Methodology and Campaign Design for the Evaluation of Semantic Search Tools

Stuart N. Wrigley
University of Sheffield
Regent Court, 211 Portobello
Sheffield, UK
s.wrigley@dcs.shef.ac.uk

Dorothee Reinhard
University of Zürich
Binzmühlestrasse 14
CH-8050 Zürich, Switzerland
dreinhard@ifi.uzh.ch

Khadija Elbedweihy
University of Sheffield
Regent Court, 211 Portobello
Sheffield, UK
k.elbedweihy@sheffield.ac.uk

Abraham Bernstein
University of Zürich
Binzmühlestrasse 14
CH-8050 Zürich, Switzerland
bernstein@ifi.uzh.ch

Fabio Ciravegna
University of Sheffield
Regent Court, 211 Portobello
Sheffield, UK
f.ciravegna@dcs.shef.ac.uk

ABSTRACT

The main problem with the state of the art in the semantic search domain is the lack of comprehensive evaluations. There exist only a few efforts to evaluate semantic search tools and to compare the results with other evaluations of their kind.

In this paper, we present a systematic approach for testing and benchmarking semantic search tools that was developed within the SEALS project. Unlike other semantic web evaluations our methodology tests search tools both automatically and interactively with a human user in the loop. This allows us to test not only functional performance measures, such as precision and recall, but also usability issues, such as ease of use and comprehensibility of the query language.

The paper describes the evaluation goals and assumptions; the criteria and metrics; the type of experiments we will conduct as well as the datasets required to conduct the evaluation in the context of the SEALS initiative. To our knowledge it is the first effort to present a comprehensive evaluation methodology for Semantic Web search tools.

Categories and Subject Descriptors

H.3.4 [**Information Systems**]: Systems and Software—*Performance evaluation (efficiency and effectiveness)*; H.5.2 [**Information Interfaces and Presentation**]: User Interfaces—*Evaluation/methodology, Benchmarking*; D.2.8 [**Software Engineering**]: Metrics—*performance measures*

General Terms

Measurement, Performance, Design, Experimentation, Human Factors, Standardization

Keywords

semantic search, semantic query, usability, evaluation, benchmarking, performance measure

WWW2010, April 26-30, 2010, Raleigh, North Carolina.

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 [1–4,6,9,13,14,16]. As a consequence, the evaluation of such search tools is a very important topic. Indeed, search evaluation is a core element of the Semantic Evaluation At Large Scale (SEALS) EU project[1], which is aimed at developing a new research infrastructure dedicated to the evaluation of Semantic Web technologies.

At the heart of the SEALS project is the largely automated evaluation of five semantic technologies (ontology engineering tools, storage and reasoning systems, ontology matching tools, semantic web service tools and semantic search tools). Clearly, for evaluation results to be comparable between executions there must exist a means of controlling for the hardware upon which the benchmarking has been performed – differing architectures, CPU speeds and memory availability will all affect the outcome. In order to address this issue, the SEALS initiative provides dedicated compute resources upon which the evaluations will be run – the SEALS Platform – using virtualisation technology to provide access to all major operating systems.

The SEALS Platform 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. The latter provides a means of specifying evaluations in a declarative way; specifically, we use the executable Business Process Execution Language (BPEL). A BPEL execution engine, provided as part of the SEALS Platform, allows the execution of an evaluation (or indeed a full campaign) with no recourse to human intervention. A consequence of this design is that each tool provider must also submit a lightweight wrapper which exposes the necessary functionality required to perform the evaluation. This functionality is specified in the form of an API (see Sec. 6.1).

The SEALS project will organise two full evaluation campaigns (summer 2010 and late 2011 / early 2012). However, it ought to be noted that the SEALS Platform is available for

[1] http://www.seals-project.eu/

use by the community at any time and use is not restricted to formal evaluation campaigns. Indeed, the availability of the SEALS Platform will extend beyond the lifetime of the SEALS project to provide a permanent service for evaluation execution and test data storage and will be administered by the community.

It is important to emphasise that the purpose of the SEALS evaluation campaigns is to evaluate semantic technologies with respect to their semantic peers as opposed to the wider topic of comparing semantic and non-semantic technologies.

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 plethora of searching approaches (e.g., keyword-based, language-based or graphical). Second, *the search task usually involves a human seeker*, which adds additional complexities into any benchmarking approach.

The main contribution of this paper is a methodology which can be applied to the evaluation of all semantic search tools regardless of their search metaphor. To that end, it 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. These fundamental aspects are the formal evaluation criterion by which we will benchmark each tool. More details are enumerated in Sec. 4.3; however, the core criterion are:

- Query expressiveness
- Usability (effectiveness, efficiency, satisfaction)
- Scalability
- Quality of documentation
- Performance

In the context of SEALS we will organise a series of evaluation campaigns following the methodology described here with the intention of providing a discussion forum on semantic search technology evaluation and objective evaluation results.

