Evaluating the usability of natural language query languages and interfaces to semantic web knowledge bases

Kaufmann, E; Bernstein, A

Abstract: 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 familiar and convenient means of query access to Semantic Web data for casual end-users. While several studies have shown that NLIs can achieve high retrieval performance as well as domain independence, this paper focuses on usability and investigates if NLIs and natural language query languages are useful from an enduser’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 natural language query sentences with a limited set of sentence beginnings over keywords or formal query languages. NLIs to ontology-based knowledge bases can, therefore, be considered to be useful for casual or occasional end-users. As such, the overarching contribution is one step towards the theoretical vision of the Semantic Web becoming reality.

DOI: [https://doi.org/10.1016/j.websem.2010.06.001](https://doi.org/10.1016/j.websem.2010.06.001)
Evaluating the Usability of Natural Language Query Languages and Interfaces to Semantic Web Knowledge Bases

Kaufmann, E; Bernstein, A
Evaluating the Usability of Natural Language Query Languages and Interfaces to Semantic Web Knowledge Bases

Abstract

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 familiar and convenient means of query access to Semantic Web data for casual end-users. While several studies have shown that NLIs can achieve high retrieval performance as well as domain independence, this paper focuses on usability and investigates if NLIs and natural language query languages 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 natural language query sentences with a limited set of sentence beginnings over keywords or formal query languages. NLIs to ontology-based knowledge bases can, therefore, be considered to be useful for casual or occasional end-users. As such, the overarching contribution is one step towards the theoretical vision of the Semantic Web becoming reality.
Evaluating the Usability of Natural Language Query Languages and Interfaces to Semantic Web Knowledge Bases

Esther Kaufmann a,1 Abraham Bernstein a,∗

a Dynamic and Distributed Information Systems Group, Department of Informatics, University of Zurich, Binzmühlestr. 14,
8050 Zurich, Switzerland

Abstract

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 familiar and convenient means of query access to Semantic Web data for casual end-users. While several studies have shown that NLIs can achieve high retrieval performance as well as domain independence, this paper focuses on usability and investigates if NLIs and natural language query languages 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 natural language query sentences with a limited set of sentence beginnings over keywords or formal query languages. NLIs to ontology-based knowledge bases can, therefore, be considered to be useful for casual or occasional end-users. As such, the overarching contribution is one step towards the theoretical vision of the Semantic Web becoming reality.

Key words: Natural Language Interfaces, Query Languages, Usability Study

1. Introduction

The Semantic Web presents the vision of a distributed, dynamically growing knowledge base founded on formal logic. The formal framework facilitates precise and effective querying to manage information-seeking tasks. Casual end-users, however, are typically overwhelmed by formal logic. So how can we help users to query a Web of logic that they do not seem to understand?

An often proposed solution to address the gap between common users and formal, logic-based systems is the use of natural languages for knowledge specification and querying. A natural language interface (NLI) is a system that allows users to access information stored in some repository by formulating the request in natural language (e.g., English, German, French, etc.). Some NLIs allow the use of full natural language, while others restrict the input to a sublanguage by a domain or to a controlled/restricted natural language by grammar and/or lexicon constraints. NLIs access different information repositories: databases, knowledge bases, or ontologies and ontology-based knowledge bases. While NLIs conveniently hide the formality of ontologies and query languages from end-users by offering them a very familiar and intuitive way of query formulation, the realization of NLIs involves various problems as discussed in the following:

---

* Corresponding author. Tel: +41 44 635 4579
Email addresses: kaufmann@ifi.uzh.ch (Esther Kaufmann), bernstein@ifi.uzh.ch (Abraham Bernstein).

