How Useful are Natural Language Interfaces to the Semantic Web for Casual End-users?

Esther Kaufmann and Abraham Bernstein

Department of Informatics, University of Zurich, Switzerland
{kaufmann,bernstein}@ifi.uzh.ch

Abstract. Natural language interfaces offer end-users a familiar and convenient option for querying ontology-based knowledge bases. Several studies have shown that they can achieve high retrieval performance as well as domain independence. This paper focuses on usability and investigates if NLIs are useful from an end-user's point of view. To that end, we introduce four interfaces each allowing a different query language and present a usability study benchmarking these interfaces. The results of the study reveal a clear preference for full sentences as query language and confirm that NLIs are useful for querying Semantic Web data.

1 Introduction

The need to make the contents of the Semantic Web accessible to end-users becomes increasingly pressing as the amount of information stored in ontology-based knowledge bases steadily increases. Natural language interfaces (NLIs) provide a means of querying access to casual end-users without having them to learn RDF, OWL, SPARQL, or other logic-based languages. While NLIs hide the formality of ontologies and query languages from end-users by offering them a familiar and intuitive way of query formulation, the realization of NLIs involves various problems:

- Due to linguistic variability and ambiguities, for which natural languages (NLs) are infamous, the development of accurate NLIs is *a very complex and time-consuming task* that requires extraordinary design and implementation efforts. Nevertheless, by restricting and controlling the query language such that the end-user has to follow it or engage the user in query formulation dialogues, we can eliminate linguistic variability [6, 22]. Moreover, the semantics that is contained in ontologies can provide the context needed to overcome ambiguities.
- Good NLIs are often domain- or application-tailored, which makes them *hardly adaptable and portable*. However, if we extract the necessary information to analyse a user's NL query from a knowledge base, NLIs become domain-independent or, at least, easily adaptable to new domains [3, 9].
- The retrieval performance (in terms of precision and recall) of a NLI is directly linked to the portability problem. *The more a system is tailored to a domain, the better its retrieval performance is*. The goal, however, is to build portable NLI without sacrificing retrieval quality because end-users would not accept unreliable and inaccurate interfaces.

– Repeatedly held discussions in the NLI literature raise the *issue of the usefulness of NILs*. Even if we design a well-performing and domain-independent NLI, it remains unclear if it is approved and adopted by end-users. In the time of Google and graphical user interfaces, where people are used to formulating their information needs with keywords and then browse through dozens of answers to find the appropriate one or to clicking through menus and graphically displayed functions, full-fledged NLIs may be redundant.

Though we have identified four problem dimensions regarding NLIs—and there may be other—we think that NLIs are a promising option for casual end-users to interact with logic-based knowledge bases. Several projects have shown that NLIs can perform well in retrieval tasks [12, 20, 24] and be portable as well as domain-independent [9, 18, 26] without being unnecessarily complex. This paper now attempts to shed some light on the problem dimension of usability and usefulness of NLIs (i.e., the last of the four issues raised above). To that end, we have implemented four interfaces, which are portable, domain-independent, and exhibit good performance, to conduct a comprehensive usability study. The four interfaces are simple in design, avoid complex configurations, and extract the knowledge needed to analyse input queries from OWL knowledge bases.

Each interface supports a different query language with a different degree of restriction and formality. We benchmarked the four systems against each other in a usability study with 48 subjects providing us with an answer to the question about the usefulness of NLIs from an end-user's point of view. Consequently, our contribution is that *we investigate if NLIs to Semantic Web data are in fact useful for and approved by casual end-users.* Note that we refer to casual end-users as defined in [4].

The remainder of the paper is structured as follows. First, we introduce each of the four interfaces and explain their major characteristics. We, then, describe the usability study in section 3, in which the four systems are benchmarked against each other, and discuss the results, which leads to the discussion of some limitations of our approach as well as future work in section 4. The paper closes with a section on related work and conclusions.

2 Four Different Query Interfaces to the Semantic Web

Given our premise that NLIs are only useful for casual end-users if they are actually approved and, therefore, used by them, we conducted a usability study with four query interfaces implemented for that purpose: *Ginseng*, *NLP-Reduce*, *Querix*, and *Semantic Crystal*. Each interface requires a different query language regarding its freedom, naturalness, and formality: ranging from keywords to complete English sentences, from menu-based options to a graphically displayed query language. In the following, we describe each of the four systems beginning with the interface that has the least restrictive and most natural query language, then continuing with the systems that feature more restricted query languages, and closing with the system requiring a formal, graphical query language.