Hopefully, the presentation of the methodology and the evaluation campaigns will spur on the adoption of this methodology serving as the basis for comparing different search tools and fostering innovation. The comparisons elicited by the evaluations will also provide the basis for establishing technology 'roadmaps', which can be used by technology adopters to guide and inform the selection of technologies most appropriate to their needs. Therefore, this paper not only describes the types of experiments we will conduct (Sec. 4) but also provides details of what is expected from technology providers (Sec. 6) and what technology adopters can expect to have access to once the evaluation will be completed (Sec. 7).

In the remainder of this paper, we will provide a brief overview of the various semantic search tool types and highlight the specific kind that forms the focus of this evaluation campaign. We will also mention previous evaluation initiatives before introducing our detailed methodology. Further, we will describe the two core datasets that we adopted and subsequently move on to the requirements for the participation. Finally, we present the various ways in which we intend to publish the evaluation results.

2. SEARCH TOOL TECHNOLOGIES

Semantic search tools are systems that take a query as their input, reason over some kind of database and return the compatible answers.

The input *query* can consist of, for example, a natural language question, a triple representation of a question, a graphical representation, keywords, phrases, etc. The databases can be ontologies, annotated text corpora, plain text documents, the Web, XML documents, RDF documents, HTML documents and so on. The returned answers can either represent pure triples, a natural language representation of the triples, ranked lists of answer triples or terms, graphical representations, links to websites, links to text documents, links to RDF documents, text excerpts, ontological instances, complete natural language sentences, etc.

In the area of semantic search there are a large number of different tool types focussing on the diverse aspects of this domain. These approaches can be divided into three main groups according to their core scope of research: tools that specialise in semantic search based on structural query languages, like *SPARQL* (e.g., *ARQ*[2]); tools for locating ontologies on the web (e.g., *Swoogle*[3] [8]); and tools which apply user-centered approaches for retrieving information and knowledge (e.g., *SemSearch* [16]). Such tools feature specific characteristics from a casual end-user point of view:

1. Structural query language tools require their users to be capable of applying a specific kind of query language implying a knowledge of logic-driven languages, which usually cannot be expected from casual end-users. Moreover, they usually do not preprocess the results since they generally do not support any kind of natural language user interface.
2. User-centered tools for retrieving ontological data on the web, however, usually do support some kind of natural language user interface (mainly supporting keyword-search). These tools allow users to locate the URIs of ontological documents like RDF, XML, OWL, etc. They are also called *semantic search engines*, because they often provide a mechanism for refining the search by applying ontological concepts, such as *Person, Organisation*, etc. However, they typically deliver only the URI of a search result term rather than specific details about it and do not further process the located answer into a natural language representation.
3. User-centered tools for retrieving information and knowledge, likewise, support some kind of natural language user-interface. However, such interfaces generally feature the recognition of grammatically richer natural language phrases or sentences. Furthermore, these semantic search tool types mostly offer a preprocessing step for the retrieved results into a natural language representation. They mainly focus on the interpretation and presentation of the knowledge in the results in a user-friendly format. Such tools do not demand the knowledge of a complex query language from the user,

[2] http://jena.sourceforge.net/ARQ/
[3] http://swoogle.umbc.edu/

nor simply present whole documents that the user has to manually read through and analyse in order to find the correct answer.

In this methodology we have decided to focus on the third type of tool since we believe it is an area where evaluation is necessary, especially when it comes to measuring the usability of a tool (see, for example, Kaufmann et al. [13]) and in terms of query effectiveness, particularly when confronted with large-scale problems and rather complex queries. Further, this is a domain where a large number of tools are available in the community and therefore the community is very likely to participate.

This group of user-centered tools for retrieving information and knowledge can be further subdivided into the following categories [6]:

- **Keyword-based approaches** considering a natural language query as a bag of words (e.g., *NLP Reduce*; [16]);
- **Natural language approaches**: trying to model the linguistics of the query (e.g., *AquaLog*; [14]);
- **Graph-based approaches** matching terms in the ontology using a graph-based interface (e.g., *Semantic Crystal* [3], *SEWASIE*);
- **Form-based approaches** (e.g., *Corese*[4]);
- **Hybrid approaches** (e.g., *K-Search*; [6]).

Naturally, these various tool types differ from the point of view of both the input processing methodology and the output presentation strategy and format. Some only accept keywords or phrases as their input, others only whole sentences or a graphical representation of the query, some tools even combine several approaches with each other. Indeed, some tools even guide the user during the query input process by proposing possible (i.e., valid) concepts extracted from the vocabulary of an ontology, e.g., *Ginseng* [4].

