Evaluating Semantic Search Tools using the SEALS platform*

Stuart N. Wrigley[1], Khadija Elbedweihy[1], Dorothee Reinhard[2], Abraham Bernstein[2], and Fabio Ciravegna[1]

[1] University of Sheffield, Regent Court, 211 Portobello, Sheffield, UK
{s.wrigley, k.elbedweihy, f.ciravegna}@dcs.shef.ac.uk
[2] University of Zürich, Binzmühlestrasse 14, CH-8050 Zürich, Switzerland
{dreinhard, bernstein}@ifi.uzh.ch

Abstract. In common with many state of the art semantic technologies, there is a lack of comprehensive, established evaluation mechanisms for semantic search tools. In this paper, we describe a new evaluation and benchmarking approach for semantic search tools using the infrastructure under development within the SEALS initiative. To our knowledge, it is the first effort to present a comprehensive evaluation methodology for semantic search tools. The paper describes the evaluation methodology including our two-phase approach in which tools are evaluated both in a fully automated fashion as well as within a user-based study. We also present and discuss preliminary results from the first SEALS evaluation campaign together with a discussion of some of the key findings.

Keywords: semantic search, usability, evaluation, benchmarking, performance

1 Introduction

Searching the Semantic Web lies at the core of many activities that are envisioned for the Semantic Web; many researchers have investigated means for indexing and searching the Semantic Web. Semantic search tools are systems that take a query as their input, reason over some kind of knowledge base and return the compatible answers. The input *query* can take the form of a natural language question, a triple representation of a question, a graphical representation, keywords, etc. and the knowledge base can be one or more ontologies, annotated text corpora or plain text documents, etc. Similarly, the answers which are returned by a tool can take a multitude of forms from pure triples to a natural language representation.

In the area of semantic search there are a large number of different tool types focussing on the diverse aspects of this domain. In this evaluation work, we focus on user-centered tools for retrieving information and knowledge including those which support some kind of natural language user-interface. The core functionality of a semantic search tool is to allow a user to discover one or more facts

* This work was supported by the European Union 7th FWP ICT based e-Infrastructures Project SEALS (Semantic Evaluation at Large Scale, FP7-238975).

or documents by inputting some form of query. The manner in which this input occurs (natural language, keywords, visual representation) is not of concern; however, the *user experience* of using the interface is of interest. Indeed, we feel it is appropriate to directly compare tools with potentially differing interfaces since tool adopters (who may not have technical expertise in the semantic search field) will place significant emphasis on this aspect in their decision process. Therefore, it is essential that the evaluation procedures emphasise the user experience of each tool.

We believe semantic search is an area where evaluation is critical and one for which formalised and consistent evaluation has, until now, been unavailable. The evaluation of semantic search technologies is a core element of the Semantic Evaluation At Large Scale (SEALS) initiative which is aimed at developing a new research infrastructure dedicated to the evaluation of Semantic Web technologies. The SEALS Platform [5] provides facilities for storing all the materials required for an evaluation to take place: the tool(s), the test data, a results storage repository and a description of the evaluation workflow.

Two aspects, however, make the evaluation of search tools more complicated than the benchmarking employed for other types of Semantic Web tools (such as reasoners or matchers): first, *different search tools use highly varying querying metaphors* as exhibited by a range of searching approaches as alluded to above (e.g., keyword-based, language-based or graphical). Indeed, it has been decided that no restriction will be placed on the of type of interfaces to be assessed. In fact we hope as wide a range of interface styles will be evaluated as possible. Second, *the search task usually involves a human seeker*, which adds additional complexities into any benchmarking approach.

This paper describes an evaluation which comprises both an automated evaluation phase to determine retrieval performance measures, such as precision and recall as well as an interactive phase to elicit usability measures. Specifically, the evaluation is comprised of a series of reference benchmark tests that will focus on the performance of fundamental aspects of the tool in a strictly controlled environment or scenario rather than their ability to solve open-ended, real-life problems.