1 Partial support provided by the Swiss National Science Foundation award 200021-100149/1
First, due to linguistic variability\textsuperscript{2} and ambiguities,\textsuperscript{3} for which natural languages are infamous, the development of accurate NLIs is a highly complicated and time-consuming task that requires extraordinary design and implementation efforts. Natural language processing (NLP) generally requires computationally and conceptually intensive algorithms relying on large amounts of domain-dependent background knowledge, which is, to make things worse, costly to produce\textsuperscript{5}. Nevertheless, by restricting the query language such that the end-user has to follow it or engage the user in query formulation dialogues that are controlled by the system, we can significantly reduce linguistic variability\textsuperscript{7,13,63} and provide the context to overcome any remaining ambiguity. The PENG environment\textsuperscript{63}, e.g., relies on a controlled subset of English, where every grammatical construct has exactly one semantic meaning. Moreover, the semantics that is contained in ontologies can provide the context needed to overcome ambiguities. As an example: a question about “Java” in the context of a software evolution ontology is more likely to be about the programming language than the beverage or island.

Second: NLIs with good retrieval performance (i.e., they find all relevant but only relevant information) are often domain- or application-tailored, which makes them hard to adapt and port. Particularly, systems that allow full natural language input are in almost every case restricted to the domain of the queried data repository\textsuperscript{30}. Their adaptation to new data repositories can only be accomplished by lengthy manual reconfiguration\textsuperscript{3}. The systems that can perform complex semantic interpretation and inference tend to require large amounts of domain-specific knowledge and engineering-intensive algorithms making the systems hard to adapt to other domains and applications, if at all possible. Hence, they have a substantial adaptivity barrier.

One approach to address the adaptivity barrier is the use of a large data repository, which specifies the universe of discourse. Such a repository allows to extract the necessary information to analyze and process a user’s natural language query. As a consequence, NLIs can overcome the barrier and successfully become domain-independent or, at least, easily adaptable to new domains\textsuperscript{3,20}. It is exactly this approach that makes NLIs so attractive for the Semantic Web: the meta-data contained in the ontologies as well as the data in the ontologies themselves provides itself ideally as background data for the automatic adaption of NLIs as witnessed by a number of domain independent Semantic Web NLIs (e.g.,\textsuperscript{49,20}).

Note that we define portable based on Grosz et al.\textsuperscript{34}, where a system is said to be portable or domain-independent, if it does not require manual customization to be used in a new domain. In contrast, a system is not portable or domain-dependent, if it is tailored to one specific domain (e.g., geography) requiring extensive hand-crafted customization for new domains (e.g., moving from geography to chemistry).

Hence, the quality of the retrieval performance (in terms of precision and recall) of an NLI is usually 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 and, therefore, valuable NLIs without sacrificing retrieval quality because end-users would not accept unreliable and inaccurate interfaces.

Third, even if we can provide well-performing and portable NLIs, another problem arises from the users’ side. Typically, users do not know what capabilities a natural language system has, since most NLIs do not help their users in assembling queries, which sometimes leads to a “clean sheet of paper” effect, also known as writer’s block\textsuperscript{8,18,55,70}. Consequently, \textit{users should be guided or at least supported} when building queries\textsuperscript{4}. Otherwise, many of the questions will not be understood correctly by an NLI or might even be rejected because the questions exceed or fall short of the capability of the system, as the user does not know what can be asked. The mismatch between the users’ expectations and the capabilities of a natural language system is called the \textit{habitability problem}\textsuperscript{71}. Current NLP tools, while easier to learn than formal logic, still suffer from the habitability problem, as they only understand some subset of natural language, but sometimes suggest full understanding. Moreover, since users type in regular sentences, they are tempted to anthropomorphize and think the computer actually understands their questions. Natural language systems, however, still need carefully de-
developed query statements. Thus, for the successful use of an NLI, users need to know what is possible to ask [3,7] and what are good questions to ask [29].

These issues generally raise the question of the usefulness of NLIs, which is repeatedly reflected in the literature discussing whether NLIs are practical and appropriate compared to formal query languages—however without any conclusive answer [8,18,22,23,53,71,72]: Formal query languages have been found inaccessible by casual users, but offer a rich tool for composing complex queries by experts; systems applying natural language query languages are afflicted with the adaptivity barrier and the habitability problem.