Likewise, the approaches the tools apply in order to retrieve their results vary. Some tools make use of information retrieval methods, like pattern matching, information extraction and reasoning as well as logic approaches for detecting the correct answer. Other tools crawl the web as a huge corpus of documents as opposed to a further group of tools which analyses large corpora of text documents, such as news or Wikipedia articles. Another class of tools queries standard databases, consults XML-databases or applies one or more semantic knowledge databases, like ontologies, in order to search for a result. Some tools even employ machine translation techniques in order to embed a feature for answering multilingual questions. Furthermore, there exist hybrid systems that apply a combination of all such various methods and kinds of knowledge sources, e.g., *K-Search*.

3. 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 [11] which describes the comparison of four different question answering systems with natural language interfaces to ontologies, namely *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 [11,12] chose the Mooney dataset (see Sec. 5.2) as the ontological knowledge base for this study, since they include geographical domain knowledge about the USA which provides a simple, well-known, easily understandable and limited area which enables casual end-users to pose questions quickly; additionally, a large predefined set of natural language test questions was already available. Furthermore, its use allowed the possibility of making the findings comparable with other evaluations of tools in this area, such as *Cocktail* [18], *PANTO* [17] and *PRECISE* [17].

Kaufmann [11, 12] employed a large usability study conducted for each of the four systems with the same group of subjects. The goal of this controlled experiment was to detect differences related to the usability and acceptance of the four varying query languages. The subject group consisted of 48 casual end-users from various backgrounds, professions and levels of acquired previous knowledge. 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 [6] 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 systems' efficiency, effectiveness and satisfaction. This study aimed at measuring the users' comprehension level of the Hybrid Search paradigm, i.e. the quality of the knowledge retrieval and the users' judgement of the returned answers' adequacy, i.e. the quality of the document retrieval [6, p.10].

The usability testing sessions took about 90 minutes per subject. At first, the users were presented a short introduction to the system. Then, they had to perform an assisted training task in order to become familiar with the functionality and characteristics of the K-Search search tool as well as the general principle of Hybrid Search. In the next section of the study, the subjects were asked to carry out a second task. This time, however, without any help from the test leader. Additionally, the participants were requested to suggest and conduct a third task reflecting their work experience and interests. Finally, the participants were asked to fill in a user satisfaction questionnaire. After the study, a short interview on the subjects' experiences was carried out [6, p.12].

However, they refrained from comparing their tool with other similar ones in this domain.

4. 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

[4] http://www-sop.inria.fr/acacia/soft/corese/

semantic search tool. We also describe the criteria and metrics by which the tools will be benchmarked and the analyses which will be made.

4.1 Assumptions

The evaluation of third-party software tools in general requires a number of assumptions to be made in order to define the remit of the campaign. Specifically, the evaluation of semantic search tools will be subject to the following assumptions:

- Part of the evaluation will be automated: a number of aspects of semantic search tools can be evaluated in a fully offline, batch process. Specific aspects include the speed of response, the amount of memory used, the response precision to a particular query, etc.
- Part of the evaluation cannot be automated: a fundamental aspect of the evaluation campaign to be conducted is assessing the usability of the tool by a human subject. Search is an inherently interactive process and the ability of the user to interact easily with a tool and extract the required information quickly and effectively will be of great interest to tool adopters needing to select appropriate technologies for their specific needs.
- Only OWL ontologies will be used as test data: in order to simplify the development of the benchmarks it has been decided that search tools operating on purely OWL ontologies will be evaluated.
- We impose no restriction of type of interfaces to be assessed. In fact we hope as wide a range of interface styles will be evaluated as possible.
- All tools must implement the SEALS Search API[5].

4.2 Two-Phase Approach

The core functionality of a semantic search tool is to allow a user to discover one or more facts 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.

In order to achieve this goal, 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. 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.

[5]The SEALS Search API allows a well-defined mechanism for automated interaction between the tool and the evaluation platform. See Sec. 6.1 for more details.

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).

It has been decided that for our first evaluation the responsibility for running the user-in-the-loop experiments will rest with the respective tool providers (i.e., they will evaluate their own tool and only their own tool). This is largely for pragmatic reasons: it would be too time-consuming and costly to conduct full user evaluations for each submitted tool by any centralised party. Furthermore, the design of the user evaluations that are run at the participant's site benefits the community: a tool's usability can be assessed independently of a formal evaluation campaign. In order to support the tool providers, all necessary software required for controlling the user-in-the-loop experiments will be provided.

Indeed, since all the user-in-the-loop materials – software, instructions, questionnaires – will be publicly available on the SEALS Platform along with the ability to upload and analyse user-in-the-loop results, it is hoped that this approach to usability testing could become a standard approach. Furthermore, none of the materials will be coupled with any particular ontology thus improving its flexibility.