2.1 NLP-Reduce

NLP-Reduce is a "naïve" and completely domain-independent NLI for querying Semantic Web knowledge bases [16]. It is called *naïve* because the approach is simple and processes NL queries as bag of words only employing a reduced set of NL processing techniques, such as stemming and synonym expansion (hence its name *NLP-Reduce*). The interface allows users to enter keywords (e.g., "Chinese restaurant San Francisco"), sentence fragments (e.g., "Chinese restaurants that are in San Francisco"), or full English sentences (e.g.,"Which Chinese restaurants are in San Francisco?").

A query is first reduced by removing stopwords as well as punctuation marks and stemming the rest of the words. The system then tries to identify triple structures in the rest of the query words and match them to the synonym-enhanced triple store that is generated from an OWL knowledge base when loaded into NLP-Reduce. The identified triples are joined and translated into SPARQL statements. To execute the SPARQL query, NLP-Reduce uses Jena[1] and the Pellet Reasoner.[2] After executing the query, the results (including the URIs) and some execution statistics are displayed to the user (see Fig. 1).[3]

When generating the triple store from a knowledge base, NLP-Reduce also obtaines synonyms from WordNet providing the users with a larger vocabulary that can be deployed when querying. This leads to better usability and eases the interface's limitation of being dependent on the quality and choice of the vocabulary used in knowledge bases. The weakness, however, is also the interface's major strength, as it does not need any adaption for new knowledge bases and is completely portable. From an end-user's point of view, the major advantage of the system is that it is robust to ungrammatical and deficient input.

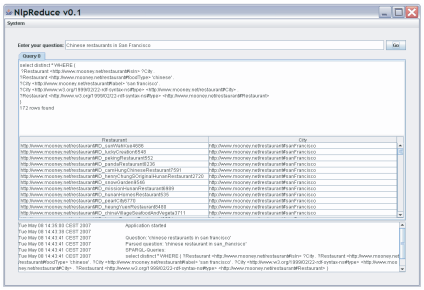

Fig. 1. The NLP-Reduce user interface

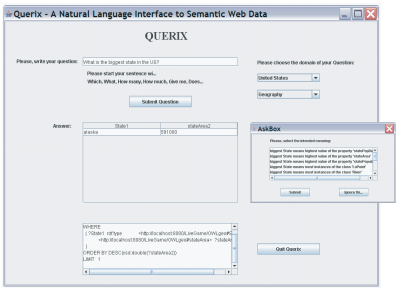

Fig. 2. The Querix user interface

[1] http://jena.sourceforge.net/
[2] http://pellet.owldl.com/
[3] Larger images of each interface can be found at:
http://www.ifi.uzh.ch/ddis/research/semweb/talking-to-the-semantic-web/

2.2 Querix

Querix is a domain-independent NLI that requires full English questions as query language [17]. Compared to a logic-based NLI, Querix does not try to resolve NL ambiguities, but asks the user for clarification in a dialog window if an ambiguity occurs in the input query. The user acts the role of the druid Getafix (hence the name *Querix*) who is consulted by Asterix, Obelix and the other villagers whenever anything strange occurs. A strange event within Querix is an ambiguity. The person composing a query benefits from the clarification dialog by better retrieval results.

The system uses a parser to analyse the input query. From the parser's syntax tree, a query skeleton is extracted, in which triple patterns are identified. Based on pattern matching algorithms that rely on the relationships that exist between the elements in a knowledge base, the triple patterns are then matched to the resources in the knowledge base. The matching and joining of the triples is controlled by domain and range information. From the joined triples, a SPARQL query is generated that can be executed by Jena. Using WordNet, synoyms of the words in the query and the labels in the knowledge base are included, providing an enhanced query language vocabulary and a better matching.

If Querix encounters an ambiguity in a query, i.e., several semantically different SPARQL queries could be generated for a single NL query, the clarification dialog of the interface pops up showing the different meanings for the ambiguous element in a menu (Fig. 2). The user can now choose the intended meaning, and the interface executes the corresponding SPARQL query. Consider, for example, the query "What is the biggest state in the US?", in which the word "biggest" can refer to the properties `statePopulation`, `statePopulationDensity`, and `stateArea` of a knowledge base containing geographical information. If the user selects `statePopulation`, the answer to the query is "California;" if `stateArea` is selected, the answer Querix returns is different, namely "Alaska."