1.1. Habitability Hypothesis

Though we have identified three problem dimensions regarding NLIs—and there may be others—as well as questioned the usefulness of NLIs in general, 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 [30,57,69] and be portable [20,49,73] without being unnecessarily complex, as such tackling the adaptivity barrier. Some studies also investigated the usefulness of natural language for different tasks with regard to end-users [20,26,39,50,61,25], therefore addressing the habitability problem. Their findings provide important insights, but do not provide a conclusive answer.

In order to find more granular answers to the question of the usefulness of NLIs, this paper proposes to break the dichotomy between full natural language approaches and formal, logic-based query approaches regarding them as ends of a Formality Continuum, where the freedom of full natural languages and the structuredness of formal query languages lie at the ends of the continuum (see Figure 1). Basing on structuration theory [32,56], which states that structure enables action by providing a guide, but can also constrain when the guide overly constricts expressibility, we argue that query interfaces should impose some structure on the user’s input to guide the entry, but not overly restrict the user with an excessively formalistic language, therefore, alienating the user.

In particular, we intend to bring full natural language approaches and formal, logic-based approaches closer to each other, since we hypothesize that the best solutions for the casual and occasional end-user (in contrast to expert users) will lie somewhere in the middle of the Formality Continuum, as this provides the best tradeoff between structuredness and freedom, as such tackling the habitability problem. We, therefore, call this hypothesis the Habitability Hypothesis. According to the hypothesis, end-users are currently caught between the imprecision of uninterpreted keyword systems and the rigour of formal query engines—neither of which really addresses their needs. Consequently, end-user friendly search systems must either let users express their information needs more naturally and analyze their queries more intelligently [18], or allow enhancements to help as well as control the user’s query entry to overcome the habitability problem and reduce the complexity of query interpretation.

Our hypothesis is supported by previous experience with controlled natural languages, which have shown that they are much easier to learn by casual end-users than formal languages like logic and are sufficient for structured querying knowledge bases [7,17,50,63,68,71].

To evaluate the hypothesis we needed to elucidate if the optimal trade-off between formality and naturalness indeed lies somewhere between the extremes of the Formality Continuum. And since the hypothesis elaborates on user interaction, we had to evaluate the usefulness of query languages with end-user experiments. Specifically, we performed the following two steps:

First, limiting our evaluation to natural language querying we developed a total of four domain-independent query interfaces for casual end-users that lie at different positions of the Formality Continuum: NLP-Reduce, Querix, Ginseng, and Semantic Crystal (see Figure 2). The first two interfaces allow users to pose questions in almost full, or slightly restricted English, respectively. The third interface offers query formulation in a controlled language akin to English. The last interface belongs to the formal approaches, as it exhibits a formal, but graphically displayed query language. Each interface allows to query Semantic Web knowledge bases. The four interfaces are simple in design, avoid complex configurations, and extract the knowledge.
needed to analyze user queries from OWL knowledge bases, while still offering well-performing and appropriate tools for composing queries to ontology-based data for casual and occasional end-users [42]. Second, we conducted a comprehensive usability study by benchmarking our four tools against each other in a controlled experiment. The goal was that casual end-users should test and assess the usability of each of the four systems and, in particular, their query languages. This provided us with sufficient evidence to determine the advantages and disadvantages of query interfaces at various points along the Formality Continuum. In turn, this lead to concrete, fine-grained answers to the question where on the Formality Continuum the best query interface solutions for the casual end-user lie. Last and most importantly, the evidence could be used to validate the Habitability Hypothesis and, in turn, shed some light on the problem dimension of usability and usefulness of NLIs.