Obviously, this design choice limits the validity of the results. First, we will not be able to make any within subjects tests. Second, we will have to rely on the tool provider's diligence to ensure proper adherence to the testing procedure. These limitations do pose a threat to the validity of the outcome that may lead us to change this practice in future evaluation campaigns.

4.3 Criteria

The evaluation of user-centred search methodologies will be assessed according to the following 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. However, this is just in theory, as no one has ever provided a formalisation of the expressiveness of the language covered. This is a well-known problem in NLP interfaces: queries can be very short and cryptic, and therefore the logical complexity interferes with the complexity of the linguistic expression. Other approaches, such as K-Search, define formally the expressiveness of the covered language, and also define specific limitations due to usability reasons. We will design 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. Kaufmann et al. [13] have shown how an NLP-based interface can be interesting for a final user. Bhagdev et al. [6] have shown how technical users liked the hybrid form based interface. 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 (see below).

Scalability. Tools and approaches will be compared on the basis of ability to scale over large data sets; scalability can be considered along three dimensions:

1. Ability to query a large repository in a reasonable time: this measures the ability of the underlying query mechanism to access a very large store when confronted with queries; we will inspect scalability against progressively large stores (e.g., 1k, 10k, 100k, 1M, 10M triples).
2. Ability to cope with a large ontology; this measures the usability of the tool; for example, form-based interfaces are well known to become difficult to use when the ontology contains thousands of terms, while NLP-methods may perform better; we will use cases where ontologies contain dozens, hundreds or thousands of terms.
3. Ability to cope with a large amount of results returned; again, measures usability (in terms of readability/accessibility of results). Some queries will be designed to return a large amount of results.

Quality of documentation. We will test if the natural language of the tool's documentation is easy to understand and well structured.

Performance. This measures the resource consumption of a particular search tool (both during query execution and when the tool is 'at rest'). Performance measures depend on the benchmark processing environment and the underlying ontology. Metrics such as execution time (speed), CPU load and amount of required memory are usually considered to measure performance.

4.4 Metrics and Analyses

As mentioned in Sec. 4.2, the evaluation of semantic search tools will be conducted using a two phase approach, with each phase operating in parallel. One phase will address the evaluation of criterion which can be performed in an offline, non-interactive manner (the 'automated' phase). This phase will address query expressiveness, scalability and quality of documentation. The other phase will address the evaluation of criterion which require a real user to be using the tool interactively with specific search goals (the 'user-in-the-loop' phase). This phase will address usability and query expressiveness. Note that both phases address a number of common criterion.

4.4.1 Automated Phase

The metrics and interpretations used for tool evaluation in the automated phase draw heavily on the work conducted by Kaufmann [11]. 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 answer triple set returned by the tool, the amount of memory used, etc. These metrics cover the query expressiveness and interoperability criteria described in Sec. 4.3:

- 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.
- Triples returned. 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 is measured by the amount of time taken by the tool return a result set. In order to have a reliable measure, it will be averaged over several runs.
- CPU load. Similarly, the CPU load during query execution will be assessed which, in combination with the specification of the evaluation platform will inform the computational requirements of the tool.
- Memory usage. This assesses the amount of memory required for the search tool to run and the additional memory required to execute a query. This will be dependent on the complexity of the ontology but will be comparable between tools executing the same query-ontology combination.

For each tool, a large amount of raw metric data will be produced. From this, it is intended to produce a number of interpretations 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:

- Ability to load ontology and query (interoperability)
- Precision and Recall (search accuracy and query expressiveness)
- ROC curves [10] (search accuracy and query expressiveness independent of the prior distribution of answers)

The scalability criterion will be assessed using the following interpretations:

- Average time to execute query with respect to ontology size
- Average CPU load to execute query with respect to ontology size
- Average memory to execute query with respect to ontology size

Tool robustness will be represented by the ratio between the number of tests executed and the number of failed executions. A defect in a tool could affect several tests. Nevertheless, we do not propose an alternative since the desired situation is that the tool poses no execution problems.

4.4.2 User-in-the-loop Phase

In common with the automated phase (Sec. 4.4.1), a number of 'standard' measures are collected.

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. The first questionnaire is a standardised usability test called the System Usability Scale (SUS) questionnaire [7]. The second questionnaire investigates the user's

satisfaction in more detail. The final questionnaire collects a range of demographic information which will allow us to investigate any correlations between particular user groups and performance of individual tools (or types of tools). See Sec. 4.5 for more details on each questionnaire.