It is intended that the presentation of the methodology and the execution of the evaluation campaigns will spur on the adoption of this methodology serving as the basis for comparing different search tools and fostering innovation.

We will briefly describe previous evaluation initiatives before introducing our methodology in detail. We will also describe the two core datasets and the mechanisms for integrating tools with the evaluation software. Finally, we will present some preliminary evaluation results and conclude with a short discussion of these.

2 Previous Related Evaluations

Few efforts exist to evaluate semantic search tools using a comprehensive, standardised benchmarking approach. One of the first attempts at a comprehensive

evaluation was conducted by Kaufmann [6] in which four different question answering systems with natural language interfaces to ontologies were compared: *NLP-Reduce*, *Querix*, *Ginseng* and *Semantic Crystal*. The interfaces were tested according to their performance and usability. These ontology-based tools were chosen by virtue of their differing forms of input. NLP-Reduce and Querix allow the user to pose questions in full or slightly restricted English. Ginseng offers a controlled query language similar to English. Semantic Crystal provides the end-user with a rather formal, graphical query language.

Kaufmann [6, 7] employed a large usability study conducted for each of the four systems with the same group of non-expert subjects using the Mooney dataset (see Sec. 4.2) as the ontological knowledge base. The goal of this controlled experiment was to detect differences related to the usability and acceptance of the four varying query languages. The experiment revealed that the subjects preferred query languages expecting full sentences as opposed to separate keywords, menu-driven and graphical query languages — in this order. Therefore, it can be concluded that casual end-users favour query languages that support the formulation process of their queries and which structure their input, but do not over-restrict them or make them learn a rather unusual new way of phrasing questions.

Another previous evaluation [3] extensively benchmarked the K-Search system, both *in vitro* (in principle) and *in vivo* (by real users). For instance, the *in vivo* evaluation used 32 Rolls-Royce plc employees, who were asked about their individual opinions on the system's efficiency, effectiveness and satisfaction. However, as is common with small-scale evaluations, they refrained from comparing their tool with other similar ones in this domain.

3 Evaluation Design

This section describes the design of the evaluation methodology in detail. It introduces the core assumptions which we have made and the two-phase approach which we have deemed essential for evaluating the different aspects of a semantic search tool. We also describe the criteria and metrics by which the tools will be benchmarked and the analyses which will be made.

3.1 Two-Phase Approach

The evaluation of each tool is split into two complementary phases: the Automated Phase and the User-in-the-loop Phase. The user-in-the-loop phase comprises a series of experiments involving human subjects who are given a number of tasks (questions) to solve and a particular tool and ontology with which to do it. The subjects in the user-in-the-loop experiments are guided throughout the process by bespoke software – the *controller* – which is responsible for presenting the questions and gathering the results and metrics from the tool under evaluation. Two general forms of metrics are gathered during such an experiment. The first type of metrics are directly concerned with the operation of the tool itself such as time required to input a query, and time to display the results. The

second type is more concerned with the 'user experience' and is collected at the end of the experiment using a number of questionnaires.

The outcome of these two phases will allow us to benchmark each tool both in terms of its raw performance but also the ease with which the tool can be used. Indeed, for semantic search tools, it could be argued that this latter aspect is the most important. In addition to usability questionnaires, demographics data will be collected from the subjects enabling tool adopters to assess whether a particular tool is suited for their target user group(s).

3.2 Criteria

Query expressiveness While some tools (especially form based) do not allow complex queries, others (e.g., NLP-based approaches) allow, in principle, a much more expressive set of queries to be performed. We have designed the queries to test the expressiveness of each tool both formally (by asking participants in the evaluation to state the formal expressiveness) and practically (by running queries to test the actual coverage and robustness).

Usability Usability will be assessed both in terms of ability to express meaningful queries and in combination with large scale — for example, when a large set of results is returned or a very large ontology is used. Indeed, the background of the user may also influence their impression of usability.

Scalability Tools and approaches will be compared on the basis of ability to scale over large data sets. This includes the tool's ability to query a large repository in a reasonable time; its ability to cope with a large ontology; and its ability to cope with a large amount of results returned in terms of readability/accessibility of those results.