1.2. Contributions

While moving along the development and evaluation of the Habitability Hypothesis, the major contribution of the paper is that we investigate if NLIs to Semantic Web data are in fact useful for and approved by casual end-users. We will not focus on a retrieval performance evaluation, as most other studies do, but conduct a thorough and comprehensive usability study with real-world people. To our knowledge this study is the first study benchmarking multiple (different) styles of user-interaction in the context of search on the Semantic Web. Consequently, the results gained in this study contribute an important corner-stone to the discussion of the usefulness and usability of different degrees of natural language query languages ranging from graphical, guided, and controlled, to full natural language query languages. The results may generalize beyond the Semantic Web to querying any data collection (such as data bases, other knowledge bases, or semi-structured knowledge-bases), where relationships between concepts sought are relevant.
The remainder of the paper is structured as follows. First, we introduce each of the four interfaces and explain the major characteristics of the systems, particularly their query languages (Section 2). We then describe the usability study in Section 3, in which the four systems are benchmarked against each other in a controlled experiment, present, and discuss the results of the study. This leads to the discussion of some limitations of our approach as well as future work. Section 4 reviews the most important related work. Finally, the paper closes with some 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 [44]. It is called naïve because the approach is simple and processes natural language queries as bag of words only employing a reduced set of natural language 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. Hence, NLP-reduce constructs the SPARQL statements by disregarding any grammatical relationships between the entered word phrases but exploiting possible relationships between the found terms in the knowledge base. To execute the SPARQL query, NLP-Reduce uses Jena including the Pellet Reasoner to infer implicit triple statements. After executing the query, the results (including the URIs) and some execution statistics are displayed to the user (see Fig. 3).

When generating the triple store from a knowledge base, NLP-Reduce also obtains synonyms from WordNet providing its users with a larger vocabulary that can be deployed when querying. This simplifies use 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 deficient, ungrammatical, and fragmentary input.

5 http://jena.sourceforge.net/
6 http://pellet.owldl.com/
7 http://wordnet.princeton.edu/
2.2. Querix

Querix is a domain-independent NLI that requires full English questions as query language [45]. Compared to a logic-based NLI, Querix does not try to resolve natural language 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 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 through better retrieval results.

Querix not only requires complete English query sentences, but also limits them to a given set of six sentence beginnings. As such, Querix mitigates the habitability problem and, furthermore, pursues the goal to avoid a tedious, complex, and domain-tailored system configuration. This can easily be attained by slightly limiting the natural language query language to a set of sentences that must begin with one of the following question or imperative sentence beginnings:

- Which ...
- What ...
- How many ...
- How much ...
- Give me ...
- Does ...

The system uses a parser to analyze 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, synonyms 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 natural language query, the clarification dialog of the interface appears showing the different meanings for the ambiguous element in a menu (Fig. 4). 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 [9,14]. 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. 5). These pop-up menus offer suggestions on how to complete a current word or what the next word might be. The number of possible choices decreases 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 facilitate its use. 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, 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 [9,14].

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 [66], 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 [66].

The domain-independent interface Semantic Crystal 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. 6. 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.

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. 6, 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

In order to test our Habitability Hypothesis we conducted a usability study on the basis of our four search systems. The hypothesis proposes that some structure should be imposed on casual end-users when formulating queries with a search interface in order to guide them, but not overly control or restrict and, hence, alienate them. Therefore, our assumption is that the best query language solutions will lie somewhere towards the middle of the Formality Continuum. Specifically, the goal of the usability study was to investigate how useful our three natural language query interfaces NLP-Reduce, Querix, and Ginseng were to find data in Semantic Web knowledge bases in comparison with each other and in comparison with the formal querying approach Semantic Crystal. We additionally aimed at gathering the data necessary to infer which degree of naturalness or formality and guidance with regard to a query language is most approved by casual end-users. As such, the study can contribute to the general discussion whether NLIs are useful from the end-users’ perspective.

After running several preliminary usability studies and gaining crucial experience with user experiments (the discussions of the preliminary evaluations can be found in Bernstein et al. 2005a [13], Bernstein et al. 2005b [14], Bernstein and Kaufmann [9], and Kaufmann and Bernstein [43]), we conducted a comprehensive usability study, in which we benchmarked our four systems against each other with 48 users. This section presents the accomplishment and the results of this usability study. 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 of the four tools to find out which query language they liked best, which query language they liked least, and why. Furthermore, we examined (1) the time they spent to perform the tasks, (2) how many queries they required to find the requested information, and (3) how successful they were in finding the appropriate answers with each system.