For each topic / questions presented to the user, it is envisaged that the following metrics will be collected:

- Execution success (OK / FAIL / PLATFORM ERROR).
- Query captured by the tool (the text typed into the NL query box or items clicked in a visual search tool, etc.)
- Underlying query (e.g., in SPARQL format)
- Triples returned.
- 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 clicks
  - number of queries required to answer question
  - demographics
  - System Usability Scale (SUS) score
  - 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, ROC curves). However, since much of the emphasis is on usability and the user's satisfaction when using the tool, a large number of correlations between user demographics and tool usability will be investigated.

The query expressiveness, accuracy and scalability criteria will be interpreted using the following interpretations:

- Precision/recall from triples returned for each query
- ROC curves based on the result set
- Search performance relative to ontology size

Usability will be interpreted using the following interpretations:

- Correlations between user demographics and measures, e.g.:
  - Number of correct tasks/questions/queries versus time to complete
  - Number of correct tasks/questions/queries versus years worked
  - Satisfaction versus number correct
  - Satisfaction versus years worked
- Correlations between demographics and other interpretations. e.g.:
  - Query time versus gender
  - Query time versus age
  - Query time versus knowledge of informatics, linguistics, formal query languages
  - Query time versus SUS score
  - SUS score versus number queries
  - SUS score versus success rate
  - SUS score versus knowledge of informatics, linguistics, formal query languages

Tool robustness will be computed in the same way as for the automated phase.

4.5 Questionnaires

For the user-in-the-loop phase we will employ three kinds of questionnaires, namely the SUS questionnaire, 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.

Therefore, after completing the usability experiment for a particular search tool, the user will be asked to answer a System Usability Scale (SUS) questionnaire [7]. 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 totally useless and 100 that the user considers the user interface ideally useful.

The Extended questionnaire will include further questions regarding the satisfaction of the users. These questions will, for example, cover domains like 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 will collect detailed demographic information regarding the participants. Demographics, or demographic data, are the characteristics of a population; it is common to combine several variables to define a 'demographic profile'. A demographic profile (often referred to as 'a demographic') provides enough information about the typical member of this group to create a mental picture of this hypothetical aggregate. As such, we intend to investigate correlations between particular demographics and the tool performance in order to identify tools or types of tools which are better suited to particular types of users.

It ought to be noted that an ethics approval for the user-in-the-loop experiments must be sought by each participating organisation including explicit permission for processing of this data and publishing of anonymised and/or aggregated analyses.

4.6 Documentation

An additional aspect of a tool's ease of use is the quality of the documentation. One purpose of the SEALS initiative is to provide technology roadmaps to potential technology adopters who may not be experts in the field. Clearly a tool which has poor quality instructions and documentation will prove significantly more difficult to deploy and use.

Tools already exist which determine if the language of a document is complicated or cumbersome. An example for this kind of language checking utility is the prototype tool *quZILLA* [5]. It was developed to automatically check

the description quality of bug reports. The tool is based on the ECLIPSE guidelines on how to write good bug reports[6]. *quZILLA* is implemented in PYTHON and employs the NLTK toolkit[7] for Natural Language Processing (NLP). The tool accepts bug descriptions as its input, which are first of all preprocessed via tokenisation and stemming techniques. In order to automatically define the quality score of the bug report, *quZILLA* employs several criteria (see Bettenburg et al. [5, p. 24] for more details). We will employ a similar approach for assessing the quality of a tool's instructions and/or documentation.

5. DATASETS

For the first evaluation campaign we have taken the decision to focus on purely ontology-based tools. This pragmatic decision was taken since the development and testing of the infrastructure (SEALS Platform and controller) will be conducted in parallel with the evaluation campaign. More complex test data (document-based, chaotic data, data with partially known schemas) will be considered for the second evaluation campaign. Indeed, the SEALS consortium actively encourages community participation in the specification of subsequent campaigns.

In contrast to the TREC initiative, which supports the evaluation of search systems based on full text documents, our evaluation campaign focuses on the evaluation of systems based on knowledge databases. Therefore, we are not able to apply the TREC evaluation methods. Furthermore, in our evaluation we introduce the usability metric that TREC does not take into account. Since TREC, for example, only accepts full text search systems, we are not able to employ their data sets for our evaluation of systems based on ontological data. In light of this, we opted for two different data sets which are described below.

5.1 Automated Phase

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.

EvoOnt 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.

EvoOnt[8] is divided into three different models that encapsulate the various aspects of object-oriented software source code, namely the Software Ontology Model (**SOM**), the Bug Ontology Model (**BOM**) and the Version Ontology Model (**VOM**). The models satisfy various meta-data information levels of the software engineering domain, i.e. they reflect, on the one hand, the design and architecture of the software and, on the other hand, further save information collected over time. Examples for such meta-data information are entries about revisions, releases, bug reports, etc. (cf. [15]).