2.3 Ginseng

Ginseng - a *guided input natural language search engine* allows users to query OWL knowledge bases using a controlled input language akin to English [5, 7]. Basing on a grammar, the system's incremental parser offers the possible completions of a user's entry by presenting the user with choice pop-up boxes (as shown in Fig. 3). These pop-up menus offer suggestions on how to complete a current word or what the next word might be. The possible choices get reduced as the user continues typing.

Entries that are not in the pop-up list are ungrammatical and not accepted by the system. In this way, Ginseng guides the user through the set of possible questions preventing those unacceptable by the grammar. Once a query is completed, Ginseng translates the entry to SPARQL statements, executes them against the ontology model using Jena, and displays the SPARQL query as well as the answer to the user.

When starting Ginseng, all knowledge bases in a predefined search path are loaded and the grammar compiler generates a dynamic grammar rule for every

class, property, and instance. These dynamic rules enable the display of the labels used in the ontology in the pop-up boxes. While the static grammar rules provide the basic sentence structures for questions, the dynamic rules allow that certain non-terminal symbols of the static rules can be "filled" with terminal symbols (i.e., the labels) that are extracted from the ontology model. As such, Ginseng is domain-independent and highly portable.

Ginseng also allows that synonyms of the labels used in the ontology model can be included by annotating the ontology with additional tags from the `ginseng` namespace. For each synonym, Ginseng also generates a dynamic grammar rule. While such annotations are not necessary for Ginseng to run correctly, they extend its vocabulary and increase its usability. Additionally, they reduce the limitation that the approach depends on the choice of vocabulary, when an ontology was built. In fact, the more meaningful the labels of an ontology are chosen, the wider and more useful the vocabulary provided by Ginseng is. More information on Ginseng and its ontology editor extension GINO can be found in [5, 7].

2.4 Semantic Crystal

Our last interface has the most formal and most restrictive query language of the four systems. In order to compare the other NLIs with a formal approach, but keeping in mind that casual end-users are better at understanding graphical query interfaces than formal query languages [23], we implemented *Semantic Crystal*. The name is an homage to Spoerri's *InfoCrystal*, a graphically-based query tool for Boolean and vector space information retrieval [23].

Semantic Crystal is new and not yet documented in the literature. The domain-independent interface can be used for querying any OWL-based knowledge base that is locally stored or on the web. It displays the ontology model to the user as shown on the left side of Fig. 4, which is a very advantageous feature for the end-user. A query is composed by clicking on elements in the graph and selecting elements from menus. Once an element has been selected, the interface presents it on the query graph dashboard on the upper right side of the user interface. The user can then continue assembling the query either on the dashboard or in the graph representation of the ontology model.

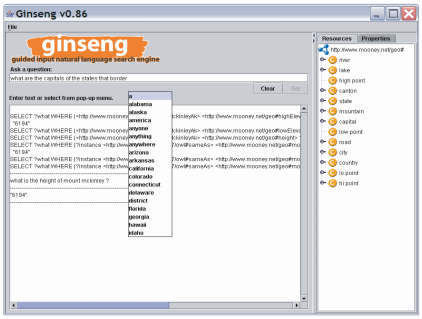

Fig. 3. The Ginseng user interface

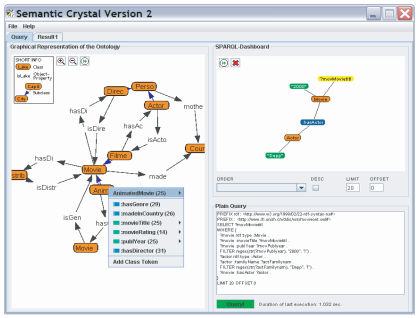

Fig. 4. The Semantic Crystal user interface

When clicking on a class (represented by the orange elements in the graph), the interface lists all properties of the class enabling the user to select only valid ones. The interface incrementally generates textual SPARQL query statements for the current state of the graphically constructed query; the SPARQL statements are exhibited on the bottom of the right side of the user interface. In the case of datatype properties (the green properties in the menu list), a user can additionally specify whether the property's value should be used as restriction or as output. If the output is specified, the query can be executed and the result is shown to the user in a new tab. Jena is again applied for query execution. On the dashboard in Fig. 4, we see a complete graphical representation of the query: "Give me the titles of the movies that have an actor with the family name 'Depp' and that were distributed in the year 2000."