To recall the range of query languages and their features provided by our four interfaces, we list them here:
- NLP-Reduce: keywords, sentence fragments, and full sentences
- Querix: full sentences that must begin with “Which,” “What,” “How many,” “How much,” “Give me,” or “Does” and end with a question mark or full stop
- Ginseng: predetermined, fixed, controlled, and menu-based words/sentences akin to English
- Semantic Crystal: graphically displayed, clickable, formal query language

3.1. Experimental Setup and Methodology

The overall design of our benchmark evaluation followed the methods proposed by Nielsen [54] and Rauterberg [60]. We performed a deductive benchmark test, as the goal of our test situation was to evaluate different interface and query language alternatives. Our evaluation employed within-subjects testing in order to avoid biases and the distortion of the results. Before running the actual experiment, we conducted three pilot tests; two are suggested by the literature. This way, flaws of the test design could be identified and eliminated.

3.1.1. The Subjects

To benchmark the four interfaces with real-world casual end-users, we promoted the usability study on the Web sites of our department and the university. We, additionally, promoted the study by billboard advertisements, which we distributed randomly all over the city of 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, and unemployed persons to name most of them in alphabetical order. The participants were composed of 19 males and 29 females. There was a normal distribution of age ranging from 19 to 52 years with a mean of 27.6 years. As such, our subjects represented the general population of casual search interface end-users. With these 48 users, we were able to cover each possible order of the four systems (= 4!) not just once but twice, a fact that increases the overall statistical significance of the results (see Section 3.2 below).

A within-subject design is an experimental setup, where every subject is exposed to many “treatments” and the subject’s reaction to each of those treatments is compared statistically.
The subjects involved in our evaluation were given a reward, i.e., a monetary experimental fee, for their work to ensure a correct incentive-set, which should not be underestimated [21]. When testing with humans, it is, furthermore, important to take ethical aspects into account [54]. We had to make sure that the test users were well aware that the query interfaces were being tested and not the users, an important issue that can severely influence the test results [60]. We also had to ensure the subjects’ anonymity and a confidential data storing.

3.1.2. Tasks / Experimental Procedure

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 in order 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 or one of the words of the question templates cannot be recognized by the interface and has to be replaced with another word or omitted. We chose the query templates very carefully to provide a maximally fair competition for the four systems. 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 counterbalance any learning effects.

After completing the questions with each interface, the users were asked to answer the System Usability Scale (SUS) questionnaire. SUS is a standardized usability test by Brooke [16] containing ten standardized 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.). Each question is 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, as well as complexity, and has proven to be very useful when investigating the usability of interfaces [6]. 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. As usability is not an absolute criterion, the resulting SUS score can usually only be understood when comparing it with others, which was the case in our study comparing four systems and their query languages.

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.

At the beginning of each experimental run, the test user was given all information and instructions concerning the experiment on paper. The written form assured that every user was given exactly the same information and instructions. At first, the purpose of the test was explained to the test users. Then, the tasks were stated; the pilot tests granted for clarity of the task descriptions. We also made sure that each test user knew that he/she could interrupt or abort the experiment anytime.

To provide an introduction to the query languages of the interfaces, users were given 1-page instructions for the three NLIs and 2-page instructions for Semantic Crystal. Hence, the procedure of the experiment for each subject was the following:

(i) read some introductory notes on the overall experiment,
(ii) read instructions on the query language of the first interface,
(iii) reformulate, enter, and execute four queries with the first interface,
(iv) fill in the SUS questionnaire for the first interface,

Informatics is the European name for computer and computational sciences.
(v) proceed by repeating steps 2 to 4 with the second, third, and fourth interface,
(vi) fill in the comparison questionnaire about which system was liked best/least and why,
(vii) and finally provide answers to the demographic questions.