The EvoOnt data set was created by a tool developed in the EvoOnt project. It allows for creating different sizes of the ABox of the EvoOnt ontology based on the same TBox. The tool imports various versions of the same software project, namely the compare plugin of the Eclipse project. The 1000 triple sized EvoOnt data file comprises the three different EvoOnt ontologies and no instances. The second file (10K triples) includes the same Tbox plus the Bugzilla Information (bugs). The third file (100K) adds several versions of the Version Information (cvs). Finally, the fourth (1M) and fifth file (10M) add more versions of the Version Information (cvs).

5.2 User-in-the-loop Phase

Since the focus of this phase of the evaluation is that of usability, the main requirement for the data set 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.

A data set which conforms to these constraints is the Mooney Natural Language Learning Data[9] which has been converted into an OWL-Lite ontology.

The *Mooney* data comprises three data sets each supplying a knowledge base, English questions, and corresponding logical queries. They pertain three different domains: geographical data, job data, and restaurant data. 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., [11], [17] and [18]). Furthermore, it is rather difficult to find good benchmark data sets allowing to work with ontology-based natural language interfaces (NLIs).

From the three data sets we chose to apply only the geography data set, because it defines data from a very simple and common domain. Such a domain is, firstly, much more appropriate for subjects in the usability study, since the questions are easily understandable. Secondly, questions in this domain are already freely available.

To make the original knowledge bases accessible to the ontology-based interfaces, Kaufmann [11] translated the Prolog knowledge bases to OWL and designed a class structure as meta model for each of the three domains. The resulting geography OWL knowledge base contains 9 classes, 11 datatype properties, 17 object properties and 697 instances. Each data set also comprises data-appropriate English questions, which were composed by undergraduate students of the computer science department of the University of Texas and gathered from 'real' people using a Web interface provided by Mooneys research group. There are 877 natural language questions for the geography knowledge base. For each question, there is also a corresponding logical representation of the question stated as Prolog terms in the data set.

The three distinct OWL and text files of the formatted Mooney data include the predefined questions for each domain and are available for download[10].

[6] https://bugs.eclipse.org/bugs/bugwritinghelp.html
[7] http://nltk.sourceforge.net/index.php/Main_Page
[8] http://www.ifi.uzh.ch/ddis/evo/

[9] http://www.cs.utexas.edu/users/ml/nldata.html
[10] http://www.ifi.uzh.ch/ddis/research/semweb/talking-to-the-semantic-web/owl-test-data/

5.3 Test Questions

Two different sets of test questions will be used, one per evaluation phase.

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 of the instances that are part 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?'*, *'What **state** bordering **Nevada** has the **largest population**?'* and *'How many **states** border **Colorado** and border **New Mexico**?'*.

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?'*.

6. PARTICIPANT REQUIREMENTS

Participation in the evaluation campaign is open to any search tool developer and no restriction is placed on the type of interface or underlying technologies. However, there are a number of logistical aspects which must be adhered to.

The evaluation of a tool's usability by human users in the user-in-the-loop phase is an essential part of the evaluation campaign. Furthermore, as mentioned in Sec. 4.2, the responsibility for conducting these experiments rests with the participant. Therefore, all campaign participants must be aware that they will have to provide appropriate facilities and equipment for the experiment to be conducted and recruit suitable subjects. Hence, any costs must be borne by the participants themselves. Detailed instructions on how to prepare and conduct these experiments (including deadlines for completion) will be provided along with any necessary software.

In order for a tool to be automatically evaluated, the tool provider must produce a tool 'wrapper' which implements a number of functions as defined in Sec. 6.1. This allows the evaluation platform to automatically issue query requests and gather the result sets, for instance. Furthermore, exposing this functionality also allows the user-in-the-loop experiment software to gather various forms of data that will be used for analysis.

6.1 API

The core functionality can be split into three different areas: functions required in both phases, functions required only for the user-in-the-loop phase and functions required just for the automated phase. More details of the API can be found on the SEALS Portal.

6.1.1 Common To Both Phases

Load a particular ontology and triple set. This method loads a given ontology and triple set, which are being provided as a bundle. The success or failure of the tool's attempt to load the ontology will inform the interoperability criterion.

Determine result type. This method is used to determine if the tool manages (and hence returns via the API) its results as a ranked list. If not, it is assumed that the results are represented as a set.

Results ready?. This method is used to determine if the results of a query are yet available or not. The method is used (in combination with a timer) to determine how long a tool takes to answer a query. The execution time (speed) will be used among other metrics to measure the tool's performance.

Get results. This method obtains the result set currently held by the tool. It is left to the tool provider to decide how many results to return for the evaluation. Note, since we are dealing with ontologically based search tools, we expect only a list of URIs (as opposed to results containing faceted information, etc) as the result type.