Performance This measures the resource consumption of a particular search tool. Performance measures (speed of execution) depend on the benchmark processing environment and the underlying ontology.

3.3 Metrics and Analyses

Automated Phase The metrics and interpretations used for tool evaluation in the automated phase draw on the work of Kaufmann [6]. A number of different forms of data will be collected each addressing a different aspect of the evaluation criteria.

A number of 'standard' measures are collected including the set of answers returned by the tool, the amount of memory used, etc. These metrics cover the query expressiveness and interoperability criteria described in Sec. 3.2:

- Execution success (OK / FAIL / PLATFORM ERROR). The value is *OK* if the test is carried out with no execution problem; *FAIL* if the test is carried out with some execution problem; and *PLATFORM ERROR* if the evaluation infrastructure throws an exception when executing the test.
- Results. This is the set of results generated by the tool in response to the query. This set may be in the form of a ranked list. The size of this set is determined (at design time) by the tool developer.

– Time to execute query. Speed with which the tool returns a result set. In order to have a reliable measure, it will be averaged over several runs.

For each tool, a large amount of raw metric data will be produced. From this, a number of interpretations can be produced which can be both presented to the community as well as be used to inform the semantic technology roadmaps which will be produced after each evaluation campaign. The automated phase is concerned with the interpretations concerning the 'low-level' performance of the search tool such as the ability to load ontology and query (interoperability) and the precision, recall and f-measure of the returned results (search accuracy and query expressiveness). The scalability criterion is assessed by examining the average time to execute query with respect to ontology size. Tool robustness is represented by the ratio between the number of tests executed and the number of failed executions.

User-in-the-loop Phase In order to address the usability of a tool, we also collect a range of user-centric metrics such as the time required to obtain the final answer, number of attempts before the user is happy with the result. In addition, data regarding the user's impression of the tool is also gathered using questionnaires (see Sec. 3.4).

For each topic / questions presented to the user, the following metrics are collected:

– Execution success (OK / FAIL / PLATFORM ERROR).
– Underlying query (in the tool's internal format; e.g., in SPARQL format)
– Results.
– Is the answer in the result set? It is possible that the experiment subject may have been unable to find the appropriate answer (even after a number of query input attempts). In this case, the subject would have indicated this via the controller software.
– User-specific statistics: time required to obtain answer; number of queries required to answer question; demographics; System Usability Scale (SUS) questionnaire [4]; in-depth satisfaction questionnaire

A small number of traditional interpretations will be generated which relate to the 'low-level' performance of the search tool (e.g., precision, recall and f-measure). However, the emphasis is on usability and the user's satisfaction when using the tool. This will be identified using the SUS score, the number of attempts made by the user, the time required to obtain a satisfactory answer as well as a number of correlations between usability metrics and other measures and / or demographics.

3.4 Questionnaires

For the user-in-the-loop phase we employ three kinds of questionnaires, namely the System Usability Scale (SUS) questionnaire [4], the Extended questionnaire

and the Demographics questionnaire. Such questionnaires represent a well-known and often applied procedure in the domain of Human Computer Interaction to assess the user satisfaction and to measure possible biases and correlations between the test subject characteristics and the outcomes of the evaluation.

SUS is a unified usability test comprising ten normalised questions (e.g., 'I think that the interface was easy to use,' 'I think that I would need the support of a technical person to be able to use this system,' etc.). The subjects answer all questions on a 5-point Likert scale identifying their view and opinion of the system. The test incorporates a diversity of usability aspects, such as the need for support, training and complexity. The final score of this questionnaire is a value between 0 and 100, where 0 implies that the user regards the user interface as unusable and that 100 implies that the user considers the user interface to be perfect. Bangor et al. [1] described the results of 2,324 SUS surveys from 206 usability tests collected over a ten year period and found that the SUS was a highly reliable indicator of usability (alpha = 0.91) for many different interface types (mobile phones, televisions as well as GUIs).

