into groups according to their structural and semantic characteristics and by examining times and other metrics related to these groups, we can have a useful indication of system's performance and user experience. For example, the frequency of occurrence for each query type can imply how easy was for users to construct fine-grained queries through the provided interface. The partitioning of queries into types depends on the underlying querying approach. In any case, a valid grouping may consider the complexity of the queries, i.e. how "hard" is for them to run and how productive they may be. In the case of entailment-based structured queries, this is interpreted as the type of the language constructs utilized, such as classes, expressions and restrictions, as well as of the axioms associated with these (see also Section 5.4).

The multi-dimensional character of these criteria is better explained in Figure 3, where their relative placement within the evaluation phases as well as their relationship to the assessment of reasoning capabilities is depicted.

4. User survey

In order to investigate users' behavior and experience in using Semantic Search, we constructed an online questionnaire[7] that mostly includes closed format questions. We are based on multiple-choice as well as on a three- and five-point Likert scale assessment. All questions are optional and organized in four groups: the first group corresponds to the criterion C1 ("Demographics"). The second group measures the C2 criterion ("Causes of failure") and the C3 criterion ("Friendliness/intuitiveness"). The third group of questions is related solely to the C3 criterion ("Friendliness/intuitiveness"). Finally, the forth group evaluates the criteria of C4 ("Perception of semantic search and reasoning") and C5 ("Improvements/suggestions"). The analysis below follows the criteria addressed.

4.1 Demographics and causes of failure

The majority of people that took part in this survey appear as being averagely familiar with Semantic Web concepts (mean average 3.07 in a five-point scale with 5 corresponding to "expert"). As expected, their Manchester Syntax level of knowledge (Figure 4, left) is a bit lower, but still most of them can at least recognize queries written in this language and possibly can construct some (mean average 2.70/5). To perform correlations later on, we categorize users into three major groups by collapsing the two upper and two lower levels into one category each.

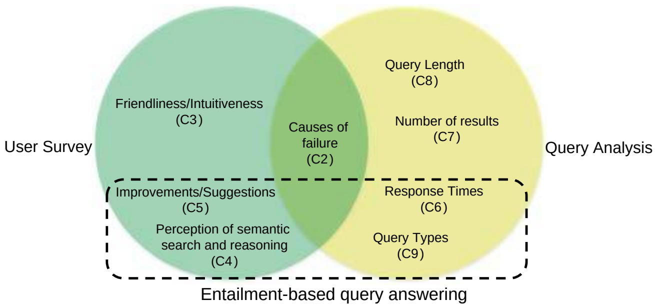


Figure 3. Evaluation criteria within evaluation phases and the reasoning aspect

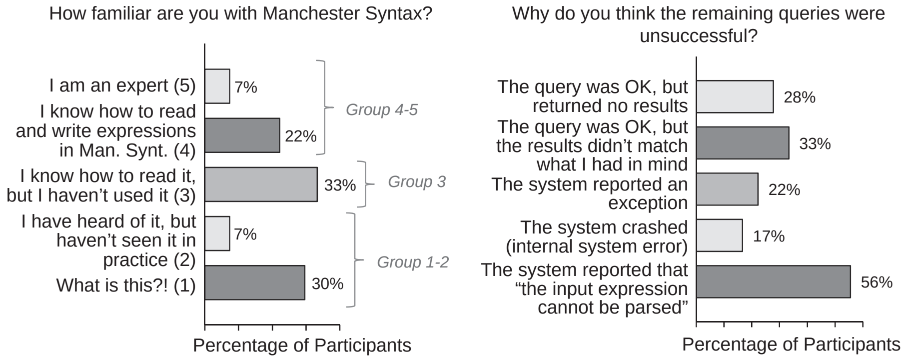


Figure 4.
Familiarity with Manchester syntax and reasons for query failure (reported)

The average percentage of successful queries reached 53 percent. This increases as users become more Manchester Syntax literate (see Table I). The percentages have been computed by taking into account the answers given by participants that have actually tried to build some semantic queries using our demo installation (81 percent of all participants), as these questions where only visible to them. Remaining queries failed mostly due to syntax-specific errors that resulted in problematic query parsing (56 percent, Figure 4, right part). Another important reason for unsuccessful queries was that, by users' claim, the results they got did not match what they expected (33 percent).

4.2 Friendliness/intuitiveness

Table I summarizes querying experience and interface intuitiveness as per expertise group and identifies user's attitude against manual query construction. It is clear that the less experienced a user is with the querying syntax, the more does he use the step-by-step query construction mechanism and refrains from doing so manually. This observation reveals that our guided query mechanism can appeal to inexperienced users and help them alleviate the complexity often found in formal language querying systems. Overall, the interface was considered to be moderately easy to understand and use, gaining a mean score equal to 2.96/5, being at its peak for the middle expertise group (1, "Very hard to understand and use"-5, "Very easy and straightforward").

In addition, the fact that the vast majority of replies (80 percent) indicates that users did not show preference to the "generated query" box (they used it "only a few times" or "never") and that a 20 percent had exclusively used the step-by-step procedure, shows that the semantic search interface came in handy for a significant part of them. Besides, everyone said they were able to assemble and submit a query in less than a minute and more than one-third of them (35 percent) answered that a time interval of 10 seconds was enough (Figure 5).