6.1.2 Automated Phase

Execute query. This method executes the given query. The content of the query is determined by the tool provider and will be tailored to an individual search tool (i.e., a string representation of the internal query representation). The success or failure of the tool's attempt to execute the query will inform the interoperability and general performance criteria.

6.1.3 User-in-the-Loop Phase

Determine user query input has been completed. This method determines whether or not the user has finished inputting the query into the tool. In other words, it would return false while the user is still typing, clicking or thinking and would return true once the user has hit the 'submit' button (or equivalent).

Get user query. This method extracts a String representation of the query entered by the user. This will be called once the user input has been completed. The returned string will contain the string representation of the user's query input. If the tool uses a Natural Language interface, this method would simply return the text entered by the user.

Get internal query. This method extracts a String representation of the internal query. This will be called once the

user input has been completed and the search tool has transformed the input into its own representation. This string should be in a form such that it could be passed to the execute query function and obtain the same results.

7. DISSEMINATION

The evaluation results will be made available via the SEALS portal; it will allow browsing and comparison of the different tools' results. In particular, it will contain functionalities to browse the usability, accuracy and robustness results of all the tools evaluated with respect to the interpretations described in Secs. 4.4.1 and 4.4.2.

The outcomes of these analyses will be published in three reports (further outlined below), each targeting a specific aspect of the evaluation and, indeed, a different audience. It ought to be emphasised that all subject data will be anonymised and/or aggregated and no individual subject will be identifiable from the data or analyses presented in any public reports.[11]

Performance Report. This focusses on the low-level performance of a particular tool: the quality of its retrieved results and associated statistics.

Performance will cover aspects such as the CPU load and the memory usage of the tool; the results of the scalability experiments will also indicate the degree of robustness of the tool. Additionally, it will contain the search performance values relative to the ontology size and the ratio of accepted to rejected queries. Furthermore, the related statistical analyses of these results will be presented. Much of the raw data for this report will be gathered during the automated phase of the evaluation.

Usability Report. This report is concerned with the user's impression of a particular tool and the performance of the group of test subjects when using the tool to obtain specific information. The raw data for this report will be gathered during the user-in-the-loop phase of the evaluation.

The first part will integrate all data regarding the users' interaction with the system in the usability study and their answers to the several questionnaires. It will collect information about the number of clicks a user made during the execution of each task. This report will integrate all data regarding the user's interaction with the tool in the user-in-the-loop phase and the anonymised answers to each questionnaire. The subjects are given a unique (and anonymised) ID in order to identify the information per subject. The report will also indicate the number of attempts a user required per question and the timing results per question and the overall experiment. Further, the report will indicate, whether an answer was found for each question.

Finally, it will compile the various results of the possible correlations between the users' specific demographics and the metrics acquired in order to evaluate, whether there exist influencing factors on the results. It will demonstrate possible correlations between the user demographics and the measures as well as potential correlations between the user demographics and the other interpretations.

Comparison Report. Unlike the reports described above, the comparison report will present the comparative performance of all the tools evaluated. This targets both tool providers and tool adopters allowing them to compare a collection of tools according to a number of core factors described in the other reports. This not only simplifies the comparison of the tool's performance with others of the same kind for tool providers but also allows tool adopters to compare tools based upon specific criteria.

8. CONCLUSIONS

This paper has presented a methodology which can be applied to the evaluation of any semantic search tool regardless of its user interface. An evaluation following this methodology will be conducted using the infrastructure provided by the SEALS project – an initiative aimed at providing a worldwide, open and sustainable evaluation infrastructure which will form a standardised means of benchmarking any Semantic Web technology.

The evaluation framework allows for automated benchmarking of the tool based upon a number of criteria; however, in order to draw useful conclusions regarding the usability of a tool, it is essential that real-world users are also involved in the process. For the purpose of evaluating semantic search tools, therefore, we adopt a two phase approach: an automated phase and a user-in-the-loop phase. The user-in-the-loop phase comprises a series of experiments incorporating human subjects who are given a particular tool and test data in order to solve a number of tasks, i.e., posing questions in the tool's interface or query language.

We have reviewed a number of previous evaluation initiatives which have informed the methodology presented here. We have defined the criteria by which the tools will be assessed and identified the particular metrics and interpretations that will be employed in each evaluation phase.