3 The Usability Study

The goal of the usability study was to investigate how useful NLIs were to find data in Semantic Web knowledge bases in comparison with each other and a formal query language. Casual end-users should test and assess the usability of each of the four systems and, in particular, their query languages. As such, we let casual end-users perform the same retrieval tasks with each tool and find out which query language they liked best, which query language they liked least, and why. Furthermore, we examined the time they spent to perform the tasks and how successful they were with each system. To recall the range of query languages and their features provided by our interfaces, we summarize them here:

- NLP-Reduce: keywords, sentence fragments, and full sentences
- Querix: full sentences
- Ginseng: predetermined and menu-based words/sentences
- Semantic Crystal: graphically displayed, clickable, formal query language

3.1 Experimental Setup

To benchmark the four interfaces in a controlled experiment, we promoted the usability study on the websites of our department and university. We, additionally, promoted the study by billboard advertisements, which we distributed in Zurich. We ended up with 48 subjects almost evenly distributed over a wide range of backgrounds and professions: bankers, biologists, computer scientists, economists, game programmers, housewives, journalists, language teachers, mechanical engineers, musicians, pedagogues, psychologists, secretaries, sociologists, veterinarians, video artists, unemployed persons to name some of them (in alphabetical order). There was a normal distribution of age ranging from 19 to 52 years with a mean of 27.6 years. With 48 users we were able to cover each possible order of the four systems not just once but twice, a fact that increases the overall statistical significance of the results (see below).

For each interface the users were asked to perform the same tasks: They had to reformulate four questions presented to them as sentence fragments into the

respective query language required by the four systems and enter the questions into the interfaces. The four questions were principally the same for each system, but we slightly changed them to make the overall experiment more interesting for the users. For example, one question was "area of Alaska?" given for NLP-Reduce and "area of Georgia?" for Querix etc. The four question templates were:

- area of Alaska?
- number of lakes in Florida?
- states that have city named Springfield?
- rivers run through state that has largest city in US?

In principle, each interface is able to answer all four queries. Each system does, however, "stumble" across one of the queries such that, for example, more than one query is needed to retrieve the correct result. For every user, we changed the order in which the interfaces were presented as well as the order of the queries for each system to prevent any learning effects from influencing the results.

After completing the questions with each interface, users were asked to answer the *SUS* questionnaire. SUS is a standardized usability test [8] containing ten standardized questions (e.g., "I think that the interface was easy to use."), each answered on a 5-point Likert scale establishing a person's impression regarding a user interface. The test covers a variety of usability aspects, such as the need for support, training, and complexity. The result of the questionnaire is a value between 1 and 100, where 1 signifies that a user found a system absolutely useless and 100 that a user found a system optimally useful.

After testing and judging all interfaces, users were explicitly asked to fill in a comparison questionnaire in which they were asked which NLI they liked best and which one they liked least; they were asked the analogous questions regarding the query languages. We also asked them about the motivations for their choices. At the end of the overall experiment, people were requested to answer a number of demographic questions such as age, gender, profession, knowledge of informatics, knowledge of linguistics, knowledge of formal query languages, and knowledge of English.

To provide an introduction to the query languages of the interfaces, users were given 1-page instructions for each system. Hence, the procedure of the experiment for each user was the following: (1) read some introductory notes on the overall experiment, (2) read instructions on the query language of the first interface, (3) reformulate, enter, and execute four queries with the first interface, (4) fill in the SUS questionnaire for the first interface, (5) proceed by repeating steps 2 to 4 with the second, third, and fourth interface, (5) fill in the comparison questionnaire, (6) and finally provide answers to the demographic questions. The overall experiment took about 45 to 60 minutes. Using the Morae Software,[4] we were able to remotely record the desktop of the users as well as log and time each of their key entries and mouse clicks.

The experiment was based on the *Mooney Natural Language Learning Data* [24]. Its geography database consists of a knowledge base that contains geographical information about the US and their logical representations. We chose

[4] http://www.techsmith.com/morae.asp

the data set because it covers a domain that can easily be understood by casual users and does not demand expert knowledge. To make the knowledge base accessible to our interfaces, we translated it to OWL and designed a simple class structure as meta model.

3.2 Results of the Experiment

The data we collected in the experiment was analysed quantitatively as well as qualitatively. For the quantitative analysis, we used *ANOVA* and *Mixed Linear Regression Models* as available in the R-Software[5] and its lme4-package.[6] The first part of the results is summarized in Table 1.

Most strikingly, our results are more than highly significant, which is due to the high number of users and the double coverage of every possible interface as well as query order. The first column shows that users were significantly fastest when entering the four queries with NLP-Reduce (p = 1.56e-26). This outcome is obvious as the query language of NLP-Reduce imposes least constraints on the user and allows entering the queries with least words. Users spent most time when working with Semantic Crystal demonstrating that the intellectual burden of composing semantically and syntactically appropriate formal queries lies exclusively with the user, whereas the other three systems carry the burden to some extent. The average time that was spent per query nicely mirrors the increasing degree of formality and restrictiveness of the interfaces' query languages.