The most important aspect of the semantic search interface proves to be its auto-complete mechanism. On average, the survey participants rated this particular feature with the mean score 3.80/5 replying to the question "Did you find the auto-complete facility helpful?" (1, "No, not at all"-5, "Very much indeed"). As far as the intuitiveness of labels is concerned, the average rates appear a bit worse, related to the ones regarding the auto-complete mechanism. On average, the names selected for labels were considered as a moderate choice, being judged with the mean score 3.04 out

Manchester syntax expertise	Percentage of successful queries		Did you have to edit the "generated query" box manually? (the two most frequent replies)		Overall, how easy was for you to understand and/or use the interface?
1-2	42	80	Only a few times	60%	2.56
37%		30	Never. I was more comfortable creating the query step-by-step	40%	
3	53	48	Only a few times	63%	3.22
33%		50	Never. I was more comfortable creating the query step-by-step	25%	
4-5	65	60	Only a few times	57%	3.14
30%		70	More often than not, it is easier for me this way	43%	
Total	53	59	Only a few times	60%	2.96
		40	Never. I was more comfortable creating the query step-by-step	20%	

Table I.
Manchester Syntax expertise level correlated with query success rates, frequency of use of the "generated query" field and overall experience with the semantic search interface

of 5. The reason is that we have to cope with Manchester Syntax's specific terminology, which can be overwhelming for non-experts.

4.3 Perception of semantic search and improvements

To evaluate the participants' perception of the service's retrieval capability and reasoning-centric character, we included some questions in the last part of the survey that are analyzed below. Compared to the traditional keyword-based search the semantic search service is considered to be better, as shown by the number of positive votes (68 percent of the respondents indicated at least one positive reason, see Figure 6).

The observation that semantic search outclasses traditional search in terms of improved precision and recall comes exactly as a consequence of the service's capability to perform reasoning. Another reasoning-related question is the one about possible improvements, with 57 percent of users requesting support for more advanced constructs of Manchester Syntax (like facets, automatic use of parentheses and inverse properties) as well as the addition of more reasoners (43 percent). Nevertheless, additional and better tooltips for the various fields and boxes seems to be a more important improvement, indicated by 62 percent of respondents. This comes as a consequence of the moderate 3.04 mean score assigned previously to the labels intuitiveness. It is worth mentioning that the majority (38 percent) of users pointing out this particular improvement gave as main reason for query failure that the results they obtained did not match what they had in mind. Although not necessarily user-friendlier, the extension of semantic search with a SPARQL endpoint has the

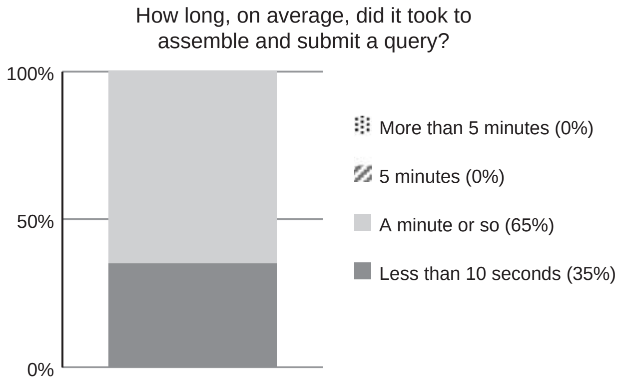
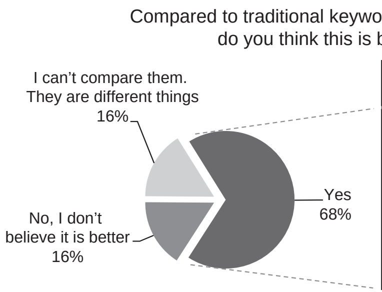


Figure 5.
Distribution of query construction times among users



Positive reasons (multiple replies are possible)	
Yes, results are more accurate.	42%
Yes, there are more results returned.	42%
Yes, it is easier to create a query	4%
Yes, for some other reason.	17%

Figure 6.
Comparison between traditional keyword-based search and the semantic search service

highest popularity among user suggestions (76 percent), possibly due to standardization and added interoperability.

The vast majority of users (78 percent) found that our semantic search interface makes semantic searching "very much" or at least "somehow" easier and only 17 percent disagrees with this viewpoint (answering "not at all"). In a three-point Likert scale this question scores an average of 2.13. It is important to mention that by collapsing overall interface intuitiveness (Table I) into a three-point scale, the average score of both these questions is quite close (1.96 and 2.13 out of 3, respectively). This indicates that there exists a direct relationship between semantic searching and the user friendliness of the underlying interface and suggests that semantic search can benefit from interface intuitiveness.

Participants' appreciation of semantic searching in digital repositories has an influence in their perception of our own system (Table II). Those that find semantic searching a confident idea rank our service higher and are more enthusiastic adopting it, while those not committed tend to exhibit a negative bias, thus lowering its average score.