A set of reference benchmark tests is described for assessing the strengths and weaknesses of the available tools and for comparing them with each other. As such, these tests focus on the performance of fundamental aspects of the tool in a strictly controlled environment / scenario rather than their ability to solve open-ended, real-life problems. These fundamental aspects are the formal evaluation criterion by which we will benchmark each tool:

- Query expressiveness
- Usability (effectiveness, efficiency, satisfaction)
- Scalability
- Quality of documentation
- Performance

In addition, we have presented the data sets that were selected based upon the evaluation requirements. Since the evaluation consists of two distinct phases addressing different criteria, they will operate on different data sets to ensure optimal applicability to each evaluation task. The user-in-the-loop phase will use the Mooney dataset since this contains concepts easily understood by casual users as well as having been used in previous semantic search tool evaluations (and hence accepted for this purpose within the community). Furthermore, it features a precompiled set of

[11] For the 2010 campaign we plan to invite all participants to take part in a workshop organised by the SEALS consortium to be held at the International Semantic Web Conference (ISWC) 2010 conference, giving them the opportunity to present and discuss their results with other evaluated tools. We plan to provide similar venues in the following years.

questions for which the groundtruth is known. Since the core criteria of the automated phase addresses scalability and performance issues, datasets of differing sizes (1k, 10k, 100k, 1M, 10M triples) were required. Unfortunately, the size of the Mooney dataset precludes it from this phase. As an alternative, EvoOnt was chosen due to its flexibility.

Participation in the evaluation is open to any tool developer and we encourage potential participants to find out more information about the semantic search evaluation campaign and the SEALS Evaluation Campaigns in general by visiting the SEALS website[12]. More detailed information regarding the timescales involved and the materials which will be provided can be found there as well as mechanisms for registering interest in the campaign and formally enrolling.

To conclude, we hope that the methodology presented here will spur the adoption of principled evaluations in the Semantic Search tool community and thereby foster innovation.

9. ACKNOWLEDGMENTS

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

10. REFERENCES

[1] A. Bernstein and E. Kaufmann. Gino - a guided input natural language ontology editor. In *5th International Semantic Web Conference (ISWC 2006)*, pages 144–157. Springer, November 2006.

[2] A. Bernstein, E. Kaufmann, A. Göhring, and C. Kiefer. Querying ontologies: A controlled english interface for end-users. In *4th International Semantic Web Conference (ISWC 2005)*, pages 112–126, November 2005.

[3] A. Bernstein, E. Kaufmann, A. Göhring, and C. Kiefer. Querying Ontologies: A Controlled English Interface for End-users. In *4th International Semantic Web Conference (ISWC 2005)*, pages 112–126, November 2005.

[4] A. Bernstein, E. Kaufmann, and C. Kaiser. Querying the Semantic Web with Ginseng: A Guided Input Natural Language Search Engine. In *15th Workshop on Information Technology and Systems (WITS 2005)*, pages 45–50, December 2005.

[5] N. Bettenburg, S. Just, A. Schröter, C. Weiss, R. Premraj, and T. Zimmermann. Quality of Bug Reports in Eclipse. In *Proceedings of the 2007 OOPSLA Workshop on Eclipse Technology eXchange*, pages 21–25, New York, NY, USA, October 2007. ACM.

[6] 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.

[7] 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.

[8] M. d'Aquin, C. Baldassarre, L. Gridinoc, S. Angeletou, M. Sabou, and E. Motta. Characterizing knowledge on the semantic web with watson. In *EON*, pages 1–10, 2007.

[9] L. Ding, T. Finin, A. Joshi, R. Pan, R. S. Cost, Y. Peng, P. Reddivari, V. C. Doshi, and J. Sachs. Swoogle: A Search and Metadata Engine for the Semantic Web. In *Proceedings of the Thirteenth ACM Conference on Information and Knowledge Management*. ACM Press, November 2004.

[10] J. A. Hanley and B. J. McNeil. The meaning and use of the area under a receiver operating characteristic (roc) curve. *Radiology*, 143:29–36, 1982.

[11] 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.

[12] 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.

[13] E. Kaufmann, A. Bernstein, and L. Fischer. NLP-Reduce: A "naïve" but Domain-independent Natural Language Interface for Querying Ontologies. In *4th European Semantic Web Conference (ESWC 2007)*, pages 1–2, November 2007.

[14] E. Kaufmann, A. Bernstein, and R. Zumstein. Querix: A Natural Language Interface to Query Ontologies Based on Clarification Dialogs. In *5th International Semantic Web Conference (ISWC 2006)*, pages 980–981. Springer, November 2006.

[15] C. Kiefer, A. Bernstein, and J. Tappolet. Mining Software Repositories with iSPARQL and a Software Evolution Ontology. In *Proceedings of the 2007 International Workshop on Mining Software Repositories (MSR '07)*. IEEE Computer Society, March 2007. to appear.

[16] Y. Lei, V. Uren, and E. Motta. Semsearch: A search engine for the semantic web. In *Proc. 5th International Conference on Knowledge Engineering and Knowledge Management Managing Knowledge in a World of Networks, Lect. Notes in Comp. Sci., Springer, Podebrady, Czech Republic*, pages 238–245, 2006.

[17] 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.

[18] 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.

[12] http://www.seals-project.eu/