The Extended questionnaire includes further questions regarding the satisfaction of the users. These questions cover domains such as the design of the tool, the tool's query language, the tool's feedback, questions according to the performance and functionality of the tool and the user's emotional state during the work with the tool.

The Demographics questionnaire collects detailed demographic information regarding the participants which allow us to identify tools or types of tools which are better suited to particular types of users.

4 Datasets

For the first evaluation campaign we have taken the decision to focus on purely ontology-based tools. More complex test data (document-based, chaotic data, data with partially known schemas) will be considered for later evaluation campaigns. Indeed, the SEALS consortium actively encourages community participation in the specification of subsequent campaigns.

4.1 Automated Phase

EvoOnt[3] is a set of software ontologies and data exchange format based on OWL. It provides the means to store all elements necessary for software analyses including the software design itself as well as its release and bug-tracking information. For scalability testing it is necessary to use a data set which is available in several different sizes. In the current campaign, it was decided to use sets of sizes 1k, 10k, 100k, 1M, 10M triples. The EvoOnt data set lends itself well to this since tools are readily available which enable the creation of different ABox sizes for a given ontology while keeping the same TBox. Therefore, all the different sizes are variations of the same coherent knowledge base.

[3] http://www.ifi.uzh.ch/ddis/evo/

4.2 User-in-the-loop Phase

The main requirement for the user-in-the-loop dataset is that it be from a simple and understandable domain: it should be sufficiently simple and well-known that casual end-users are able to reformulate the questions into the respective query language without having trouble to understand them. Additionally, a set of questions are required which subjects will use as the basis of their input to the tool's query language or interface. The Mooney Natural Language Learning Data[4] fulfils these requirements and is comprised of three data sets each supplying a knowledge base, English questions, and corresponding logical queries. They cover three different domains: geographical data, job data, and restaurant data. We chose to apply only the geography data set, because it defines data from a domain immediately familiar to casual users. The geography OWL knowledge base contains 9 classes, 11 datatype properties, 17 object properties and 697 instances. An advantage of using the Mooney data for the user-in-the-loop evaluation is the fact that it is a well-known and frequently used data set (e.g., [6], [8] and [9]). Furthermore, its use allowed the possibility of making the findings comparable with other evaluations of tools in this area, such as *Cocktail* [9], *PANTO* [8] and *PRECISE* [8].

4.3 Test Questions

User-in-the-loop Phase The Mooney geography question set has been augmented using the existing questions as templates. In the question *'How many **cities** are in **Alabama**?'*, for example, the class concept *city* can be exchanged on the vertical level by other class concepts, such as *lake, mountain, river*, etc. Furthermore, the instances can be exchanged to obtain more questions. For example, *Alabama* could be replaced by any instance of the class *state* (e.g., *California, Oregon, Florida*, etc.). We also added more complicated questions that ask for more than one instance and produce more complex queries, such as *'What **rivers** run through the **state** with the **lowest point** in the USA?'* and *'What **state** bordering **Nevada** has the **largest population**?'*.

Automated Phase The EvoOnt data set comprises knowledge of the software engineering domain; hence, the questions will have a different character than the Mooney questions and make use of concepts like programming *classes, methods, bugs (issues), projects, versions, releases* and *bug reports*. Simpler questions will have the form *'Does the **class** x have a **method** called y?'* or *'Give me all the **issues** that were reported by the **user** x and have the **state** fixed?'*, where **x** and **y** are specific instances of the respective ontological concept. Examples for more complex questions that enclose more than three concepts are *'Give me all the **issues** that were reported in the **project** x by the **user** y and that are fixed by the **version** z?'* and *'Give me all the **issues** that were reported in the **project** w by the **user** x after the **date** y and were fixed by the **version** z?'*.

[4] http://www.cs.utexas.edu/users/ml/nldata.html

5 API

In order for a tool to be evaluated, the tool provider had to produce a tool 'wrapper' which implemented a number of methods[5]. This allowed the evaluation platform to automatically issue query requests and gather the result sets, for instance. Furthermore, exposing this functionality also allowed the user-in-the-loop experiment software to gather various forms of data during the user experiment that will be used for analysis.