5. Query analysis

In this section we analyze real queries posed through a demo installation of the semantic search mechanism[8]. The demo runs on a 64bit Intel Xeon at 2.8 GHz and Java is allowed a maximum heap memory of 1,024 MB. Data were collected during a period of 12 months, from March 2012 to March 2013 and correspond to the official deployment of the semantic search v2 minor upgrade (v2.1).

5.1 Ontologies used

During this evaluation phase the most part of user queries were posed against the DSpace ontology, since this is the one selected by default. However, the interface permits loading and use of any ontology document. We found that users experimented also with other ontologies. A summary of collective data and metrics regarding user experience and system response with the four most popular ontologies (Table III) is shown below (Figures 7 and 8). To give a better understanding of overall query efficiency we also report the total and/or average values of these metrics for all ontologies.

5.2 Success rates

Query percentage (percent queries) is the ratio of successful queries to the total number of queries logged by the system. Results percentage (percent results) is the average number of results returned by a group of queries relative to the maximum number of results that may be returned by any query, i.e. the total number of individuals in the ontology. In the case of all ontologies, this represents the weighted mean of the results percentage for each ontology. Total quantities are also shown in absolute numbers in Figure 7. Average query length is given in characters.

From a technical point of view "successful" queries are those that achieve reasoner response without errors, i.e. succeed in fetching results unabruptly, even zero ones. Query analysis indicates a 55.3 percent of such queries over all ontologies, increasing to 66.5 percent for the default DSpace ontology case. This is in accordance to the findings of the user survey (53 percent, Table I), though leaning a little on the positive side by also including queries that did not match what users had in mind, but worked nevertheless (see Figure 4, right part).

The most part of technically unsuccessful queries (i.e. queries that caused one exception or another) failed unsurprisingly due to syntax errors (83.9 percent of logged

Table II. Users' opinion about usefulness of general semantic searching in repositories, correlated with their attitude toward the semantic search service

	Do you think semantic searching is useful in repositories?	Do you think this interface makes semantic searching easier?		Would you like to have this service in your repository?		Overall, how easy was for you to understand and/or use the interface? (1-5)
	3: "Very useful" (74%)	Very much	53%	Yes	94%	3.29
		Somehow	29%	No	6%	
		Not at all	12%	n/a	–	
		n/a	6%			
	2: "Somehow" (22%)	Very much	–	Yes	40%	2.20
		Somehow	80%	No	20%	
		Not at all	20%	n/a	40%	
		n/a	–			
	1: "Not at all" (4%)	Very much	–	Yes	–	1.00
		Somehow	–	No	100%	
		Not at all	100%	n/a	–	
		n/a	–			
Total	2.70	Very much	39%	Yes	78%	2.96
		Somehow	39%	No	13%	
		Not at all	17%	n/a	9%	
		n/a	5%			

exceptions), just like the users report in the survey. This is followed at a distance by errors related to reasoner limitations (6.4 percent), like unsupported datatypes and facets, thus explaining to some extent the explicit exceptions reported by users (Figure 4). An important part of logged exceptions consists of ontology-specific errors

Short name	Description	No. of entities	Expressivity
dspace-ont	The DSpace ontology	304	*SROIF* (OWL 2)
GCMD	Location keywords of the Global Change Master Directory (GCMD) in SKOS	464	*SHIF* (OWL-Lite)
SPASE	Part of the Space Physics Archive Search and Extract (SPASE) data model in SKOS	461	*SHIF* (OWL-Lite)
SPASE-test	A test version of SPASE, including a nominal	461	*SHOIF* (OWL-DL)