We can see in column 3 that it took users on average 7.02 queries to find answers to the four questions given in the experiment with Semantic Crystal and 11.06 query trials with Ginseng. NLP-Reduce and Querix lie in between and close to each other. The high number of query trials in Ginseng is a result of its query language's restrictiveness causing users to repeatedly reformulate and execute their queries in a kind of backtracking behavior. The log files revealed that the lowest number of query trials in Semantic Crystal emerged from users giving up and not willing to keep trying until an appropriate query was composed.

The average success and failure rates indicate how many of the four queries retrieved a satisfying answer from the users' perspective (i.e., the user thought

[5] http://www.r-project.org/
[6] http://stat.ethz.ch/CRAN/

	avg time for all 4 queries	avg time per query	avg number of queries	avg success rate	avg failure rate
NLP-Reduce	2 min 39 sec	23.54 sec	7.94	69.27 %	30.73 %
Querix	4 min 11 sec	29.31 sec	7.75	77.08 %	22.92 %
Ginseng	6 min 06 sec	34.82 sec	11.06	63.54 %	36.46 %
Semantic Crystal	9 min 43 sec	89.53 sec	7.02	54.86 %	45.14 %
p-value (single factor) ANOVA with 4 levels)	1.56e-26	4.91e-40	3.92e-06	1.06e-05	2.54e-05

Table 1. Average time, number of queries, and success/failure rate results.

that she/he had found the correct answer). Though Semantic Crystal in fact provides more precise answers than its competitors, the success rate of only 54.86% is due to inappropriate and invalid query formulations. The best success rate achieved by Querix from the users' point of view is due to Querix's answer display. For example, if a user enters a query "How many rivers run through Colorado?", the answer of Querix is: "There are 10.", while the other three interfaces show a list with the names of ten rivers and the number of results found. Some users specifically pointed out in the questionnaires that they trusted the answers of Querix more because the NL answer created the impression that the system "understood" the query.

Using the regression model, we found that the order of the four queries and the knowledge of informatics, linguistics, formal query languages, and English did not significantly affect the time. While there was no correlation between the variable *gender* and the average time spent per query either, the variable *age* influenced the average time: With every year a user's age grows, the average time to reformulate a query increases by 3.30 seconds ($p = 0.010$).

Table 2 contains the results of the SUS and the comparison questionnaires. Querix achieved the highest average SUS score of 75.73 and significantly outperformed the other three interfaces ($p = 7.36e\text{-}17$). The graphical query interface Semantic Crystal did not get much appreciation, which is reflected in the average SUS score of 36.09. NLP-Reduce and Ginseng achieved similar SUS scores somewhere in the middle of the other two NLIs; their scores do not significantly differ from each other (paired, one-tailed t-Test: $p = 0.356$). It is no surprise that 66.67% of the users liked the Querix interface best and only 2.08% liked it least, even if this result is not significant (columns 2 and 3 in Table 2). Querix obtained almost the same feedback for its query language (QL), this time reaching statistical significance (columns 4 and 5). Even though 60.42% of the users disliked Semantic Crystal as query interface when comparing it to the other three NLIs, a surprising portion of 14.58% assessed Semantic Crystal as favorite interface. The graphically displayed knowledge base was found useful by five users. Only 12.50% liked NLP-Reduce best and 6.25% Ginseng. With respect to the query language, the results show the same ranking as the SUS scores except for the query language liked least. Here, the keywords provided by NLP-Reduce were more disliked (25.00%) than the restricted query language of Ginseng (12.50%).

	ave SUS score	interface liked best	interface liked least	QL liked best	QL liked least
NLP-Reduce	56.72	12.50 %	25.00 %	18.75 %	25.00 %
Querix	75.73	66.67 %	2.08 %	60.42 %	4.17 %
Ginseng	55.10	6.25 %	12.50 %	16.67 %	12.50 %
Semantic Crystal	36.09	14.58 %	60.42 %	4.17 %	58.33 %
p-value (single factor ANOVA with four levels)	7.36e-17	0.297	0.297	0.0075	0.0075

Table 2. Results of SUS and comparison questionnaires.

We can, therefore, hypothesize that the full freedom of keyword-based query languages is less suitable for casual end-users, since it does not support the user in the process of query formulation. The overall preference for Querix may further reflect this query language tradeoff between freedom that can produce confusion and restrictiveness that can enable guidance.