The core functionality can be split into three different areas: methods required in both phases, methods required only for the user-in-the-loop phase and methods required just for the automated phase.

Functionality which is common to both evaluation phases include the method to load an ontology into the tool. The other methods are related to the results returned by the tool. The first determines if the tool manages (and hence returns via the API) its results as a ranked list. The second determines if the tool has finished executing the query and, consequently, the results are ready. The final method retrieves the results associated with the current query; the method returns URIs in the SPARQL Query Results XML Format[6].

Only one method is required specifically for the automated phase: execute query. This executes a query which has been formatted to suit an individual search tool's internal query representation. Three methods are required for the user-in-the-loop phase. The first determines if the user has finished inputting their query to the tool. The second retrieves the String representation of the query entered by the user. For example, if the tool uses a Natural Language interface, this method would simply return the text entered by the user. The final method retrieves the tool's internal representation of the user's query. This should be in a form such that it could be passed to the automated phase's execute query method and obtain the same results.

6 Evaluation results

This section presents the preliminary results and analyses from the first SEALS Evaluation Campaign which was conducted during Summer 2010. The list of participants and the phases in which they participated is shown in Table 1. Formal analysis of the results is still ongoing and is the subject of current and future work. However, these preliminary results contain a number of interesting points which merit discussion. Furthermore, it should be noted that for some tools, the formal evaluation is still ongoing; indeed, this is this case for PowerAqua hence no detailed results will be presented for this tool. Due to space constraints, we concentrate on the user-in-the-loop experiment results since this are the most interesting for benchmarking semantic search tools and obtaining an insight into what functionality users want from such a tool and whether or not the tools

[5] http://www.seals-project.eu/seals-evaluation-campaigns/semantic-search-tools/connect-your-tool

[6] http://www.w3.org/TR/rdf-sparql-XMLres/

Table 1. Evaluation campaign participating tools. The last two columns indicate if the tool participated in the user-in-the-loop (UITL) and automated (Auto) phases.

Tool	Description	UITL	Auto
K-Search	K-Search allows flexible searching of semantic concepts in ontologies and documents using a form-based interface.	x	
Ginseng	Guided Input Natural Language Search Engine (Ginseng) is a natural language interface question answering system.	x	x
NLP-Reduce	NLP-Reduce is a natural language query interface that allows its users to enter full English questions, sentence fragments, and keywords.	x	x
Jena Arq	ARQ is a query engine for Jena that supports the SPARQL RDF Query language. This tool has been used as a 'baseline' for the automated phase		x
PowerAqua	PowerAqua is an open multi-ontology Question Answering (QA) system for the Semantic Web (SW) using a Natural Language (NL) user interface.	x	x

included in this campaign meet those requirements. As described in Sec. 3.4, the subjects in each experiment provided feedback via questionnaires which will also be discussed.

6.1 Results and discussion

The user-in-the-loop evaluation results are shown in Table 2. In order to facilitate the discussion, the responses to each of the twenty questions by all users, along with the average experiment time and feedback scores have been averaged[7].

The *mean experimental time* indicates how long, on average, the entire experiment (answering twenty pre-defined questions) took for each user. The *mean SUS* indicates the mean system usability score for each tool as reported by the users themselves. The *mean extended questionnaire* shows the average response to the questionnaire in which more detailed questions were used to establish the user's satisfaction and is scored out of 5[8]. The *mean number of attempts* shows how many times the user had to reformulate their query using the tools interface in order to obtain answers with which they were satisfied (or indicated that they were confident a suitable answer could not be found). This latter distinction between finding the appropriate answer after a number of attempts and the user 'giving up' after a number of attempts is shown by the *mean answer found rate*. *Input time* refers to the amount of time the subject spent formulating their query using the tool interface before submitting the query.

The results show that the difference in perceived usability between K-Search and Ginseng is not significant – their SUS scores are almost identical – whereas

[7] Extended results and analysis for both the user-in-the-loop and automated phases will be available from the SEALS website from December 2010.