Table III. Most frequently used ontologies and their expressiveness

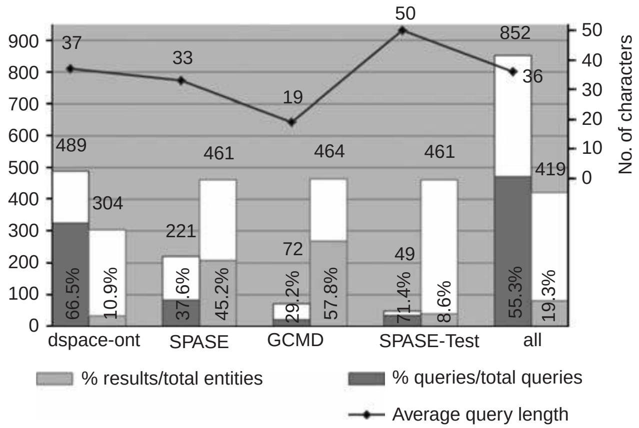


Figure 7. Query success rates and results per ontology

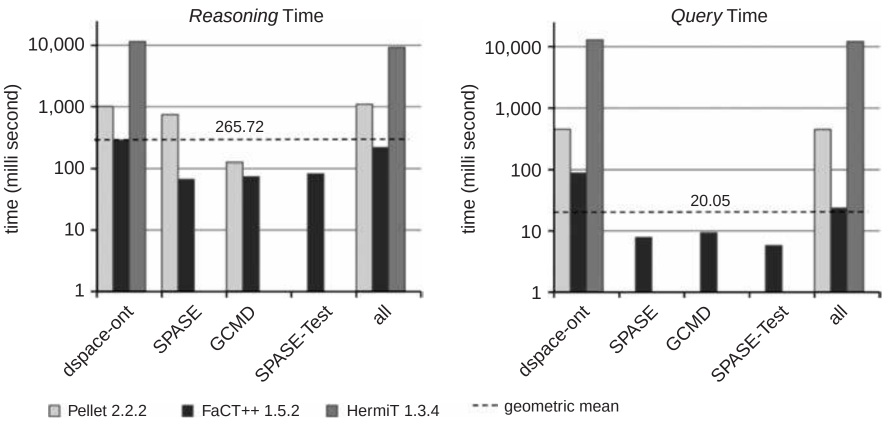


Figure 8. Reasoning and query time per reasoner per ontology

(9.7 percent), like unresolvable URLs, malformed or inconsistent ontologies, which are responsible for "internal system errors". This is to be expected though, since the system currently allows users to operate on any web-accessible ontology.

5.3 Performance

Figure 8 summarizes average time measurements for the set of successful queries, and groups them per reasoner and ontology used. Reasoning time corresponds to inferences precomputation by the reasoner. This is an initialization phase were the reasoner classifies the ontology, and precomputes certain reasoning operations like, for example, realization, satisfiability and instance checking (Nardi *et al.*, 2002). Reasoning is a hard computational task and increases as an ontology grows in complexity (Table III). However, precomputation happens only once and results remain available in memory for future queries to perform on.

Query time corresponds to query evaluation, i.e. the time it takes for the reasoner to evaluate the input class expression and to decide its instance membership (what individuals belong to the evaluated class). Due to precomputation, query times are generally submultiple of reasoning times (but see also next section).

As it is evident in Figure 8, measured reasoner times exhibit great discrepancies, since they are spreading over totally different orders of magnitude. This is primary due to the inherent differences in the reasoners themselves, their architecture and optimization strategies (Bock *et al.*, 2008). In such situations, it is therefore preferable to use the geometric mean, as a means to mitigate the asymmetry of different reasoner scaling, which does not really account for actual system behavior and to better convey overall tendency. In the following, when we refer to average times we use the geometric mean.

These times are essential, since users may be accustomed to rapid response times from their everyday experience with traditional search engines. On the other hand, systems that employ semantics to offer enhanced search services can be somewhat more burdened. A recent study on the evaluation of such systems reports average response times between a few milliseconds and several seconds (Elbedweihy *et al.*, 2012). In this setting, the fact that semantic search for DSpace exhibits average query (and reasoning) times of less than half a second can be considered satisfactory.

Finally, Figure 9 shows queries distribution along a logarithmic time scale. Most frequent appear to be the queries that take between 1 and 10 milliseconds to get evaluated, meaning that users get, for the most part, a quite responsive experience by the system.

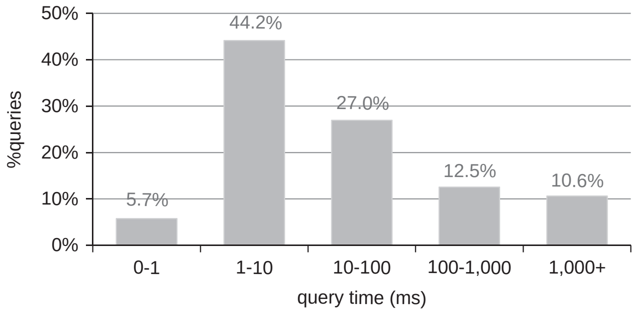


Figure 9.
Percentage of queries per time scale

5.4 Types of queries

Figure 10 partitions logged queries into groups based on their types and characteristics. As will become evident, the number of results returned as well as query evaluation times are directly dependent on these query types. Both an analysis of variance (ANOVA) as well as a Kruskal-Wallis test yield a $p \ll 0.0001$, thus attesting that this partitioning and the observations that follow are also statistically significant.

Query evaluation times. Clearly, since a precomputation has preceded, query evaluation may range from very low (practically insignificant) times to several milliseconds, depending on whether query results are already in memory or a recomputation is required. For example, a named class expression like "owl:Thing" or "dspace-ont:Item" must be fast in principle, due to the fact that the realization and instance checking results are already available. On the other hand, queries involving properties, e.g. existential, value and number restrictions may require reclassification of the ontology so as to decide the placement of the new expression in the existing class hierarchy and can be much more costly.

This is especially true in cases where properties subsume long hierarchies of other properties or role chains (with transitivity being a special case of chain), since the corresponding Description Logics algorithms have to investigate exhaustively all possible paths connecting instances, based on the role hierarchy.

Number of results. As shown in Figure 10, more than half of user queries (60.3 percent) consist of property restrictions. However, these queries resulted on average in a small percent (12.9 percent) of total individuals. On the other hand, the simpler single-named class expressions are somewhat less frequent, but are accountable for a far greater number of results (30.5 percent). This confirms that users were able to take advantage of the new features of the interface and construct also complex and "educated" queries filtering effectively the result space. Besides, the vast majority of users reported that only rarely did they opt for manual query construction and liked better to utilize the guided query mechanism, according to the user survey.