The regression analysis showed that with each second spent more with a system, the SUS score dropped by 0.06 ($p = 1.79e\text{-}09$), whereas the number of queries used, the success/failure rate, and the order of the queries did not influence the SUS ratings. The order in which the interfaces were presented to the user, however, made an impact: The system that was tested last always obtained a higher SUS score ($p = 0.0025$), i.e., an increase by 5.3. *Knowledge of informatics* was the only additional variable that also influenced the SUS ratings: The better the knowledge of informatics of a user was, the higher the SUS score turned out for each interface ($p = 0.0029$).

When categorizing and counting the comments that users gave in the comparison questionnaire, the most often-named comments for each interface were the following:

- NLP-Reduce: +easy, +no training necessary, –QL not apparent, –QL too relaxed
- Querix: +obvious and non-constraining QL, +full sentences possible, +asks for clarification, –full sentences too restrictive
- Ginseng: +easy, +supports the user, –QL too restrictive, –few sentence structure possibilities
- Semantic Crystal: +graphical display of data, –too complex, –too laborious, –much training necessary

3.3 Discussion of the Most Remarkable Results

The results of the usability study with 48 users clearly show that *Querix and its query language requiring full English questions was judged to be the most useful and best-liked query interface.* This finding contradicts another usability study investigating different query languages and showing that students generally preferred keyword-based search over full-questions search [21]. The users in that study declared that they would only accept full query sentences if the retrieval results were better. In contrast, our results exhibit a highly significant preference for full-sentence queries independent of the retrieval performance.

One of the most prominent qualitative results was that several users, who rated Querix as best interface, explicitly stated that *they appreciated the "freedom of the query language."* Nevertheless, full sentences are more restrictive than keywords and sentence fragments meaning that the query language of NLP-Reduce actually offers more freedom and less restriction than Querix. There may be two reasons for the comment: (1) With full-sentence questions, users can communicate their information need in a familiar and natural way without having to think of appropriate keywords in order to find what they are looking for. (2) People can express more semantics when they use full sentences and not just keywords. Using verbs and prepositions to link loosely listed nouns enables

semantic associations, which users may experience as more freedom in query formulation.

Though Semantic Crystal was assessed as difficult and laborious to use, some users pointed out *the advantage of graphically displayed knowledge bases and queries*. Consequently, we should consider interfaces to Semantic Web data that offer a combination of graphically displayed and NL query languages. A user could then choose between different querying possibilities. Furthermore, we might have to think of adequate *NL answer generation components* [2], which seems to increase a user's trust in a system and the overall user satisfaction.

4 Limitations and Future Work

We are well aware that our usability study does not provide a definitive answer to the discussion of the usefulness of NLIs. We deliberately omitted both a retrieval performance evaluation and a portability evaluation of our systems concentrating only on the dimension of usability. The former two evaluations have partially been done for some of the systems and their completion is part of our future work.

Concerning valid conclusions to be drawn from a usability study, we would still need a more comprehensive usability study with more users to cover more precisely distinguished degrees of query languages along a well-defined formality continuum. To prevent influences from variables that are not directly linked to the query languages, the NLIs should be the same except for the query languages. In our study the appearance of the interfaces was different.

We limited ourselves to four interfaces and four queries for several reasons. First, we wanted to cover each possible tool order; consider that a usability study with five different interfaces requires 120 users to cover each order of the interfaces. Second, we preferred to not overload the users in an exhaustive experiment risking to taint the results due to fatigue. Last, our users should not be students (like in most usability studies), but people representing a general public. Finding such users is a difficult, time-consuming, and also expensive endeavor, since we offered our users a small monetary reward for taking part.

We still believe that our usability study provides a substantial contribution to the discussion of how useful NLIs are for casual end-users. Motivated by the work of [27], we will, therefore, develop and implement a combined interface as described above and conduct further usability studies in the future.

5 Related Work

NLIs have been developed since the 70s, but oftentimes with moderate success [3, 25], which resulted in a decreasing interest in the topic in the 90s. The necessity for robust and applicable NLIs has become more acute in recent years as the amount of information has grown steadily and immensely. A number of well-performing NLIs to databases emerged [1, 10, 13, 14, 19, 20]. Considering the difficulties with full NL, it seems comprehensible that restricted NL or menu-guided interfaces have been proposed by some approaches [14, 22, 25]. The popularity of the Semantic Web created a number of NLIs that provide access to

ontology-based knowledge bases [9, 11, 12, 15, 18, 26]. Most of the evaluations of NLIs mainly focus on retrieval performance and/or the portability dimension. As our work concentrates on usability, we will only discuss three closely related projects that conducted a usability study: ORAKEL, Squirrel, and CHESt.