[8] For details of the questions used in the extended questionnaire, download the experiment pack from http://www.seals-project.eu/seals-evaluation-campaigns/semantic-search-tools/experiment-pack

Table 2. User-in-the-loop tool performance

Criterion	K-Search	Ginseng	NLP-Reduce
Mean experiment time (s)	1 hr 11 mins 54 s	1hr 0 mins 12 s	1 hr 19 mins 59 s
Mean SUS (%)	44.38	40	25.94
Mean ext. questionnaire (%)	47.2	45	44.6
Mean number of attempts	2.37	2.03	5.54
Mean answer found rate	0.41	0.19	0.21
Mean execution time (s)	0.44	0.51	0.51
Mean input time (s)	69.11	81.63	29
Max input time (s)	300.17	300.16	278.95
Mean overall question time (s)	257.25	216.19	246.95
Mean precision	0.44	0.32	0.16
Mean recall	0.61	0.32	0.55
Mean f-measure	0.46	0.27	0.21

the SUS score for NLP-Reduce is much lower. It is also evident that none of the tools received a score which indicates satisfactory user experience. Bangor et al. [2] associated 'adjective ratings' to the SUS score. According to these adjective ratings, both K-Search and Ginseng fall into the *Poor* to *OK* ratings and NLP-Reduce being classified as *Awful* (see Table 3 in [2]). This is confirmed by the details of the recorded user behaviour. For instance, for K-Search and Ginseng, subjects required more than two attempts to formulate their query before they were satisfied with the answer or moved on. Subjects using NLP-Reduce, however, required more than five attempts – twice that of the other tools. Users of K-Search found satisfactory answers twice as often as those who used Ginseng and NLP-Reduce which is supported by the higher f-measure score for K-Search compared with the other tools.

This usability performance is supported both by the low extended questionnaire results and also the feedback which was collected from each of the experiment subjects. This is interesting since despite the tools using different interface approaches (form-based versus natural language) neither provided the flexibility desired by the subjects. When using K-Search, many subjects reported that they liked the interface and particularly the ability to 'see the ontological concepts and relations between concepts easily' thus allowing 'the user to know just what sort of information is available to be retrieved from the system'. However, the rigid framework of a form-based interface was also the cause of many of the subjects' dislikes. K-Search provided no mechanism for negation: it was not possible to formulate queries to answer questions such as *Tell me which rivers do not traverse the state with the capital nashville?*. Furthermore, while the form-based approach allows the creation of queries containing multiple concepts, it was not clear how these related to each other. For instance, one subject reported that 'if I had 3 states on my form and i added a *hasCity* relation it was not obvious which state should have the city'.

Natural language interfaces are often promoted as a more flexible way of entering a query than keyword- or form-based approaches. However, this provides

a significant challenge to such tools: how to cope with the vast range of possible ways of formulating a query. For instance, Ginseng employs a commonly used solution: restrict the vocabulary and/or grammar which can be used for query entry. The use of a very restricted language model can resemble 'autocompletion' when creating simple queries. Subjects liked the speed with which (simple) queries could be entered; however, difficulties arose with more complex questions. Subjects reported that the language model could 'railroad' them into a particular direction. In this situation, it was commonly acknowledged that the only alternative was to start again. Furthermore, it was sometimes unclear to subjects as to which suggested search terms related to which ontological concepts leaving subjects confused. The language model (or underlying query engine) of Ginseng did not allow comparative queries using terms such as 'biggest' or 'smaller than'. Although not employing a restrictive language model, NLP-Reduce suffered from similar criticisms as Ginseng regarding it's NL input – largely due to the naïve underlying NLP engine. Indeed, as the SUS score indicates, the subjects found it much harder to use; for instance, the tool didn't allow the use of superlatives and subjects commonly reported that the tool didn't understand what they had entered, thus forcing the subject to start again (hence NLP-Reduce having twice the number of attempts).