It should be noted that the syntactic characteristics of user queries, as they are generally identified in Figure 10 do not have any direct effect on their knowledge retrieval capabilities. Instead, it is the semantic qualities of the vocabulary and expressions used within the queries that determine their ability to perform knowledge acquisition.

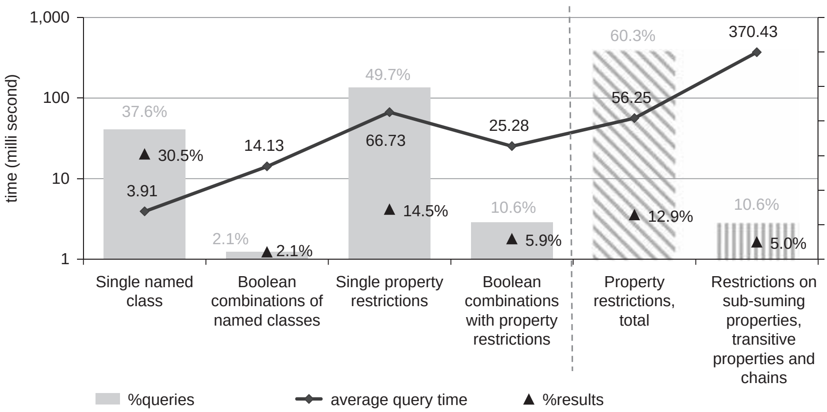


Figure 10. Percentage of queries, average times and average percentage of results per query type

A notable exception is restrictions on subsuming properties, transitive properties and chains, which actually consider semantically "intense" properties, due to their manipulation in the ontology. Queries of this type are expected to exhibit improved retrieval capabilities, by design; however, the opposite does not necessarily hold.

As a final observation, users appear reluctant to build Boolean combinations, possibly because this requires an extra step in the query construction process ("Add term" button in Figure 1).

5.5 Length of queries

Next, Figure 11 presents queries organized into groups based on their length and correlates them with their average query time and returned results percentage. It can be noticed that:

- Most user queries are between ten and 20 characters long. In addition, average query length is 36 characters (Figure 7), which is larger than usual web search queries. For example, average query length in Google is around 20 characters (Tsotsis, 2010) and similarly in Bing, new queries appear to be almost 19 characters long (Tyler and Teevan, 2010). This is due to the fact that semantic search offers several enhancements that reduce users' workload, such as prefix auto-completion, pull-down menus and auto-complete suggestions for entity names, all making easier for users to construct longer and expressive queries.
- Query evaluation time is not directly dependent on query length. This is a reasonably expected result, since query evaluation does not depend on the query size but on the type and characteristics of the query as explained above (restrictions, properties, etc.). Still, however, average time may increase as queries lengthen, because there is greater chance for "hard" queries to occur as query length increases.
- Finally, there is a general tendency for the result-set size to decline as query length increases. In conjunctive term-based search this is a well-expected observation, since multiple conjunctions tend to prune progressively the result set (they have higher precision: Kumar and Prakash, 2009). Here, longer queries are not necessary conjunctive; however, small queries are impossible to contain restrictions or fine-grained class expressions that would limit the number of results and that would only be possible within larger queries (e.g. the Manchester

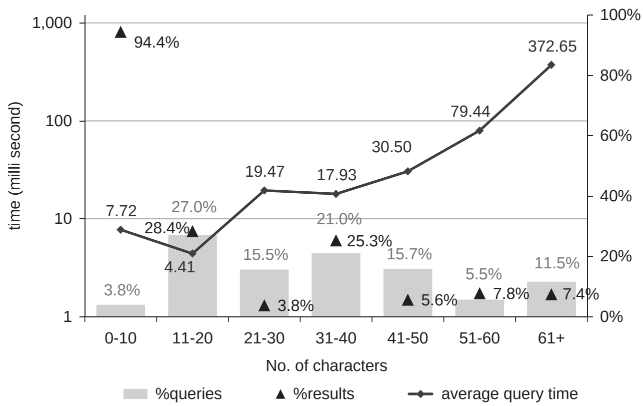


Figure 11.
Percentage of queries, results and average query times per query length

Syntax keyword “some” already contains four characters and accounts for 11 percent of the average query length).

6. Conclusions

The added value of semantic query answering is sometimes overlooked, possibly because of the dominance of traditional, text-based searching and its satisfactory results. This is not to say that semantic search can replace traditional methods; however, it can evidently provide additional querying dimensions, made possible by unfolding content semantics and automated reasoning.

In this paper we have introduced a systematic approach toward the evaluation of semantic searching in knowledge intensive set-ups, like digital repositories. It is easily seen that traditional IR metrics, such as recall and precision, are often incompatible or inappropriate in these scenarios. The user-centric character as well as the reasoning-specific characteristics of such systems necessitates a pragmatic stance and we have shown specific requirements, criteria and procedures on how this can be achieved.

This methodology has been applied on the DSpace semantic search plug-in and has helped to identify the system’s strengths and weaknesses as well as to confirm the worthiness of improvements introduced with v2. For example, Manchester Syntax may not arise as an alternative to natural language, but there are options that can alleviate this burden. In any case, no matter what syntax or querying approach is actually followed, we have indicated specific metrics like the results ratio or query success rates that can help estimate query quality and user engagement.