The overall experiment took about 45 to 60 minutes for each subject. Using the Morae Software,\(^\text{10}\) we were able to remotely record any desktop activity of the users as well as log and time each of their key entries and mouse clicks. An observer can annotate important incidents while an experiment is “on air.” All activities and annotations can be analyzed and visualized with the software’s manager tool after an experiment. Figure 7 shows a printscreen of the Morae Manager with the recorded desktop of a subject.

3.1.3. Data Set

The usability study was based on the knowledge base containing geographical information about the US from the Mooney Natural Language Learning Data\(^\text{11}\) provided by Tang and Mooney [69]. We chose the data set because it covers a domain that can easily be understood by casual users and does not demand expert knowledge [10]. To make the original knowledge base accessible to our ontology-based interfaces, we translated the Prolog knowledge base to OWL and designed a class structure as meta model, which is represented as graph in Figure 8. The resulting geography OWL knowledge base contains 9 classes, 11 datatype properties, 17 object properties, and 697 instances.

---

\(^{10}\)http://www.techsmith.com/morae.asp

\(^{11}\)http://www.cs.utexas.edu/users/ml/nldata.html
3.1.4. Data Analysis
The data we collected in the usability study was analyzed quantitatively as well as qualitatively. For the quantitative analysis, we used the SUS scores and the usual statistical methods *ANOVA*, *T-test*, and *Mixed Linear Regression Models* as available in the statistics software *R* [12] and its *lme4-package*: [13]

- In cases where we compared the results of only two systems, we used Student’s *T-tests*, as they are applied when checking two data sets. Given two data sets, each characterized by its mean, standard deviation, and number of data points, we can use a T-test to determine whether the means are in fact distinct or not.

- *ANOVA* or *Analysis of Variance* is a statistical method of checking, if there is a relationship between two or more data sets. It is a test among multiple data sets simultaneously, and basically tells whether the results from an experiment were due to random chance or not. With ANOVA we can, for example, determine if there is a significant difference between the quality assessment our subjects gave one system and the assessments they gave the other three systems. When comparing the measured assessments that were given by each subject with the four interfaces using ANOVA, we can identify one independent variable or factor “interface” and, hence, we have a single factor ANOVA (also called one-way ANOVA) with four levels (i.e., the four values of the factor “interface” which are the four interfaces NLP-Reduce, Querix, Ginseng, and Semantic Crystal).

- For statistical tests such as the T-test or ANOVA, R outputs *p-values* (probability values). A p-value measures the significance of a statistical test and indicates whether there is statistical evidence to say that some measurement set differs significantly from another measurement set. The p-value is a value between 0 and 1. A p-value of 0.05 means that, if you say that there is a difference between two data sets, you have an error rate of 5% that the difference relies on random chance alone. The smaller the p-value is, the safer it is to say that there is a difference between two or more data sets [59]. Usually, a p-value smaller than 0.05 indicates statistical significance [62] meaning that two or more data sets do not differ by chance, but by statistical evidence. Consequently, a p-value equal or greater than 0.05 indicates no level of significance. Two or more data sets that are associated with a p-value of 0.71, for example, will not be considered to be of statistical difference.

- Furthermore, we used *Mixed Linear Regression Models* to analyze the data collected from the study. Mixed Linear Models are statistical Re-

---

12 [http://www.r-project.org/](http://www.r-project.org/)
gression Models to model means, variances, and covariances in data. Basically, linear regression is a method that models the relationship between a dependent variable (also called response variable) and independent variables (also called explanatory variables or regressors), such that the independent variables have some influence or impact on the outcome of the dependent variable. With Mixed Linear Regression Models we can, for example, find out if the independent variables knowledge of informatics, knowledge of linguistics, knowledge of formal query languages, and knowledge of English significantly influenced the dependent variable time that was needed to reformulate and enter the queries into the four systems. Statistical evidence is again indicated by a p-value less than 0.05 and provided in the lme4-package of the R-Software [27,28].

When qualitatively analyzing the data we collected by the comparison questionnaires, we looked for patterns for categorization and peculiar incidents [58]. Additionally, we tried to satisfy the internal as well as the external validity when interpreting the results and drawing conclusions [15].