Finally, a commonly reported deficiency of all the tools was the manner in which a query's results could be managed or stored. Since a number of the questions used in the experiment had a high complexity level and needed to be split into two or more sub-queries, subjects reported that they would have liked to have either used previous results as the basis of the next query or to have simply temporarily stored the results to allow some form of intersection or union operation with the current result set.

7 Conclusions

This paper has presented a methodology for the evaluation of any semantic search tool regardless of its user interface. A critical aspect of semantic search tool benchmarking is the user's experience of using the tool. Search is a user-centric activity and without a formalised evaluation of the tool's interface, only a limited insight into a tool's applicability to a particular task can be gained. Therefore, we adopted a two phase approach: an automated phase and a user-in-the-loop phase. This approach has impacted all aspects of the evaluation methodology: the criteria and metrics, the datasets and the analyses have all had to have been carefully chosen to accommodate the two phases. Indeed, in many cases, each phase is distinct (for example, each phase has its own, distinct, dataset).

As can be seen in the results section, the evaluation has provided a rich source of data – only a small amount of which we have been able to present in this paper. It is clear that users of search tools have very high expectations of their performance and usability. The pervasive use of web search engines, such as Google, condition the way in which non-expert users view search; indeed,

a number of subjects in the user-in-the-loop experiment compared the tools (unfavourably) to Google. However, with respect to the results, many subjects reported they wanted a much more sophisticated management (and subsequent additional querying) of the result set rather than the traditional list of answers and simple query refinement.

The identification of such deficiencies in current search technologies is the purpose of the SEALS benchmarking initiative and will help drive the technology to meet the needs of the user. Furthermore, the regular SEALS evaluation campaigns will help monitor this progress and, as the benchmarking approaches become increasingly sophisticated, provide increasingly detailed insights into the technology and user interfaces employed.

The results and analyses presented in this paper are preliminary and a more detailed study of the results is currently underway. Indeed, the first campaign has acted as an evaluation not only of the participating tools but of the methodology itself. This is the first evaluation of its kind and the experiences of organising and executing the campaign, as well as feedback from the participants, will help improve the methodology and organisation of future campaigns.

References

1. A. Bangor, P. T. Kortum, and J. T. Miller. An empirical evaluation of the system usability scale. *International Journal of Human-Computer Interaction*, 24(6):574–594, 2008.
2. A. Bangor, P. T. Kortum, and J. T. Miller. Determining what individual sus scores mean: Adding an adjective rating scale. *Journal of Usability Studies*, 4(3):114–123, 2009.
3. R. Bhagdev, S. Chapman, F. Ciravegna, V. Lanfranchi, and D. Petrelli. Hybrid search: Effectively combining keywords and semantic searches. In *The Semantic Web: Research and Applications*, pages 554–568. Springer Berlin / Heidelberg, 2008.
4. J. Brooke. SUS: a quick and dirty usability scale. In P. W. Jordan, B. Thomas, B. A. Weerdmeester, and I. L. McClelland, editors, *Usability Evaluation in Industry*, pages 189–194. Taylor and Francis, 1996.
5. M. Esteban-Gutiérrez, R. García-Castro, and A. Gómez-Pérez. Executing evaluations over semantic technologies using the seals platform. In *International Workshop on Evaluation of Semantic Technologies (IWEST 2010), ISWC 2010*, 2010.
6. E. Kaufmann. *Talking to the Semantic Web — Natural Language Query Interfaces for Casual End-Users*. PhD thesis, Faculty of Economics, Business Administration and Information Technology of the University of Zurich, September 2007.
7. E. Kaufmann and A. Bernstein. How useful are natural language interfaces to the semantic web for casual end-users? In *ISWC/ASWC*, pages 281–294, 2007.
8. A.-M. Popescu, O. Etzioni, and H. Kautz. Towards a theory of natural language interfaces to databases. In *IUI '03: Proceedings of the 8th international conference on Intelligent user interfaces*, pages 149–157, New York, NY, USA, 2003. ACM.
9. L. R. Tang and R. J. Mooney. Using multiple clause constructors in inductive logic programming for semantic parsing. In *In Proceedings of the 12th European Conference on Machine Learning*, pages 466–477, 2001.