In fact, this idea can be carried along all the other aspects a semantic search mechanism may have, from system architecture to interface design to reasoning strategy; no matter how they are actually implemented, our procedure can be equally applied and confirm (or refute) the validity of design decisions. As a result, this methodology may serve as a useful paradigm on how one can go on with assessing the effects, merits and challenges of deploying semantic search capabilities over existing repository infrastructures.

Notes

1. www.dspace.org
2. http://maps.repository66.org/
3. http://code.google.com/p/dspace-semantic-search/
4. https://wiki.duraspace.org/display/DSPACE/Extensions+and+Addons+Work#Extensionsand AddonsWork-SemanticSearchforDSpacev2.0
5. http://swig.hpclab.ceid.upatras.gr/dspace-ss-demo
6. http://trec.nist.gov/
7. http://goo.gl/Zt0aP
8. http://swig.hpclab.ceid.upatras.gr/dspace-ss-demo

References

Bhagdev, R., Chapman, S., Ciravegna, F., Lanfranchi, V. and Petrelli, D. (2008), “Hybrid search: effectively combining keywords and semantic searches”, *5th European Semantic Web Conference, ESWC2008, Tenerife, June 1-5*.

Bjork, K., Isaak, D. and Vyhnanek, K. (2013), "The changing roles of repositories: where we are and where we are headed", Library Faculty Publications and Presentations, available at: http://pdxscholar.library.pdx.edu/ulib_fac/84 (accessed November 25, 2013).

Blanco, R., Halpin, H., Herzig, D., Mika, P., Pound, J., Thompson, H. and Duc, T. (2011), "Entity search evaluation over structured wed data", *1st International Workshop on Entity-Oriented Search at 34th Annual ACM SIGIR Conference*, *ESWC2008, Beijing, June 1-5*.

Bock, J., Haase, P., Ji, Q. and Volz, R. (2008), "Benchmarking OWL reasoners", *Advancing Reasoning on the Web: Scalability and Commonsense Workshop at the 5th European Semantic Web Conference*, *ARea2008, Tenerife, June 1-5*.

Brooke, J. (1996), "SUS: a 'quick and dirty' usability scale", in Jordan, P.W., Thomas, B., Weerdmeester, B.A. and McClelland, A.L. (Eds), *Usability Evaluation in Industry*, Taylor and Francis, London.

Cimiano, P., Lopez, V., Unger, C., Cabrio, E., Ngonga Ngomo, A.-C. and Walter. S. (2013), "Multilingual question answering over linked data (QALD-3): lab overview", *4th Conference and Labs of the Evaluation Forum*, LNCS 8138, Springer-Verlag, Berlin and Heidelberg, pp. 321-332.

Ekaputra, F.J., Serral, E., Winkler, D. and Biffl, S. (2013), "An analysis framework for ontology querying tools", *9th International Conference on Semantic Systems*, *ISEM2013, Graz, September 4-6*.

Elbedweihy, K., Wrigley, S.N., Ciravegna, F., Reinhard, D. and Bernstein, A. (2012), "Evaluating semantic search systems to identify future directions of research", *2nd International Workshop on Evaluation of Semantic Technologies at 9th Extended Semantic Web Conference*, *ESW2012, Heraklion, May 27-31*.

Fernandez, M., Lopez, V., Motta, E., Sabou, M., Uren, V., Vallet, D. and Castells, P. (2009), "Using TREC for cross-comparison between classic IR and ontology-based search models at a web scale", *Semantic Search Workshop at 18th International World Wide Web Conference*, *Madrid*.

Fernandez, M., Lopez, V., Sabou, M., Uren, V., Vallet, D., Motta, E. and Castells, P. (2008), "Semantic search meets the web", *IEEE International Conference on Semantic Computing*, *Santa Clara, CA*, pp. 253-260.

Harris, S. and Seaborne, A. (2012), SPARQL 1.1 Query Language. W3C Recommendation, available at: www.w3.org/TR/sparql11-query/ (accessed December 28, 2013).

Hildebrand, M., van Ossenbruggen, J.R. and Hardman, L. (2007), "An analysis of search-based user interaction on the semantic web", Technical Report No. INS-E0706, Centrum voor Wiskunde en Informatica (CWI), Amsterdam.

Horridge, M. and Bechhofer, S. (2009), "The OWL API: a Java API for working with OWL 2 ontologies", 6th OWL Experiences and Directions Workshop, Chantilly, VA, October 23-24.

Horridge, M. and Patel-Schneider, P.S. (2008), "Manchester Syntax for OWL 1.1", 4th OWL Experiences and Directions Workshop, Gaithersburg, MD, April 1-2.

Horrocks, I. (2008), "Ontologies and the semantic web", *Communications of the ACM*, Vol. 51 No. 12, pp. 58-67.

Koutsomitropoulos, D., Borillo, R. and Solomou, G. (2011), "A structured semantic query interface for reasoning-based search and retrieval", *8th Extended Semantic Web Conference*, *ESWC2011, Heraklion, May 29-June 2*.