ORAKEL by Cimiano and colleagues [9] is a portable NLI to knowledge bases that is ontology-based in two ways. First, it uses an ontology in the inference process to answer users' queries. Second, the system employs an ontology in the process of adapting the system to a domain and a specific knowledge base. This adaptation is performed by domain experts and has been evaluated in a user study. It was shown that people without any NLI expertise could adapt ORAKEL by generating a domain-specific lexicon in an iterative process. The controlled study involved 26 users from both academic and industrial institutions. Results were reported in terms of recall and precision showing that the iterative methodology to lexicon customization was indeed successful. A second experiment was performed to determine the linguistic coverage of 454 questions asked by end-users. They report an excellent coverage of 93%, but did not investigate the usefulness from the end-users' point of view.

The Squirrel system presented by Duke et al. [11] is a search and browse interface to semantically annotated data. It allows combined search facilities consisting of keyword-based and semantic search in order to balance between the convenience for end-users and the power of semantic search. Users can enter free text terms, see immediate results, and follow with a refinement of their query by selecting from a set of matching entities that are associated with the result set and returned by the system on the basis of an ontology. Squirrel has been evaluated in three steps: (1) in a heuristic evaluation, in which usability experts judged the interface according to a list of usability heuristics, (2) in a walk-through evaluation, where users were asked to complete a number of tasks, while their actions were recorded, and (3) in a set of field tests giving users information seeking tasks and collecting feedback. Promising results obtained from 20 users are reported: Squirrel achieved an average perceived information quality of 4.47 on a 7-point scale. It was rated positively regarding its properties but skeptically in terms of performance and speed. Regrettably, the authors provide neither a detailed description of the evaluations nor explicit results.

The core of the work by Reichert and her colleagues [21] lies in a usability study, making it most closely related to our work. They investigate how students assess the possibility of querying a multimedia knowledge base by entering full questions instead of just keywords. For this purpose, two versions of the e-learning question-answering tool CHESt were implemented. The first version offers a keyword-based search; the second version allows a semantic search with full sentences as query input. They conducted three task-oriented experiment sessions with 18, 18, and 14 students and benchmarked the two versions of CHESt. The outcome of the three sessions is that the students generally preferred the keyword-based search to the full questions search (76% on average). This was found to be independent of the appropriateness of the results. The students reported that they would use the option of complete questions if this yielded in better results. Nonetheless, the authors conclude that the intellectual

task of thinking and formulating full-sentence queries must not necessarily be considered as a burden compared to entering loose keywords. We can confirm this conclusion from our usability study, which presents a wider choice of query languages, and draw even more detailed conclusions.

The approaches in the field of NLIs nicely show that such interfaces can successfully tackle the performance and transportability dimension. As such, they complement our findings, which focuses on the usability dimension. However, more work is needed regarding NLIs to Semantic Web data and further comprehensive usability studies to investigate the end-users's perspective.

6 Conclusions

This paper attempted to answer the question *if NLIs are actually useful for casual end-users.* While most studies concerning NLIs to structured data aim at high-quality retrieval performance and transportability, we focused on the usability dimension. Our usability study with 48 users and four interfaces featuring four different query languages showed that the full-sentence query option was significantly preferred to keywords, a menu-guided, and a graphical query language. NLIs offering an adequate query language can, therefore, be considered to be indeed useful for casual end-users. We believe that our study generally shows the potential of NLIs for end-user access to the Semantic Web, providing a chance to offer the Semantic Web's capabilities to the general public.

This work was partially supported by the Swiss National Science Foundation (200021-100149/1).