Kumar, B.T.S. and Prakash, J.N. (2009), "Precision and relative recall of search engines: a comparative study of Google and Yahoo", *Singapore Journal of Library & Information Management*, Vol. 38 No. 1, pp. 124-137.

Lagoze, C., Van de Sompel, H., Nelson, M. and Warner, S. (2002), "The Open Archive Initiative Protocol for Metadata Harvesting", available at: www.openarchives.org/OAI/openarchivesprotocol.html (accessed December 28, 2013).

McCool R, Cowell, A.J. and Thurman, D.A. (2005), "End-user evaluations of semantic web technologies", *Workshop on End User Semantic Web Interaction at 4th International Semantic Web Conference*, *EUSW2005, Galway, November 6-10*.

Motik, B., Parsia, B. and Patel-Schneider, P.F. (Eds) (2012), *OWL 2 Web Ontology Language XML Serialization*, 2nd ed., W3C Recommendation, available at: www.w3.org/TR/owl2-xml-serialization/ (accessed January 3, 2014).

Nardi, D. and Brachman, R.J. (2002), "An introduction to description logics", in Baader, F., Calvanese, D., McGuinness, D.L., Nardi, D. and Patel-Schneider, P.N. (Eds), *Description Logic Handbook*, Cambridge University Press, Cambridge.

Nixon, L., Garcia-Castro, R., Wrigley, S.N., Yatskevich, M., Trojahn Dos Santos, C. and Cabral, L. (2011), "The state of semantic technology today – overview of the first SEALS evaluation campaigns", *7th International Conference on Semantic Systems*, *Graz*.

Pan J.Z., Thomas, E. and Sleeman, D. (2006), "ONTOSEARCH2: searching and querying web ontologies", *IADIS International Conference WWW/Internet*, *IADIS WWW/Internet 2006, Murcia, October 5-8*.

Schandl, B. and Todorov, D. (2008), "Small-scale evaluation of semantic web-based applications", technical report, University of Vienna, Vienna.

Strasunskas, D. and Tomassen, S.L. (2010), "On variety of semantic search systems and their evaluation methods", *International Conference on Information Management and Evaluation, Academic Conferences International*, pp. 380-387.

Tsotsis, A. (2010), "Google instant will save 350 million hours of user time per year", TechCrunch, available at: http://techcrunch.com/2010/09/08/instant-time/ (accessed January 3, 2014).

Tyler S.K. and Teevan, J. (2010), "Large scale query log analysis of re-finding", *3rd ACM International Conference on Web Search and Data Mining*, *WSDM2010, New York, NY, February 3-6*.

Uren, V., Sabou, M., Motta, E., Fernandez, M., Lopez, V. and Lei, Y. (2011), "Reflections on five years of evaluating semantic search systems", *International Journal of Metadata, Semantics and Ontologies*, Vol. 5 No. 2, pp. 87-98.

Wrigley, S.N., Reinhard, D., Elbedweihy, K., Bernstein, A. and Ciravegna, F. (2010), "Methodology and campaign design for the evaluation of semantic search tools", *3rd International Semantic Search Workshop at 19th International World Wide Web Conference, SemSearch'10, Raleigh, NC, April 26-30*.

Appendix 1. Questionnaire

A. About the user

1. Your email: optional

2. How familiar are you with Semantic Web concepts?

	1	2	3	4	5	
Novice	○	○	○	○	○	Expert

3. How familiar are you with Manchester Syntax?

- ○ What is this?!
- ○ I have heard of it, but haven't seen it in practice.
- ○ I know how to read it, but I haven't used it.
- ○ I know how to read and write expressions in Manchester Syntax.
- ○ I am an expert.

4. Did you try to submit any queries?

- ○ Yes.
- ○ No.

B. Hands-on experience

(The following set of questions appear only if question 4 has been answers with "Yes")

5. How many queries did you actually submit? *(number)*

6. Out of them, how many were successful, i.e. they returned what you expected to return? *(number)*

7. Why do you think the remaining queries were unsuccessful? *(Tick all that apply)*

- ☐ The system reported that "the input expression cannot be parsed".

- The system crashed (internal system error).
- The system reported an exception.
- The query was OK, but the results didn't match what I had in mind.
- The query was OK, but returned no results.
- Other:

8. Did you have to edit the "generated query" box manually?

- Never. I was more comfortable creating the query step-by-step.
- Only a few times.
- More often than not, it is easier for me this way.
- All the time. There seems to be no other way to express my query.

9. How long, on average, did it took to assemble and submit a query?

- Less than 10 seconds.
- A minute or so.
- 5 minutes.
- More than 5 minutes.

10. Did you use "help"?

- Yes.
- No.

C. The Interface

11. Did you find the auto-complete facility helpful?

	1	2	3	4	5	
No, not at all.	◯	◯	◯	◯	◯	Very much indeed.

12. Did you find the text-box labels intuitive?

	1	2	3	4	5	
No, they don't match what I had in mind.	◯	◯	◯	◯	◯	Yes, they are precise and helpful.

13. Did you find "help" any helpful?

	1	2	3	4	5	
No, not at all.	○	○	○	○	○	Very much indeed.

14. Your comments on how to improve help:

15. Overall, how easy was for you to understand and / or use the interface?

	1	2	3	4	5	
very hard to understand and use..	○	○	○	○	○	very easy and straightforward.

D. Comparison and Suggestions

16. Compared to traditional keyword-based search, do you think this is better? *(tick all that apply)*

- ☐ Yes, results are more accurate.
- ☐ Yes, there are more results returned.
- ☐ Yes, it is easier to create a query.
- ☐ Yes, for some other reason.
- ☐ No, I don't believe it is better
- ☐ I can't compare them. They are different things.

17. Do you think semantic searching is useful in repositories?

- ○ Very useful.
- ○ Somehow.
- ○ Not at all.

18. Do you think this interface makes semantic searching easier?

- Very much.
- Somehow.
- Not at all.

19. Would you like to have this service in your repository?

- Yes.
- No.

20. Would you like to see this service as part of the official DSpace?

- Yes.
- No.

21. How can this service be improved? *(Tick all that apply)*

- Adding support for more reasoners.
- Adding a SPARQL endpoint.
- Support more Manchester Syntax constructs (facets, parentheses, inverse properties).
- Adding tooltips that explain the various fields/boxes.
- Other:

22. Other comments/suggestions:

Appendix 2. DSpace semantic search

Example Queries

1. Queries that obtain the same results

a. Queries also possible through DSpace "Advanced search"

Semantic Query	Ask for:	DSpace Equivalent
`dcterms:type value dspace-ont:Learning_Object`	*All learning objects*	**Type:** "learning object"
`dspace-ont:author some (foaf:surname value "Solomou")`	*All items having as author a person with surname "Solomou"*	**Author:** "Solomou"
`dspace-ont:author some (foaf:surname value "Solomou") and dcterms:type value dspace-ont:Working_Paper`	*All items of type "Working Paper" also having as author a person with surname "Solomou"*	**Author:** "Solomou" **AND** **Type:** "Working Paper"
`dcterms:type value lom:Slide and dcterms:type value lom:Easy`	*All entities that are of type Slide and are easy to learn*	**Educational Metadata:** "Slide" **AND** **Educational Metadata:** "Easy"
`dcterms:subject value "Linked Data"`	*All items that include "Linked Data" among their subjects*	**Subject**: "Linked Data"
`dcterms:instructionalMethod value lom:Expositive`	*All items that use expositive instructional method*	**Educational Metadata:** "Expositive"
`dcterms:extent value lom:Three_hours_to_Five_hours`	*All items that need three to five hours in order to be learned*	**Educational Metadata:** "three hours"
[HermiT only] `dcterms:subject some rdf:PlainLiteral [pattern ".*[sS]emantic.*"]`	*All items that contain "semantic" in their subject*	**Subject:** "semantic"

b. Queries also possible by DSpace "Browse"

Semantic Query	Ask for:	DSpace Equivalent
`dcterms:hasPart some dspace-ont:Item`	*All entities that have items as parts*	*Browse "Collections"*
`dcterms:issued value "2010"`	*All entities that have been issued during 2010*	*Browse by "Issue Date"*

2. Queries that obtain more results

Semantic Query	Ask for:	DSpace Equivalent
`dspace-ont:sponsorship value dspace-ont:European_Commission`	*All items/authors that are sponsored by a specific institution*	**Sponsor:** "European Commission"
`dcterms:type value dspace-ont:Presentation`	*All items of type "Presentation"*	**Type**: "Presentation"

3. Queries impossible through traditional search

a. Number Restrictions

Semantic Query	Ask for:	DSpace Equivalent
`dcterms:type min 2 dspace-ont:DspaceType`	*All items that have been characterized with at least two DSpace types (e.g., Article, Book Chapter, ...)*	*Not possible*
`dcterms:type min 2 owl:Thing`	*All items that have been characterized with at least two types (LOM, DSpace type, or other)*	*Not possible*
`dcterms:hasPart min 6` *Note: not explicitly declared but gives results because hasPart is the inverse property of isPartOf*	*All collections that contain at least 6 items*	*Not possible*
`inverse dspace-ont:author min 2`	*All authors that are contributors to at least 2 items*	*Not possible*
`dspace-ont:author min 4`	*All items having 4 or more authors*	*Not possible*

b. Other

Semantic Query	Ask for:	DSpace Equivalent
`dspace-ont:co_author some (foaf:surname value "Koutsomitropoulos")` `dspace-ont:co_author value dspace-ont:Koutsomitropoulos_Dimitrios`	*All co-authors of author with surname "Koutsomitropoulos"*	*Not possible (because of 'co_author')*

dcterms:relation some (foaf:surname value "Kalou") dcterms:relation value dspace-ont:Solomou_Georgia *Note: dcterms:relation is super property of dspace-ont:co_author*	*All things that are related to author named "Solomou"*	*Not possible*
dcterms:hasPart some(dcterms:type value dspace-ont:Learning_Object)	*All collections that contain learning objects*	*Not possible (nested class expression that allows retrieving precisely those collections that contain LOs)*

Corresponding author
Dr Dimitrios Koutsomitropoulos can be contacted at: kotsomit@ceid.upatras.gr