References

1. T. Andreasen. An approach to knowledge-based query evaluation. *Fuzzy Sets and Systems*, 140(1):75–91, 2003.
2. I. Androutsopoulos, S. Kallonis, and V. Karkaletsis. Exploiting owl ontologies in the multilingual generation of object. In *10th Europ. Workshop on Natural Language Generation*, pages 150–155, Aberdeen, UK, 2005.
3. I. Androutsopoulos, G. D. Ritchie, and P. Thanisch. Natural language interfaces to databases - an introduction. *Natural Language Engineering*, 1(1):29–81, 1995.
4. L. Battle. Preliminary inventory of users and tasks for the semantic web. In *3rd Intl. Semantic Web User Interaction Workshop*, Athens, GA, 2006.
5. A. Bernstein and E. Kaufmann. Gino - a guided input natural language ontology editor. In *5th ISWC*, pages 144–157, Athens, GA, 2006.
6. A. Bernstein, E. Kaufmann, A. Göhring, and C. Kiefer. Querying ontologies: A controlled english interface for end-users. In *4th ISWC*, pages 112–126, Galway, Ireland, 2005.
7. 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 Technologies and Systems*, pages 112–126, Las Vegas, NV, 2005.
8. J. Brooke. Sus - a "quick and dirty" usability scale. In P. Jordan *et al.*, editors, *Usability Evaluation in Industry.* Taylor & Francis, London, 1996.

9. P. Cimiano, P. Haase, J. Heizmann, and M. Mantel. Orakel: A portable natural language interface to knowledge bases. Technical report, Institute AIFB, University of Karlsruhe, 2007.
10. M. Dittenbach, D. Merkl, and H. Berger. A natural language query interface for tourism information. In *10th Intl. Conf. on Information Technologies in Tourism*, pages 152–162, Helsinki, Finland, 2003.
11. A. Duke, T. Glover, and J. Davies. Squirrel: An advanced semantic search and browse facility. In *4th ESWC*, pages 341–355, Innsbruck, A, 2007.
12. A. Frank, H.-U. Krieger, F. Xu, H. Uszkoreit, B. Crysmann, B. Jörg, and U. Schäfer. Question answering from structured knowledge sources. *Journal of Applied Logic, Special Issue on Questions and Answers: Theoretical and Applied Perspectives*, 5(1):20–48, 2007.
13. N. Guarino, C. Masolo, and G. Vetere. Ontoseek: Content-based access to the web. *IEEE Intelligent Systems*, 14(3):70–80, 1999.
14. C. Hallett, R. Power, and D. Scott. Intuitive querying of e-health data repositories. In *UK E-Science All-hands Meeting*, Nottingham, UK, 2005.
15. B. Katz, J. Lin, and D. Quan. Natural language annotations for the semantic web. In *Intl. Conf. on Ontologies, Databases, and Applications of Semantics*, Irvine, CA, 2002.
16. E. Kaufmann, A. Bernstein, and L. Fischer. Nlp-reduce: A "naïve" but domain-independent natural language interface for querying ontologies. In *4th ESWC*, Innsbruck, A, 2007.
17. E. Kaufmann, A. Bernstein, and R. Zumstein. Querix: A natural language interface to query ontologies based on clarification dialogs. In *5th ISWC*, pages 980–981, Athens, GA, 2006.
18. V. Lopez, E. Motta, and V. Uren. Poweraqua: Fishing the semantic web. In *3rd ESWC*, pages 393–410, Budva, Montenegro, 2006.
19. M. Minock. A phrasal approach to natural language interfaces over databases. Technical Report UMINF-05.09, University of Umea, 2005.
20. A.-M. Popescu, O. Etzioni, and H. Kautz. Towards a theory of natural language interfaces to databases. In *8th Intl. Conf. on Intelligent User Interfaces*, pages 149–157, Miami, FL, 2003.
21. M. Reichert, S. Linckels, C. Meinel, and T. Engel. Student's perception of a semantic search engine. In *IADIS Cognition and Exploratory Learning in Digital Age*, pages 139–147, Porto, Portugal, 2005.
22. R. Schwitter and M. Tilbrook. Let's talk in description logic via controlled natural language. In *Logic and Eng. of Natural Language Semantics*, Tokyo, Japan, 2006.
23. A. Spoerri. Infocrystal: A visual tool for information retrieval management. In *2nd Intl. Conf. on Information and Knowledge Management*, Washington, D.C., 1993.
24. L. R. Tang and R. J. Mooney. Using multiple clause constructors in inductive logic programming for semantic parsing. In *12th Europ. Conf. on Machine Learning*, pages 466–477, Freiburg, Germany, 2001.
25. C. W. Thompson, P. Pazandak, and H. R. Tennant. Talk to your semantic web. *IEEE Internet Computing*, 9(6):75–78, 2005.
26. C. Wang, M. Xiong, Q. Zhou, and Y. Yu. Panto - a portable natural language interface to ontologies. In *4th ESWC*, pages 473–487, Innsbruck, A, 2007.
27. T. D. Wang and B. Parsia. Cropcircles: Topology sensitive visualization of owl class hierarchies. In *5th ISWC*, pages 695–708, Athens, GA, 2006.