3.2. Results of the Usability Study

The results concerning the time that the users spent to reformulate the queries in our usability study are summarized in Table 1.

<table>
<thead>
<tr>
<th>System</th>
<th>Average time for all 4 queries</th>
<th>Average time per query</th>
</tr>
</thead>
<tbody>
<tr>
<td>NLP-Reduce</td>
<td>2 min 39 sec</td>
<td>23.54 sec</td>
</tr>
<tr>
<td>Querix</td>
<td>4 min 11 sec</td>
<td>29.31 sec</td>
</tr>
<tr>
<td>Ginseng</td>
<td>6 min 06 sec</td>
<td>34.82 sec</td>
</tr>
<tr>
<td>Semantic Crystal</td>
<td>9 min 43 sec</td>
<td>89.53 sec</td>
</tr>
<tr>
<td>p-value</td>
<td>1.56e-26</td>
<td>4.91e-40</td>
</tr>
</tbody>
</table>

Table 1

Results of the average time that users needed to reformulate all four queries with each interface and the average time they spent per query with each interface. The p-value was calculated by a single factor ANOVA with four levels.

Most strikingly, our results are much more than highly significant (statistical significance, if p < 0.05), 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, which can be full sentences, sentence fragments, or just keywords with no restrictions, 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 linearly increasing average time that was spent per query (column 2 in Table 1) nicely mirrors the increasing degree of formality and control of the interfaces' query languages (see the Formality Continuum in Figure 2).

<table>
<thead>
<tr>
<th>System</th>
<th>Average number of queries</th>
<th>Average success rate (biased)</th>
<th>Average failure rate (biased)</th>
</tr>
</thead>
<tbody>
<tr>
<td>NLP-Reduce</td>
<td>7.94</td>
<td>69.27 %</td>
<td>30.73 %</td>
</tr>
<tr>
<td>Querix</td>
<td>7.75</td>
<td>77.08 %</td>
<td>22.92 %</td>
</tr>
<tr>
<td>Ginseng</td>
<td>11.06</td>
<td>63.54 %</td>
<td>36.46 %</td>
</tr>
<tr>
<td>Semantic Crystal</td>
<td>7.02</td>
<td>54.86 %</td>
<td>45.14 %</td>
</tr>
<tr>
<td>p-value</td>
<td>3.92e-06</td>
<td>1.06e-05</td>
<td>2.54e-05</td>
</tr>
</tbody>
</table>

Table 2

Results of the average number of queries that was required to find the answers for all four queries with each system and the success/failure rates of the queries from the test users' point of view. These “biased” results are measured against the subjects' stated belief of success in the experimental situation. The p-value was calculated by a single factor ANOVA with four levels.

In Table 2 the average number of queries to find answers to the four question fragments and the success respectively the failure of these queries are presented. We can see in column 1 that it took users 7.02 queries on average to find an answer 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 control causing users to repeatedly reformulate 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 (biased) in Table 2 indicate how many of the four queries retrieved a satisfying answer from the users' perspective (i.e., the user thought that she/he had
found the correct answer). Though Semantic Crystal in fact provides more precise answers than its competitors (see also Table 3), the success rate of only 54.86% is due to inappropriate and invalid query formulations. The significantly best success rate achieved by Querix from the users’ point of view seems to be 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 natural language answers of Querix more because the linguistic answer created the impression that the system “understood” the query.

When reviewing the recorded log files in order to find out what the success and failure rates in fact were from an objective point of view (unbiased), we discovered that there is no discrepancy between the subjects’ and the objective success/failure rates in Ginseng and Semantic Crystal. There was one case in which NLP-Reduce returned an incorrect answer, though the user thought it was correct. Astonishingly, Querix produced 3 incorrect answers to the total of 420 queries posed by all subjects, but the subjects were actually satisfied with the answers. We traced the false impressions to answers such as “There are 25.” for which we already discussed that they created confidence towards the interface. In contrast to confidence, the natural language answers apparently also created skepticism, since most queries that were entered in order to double-check previous answers occurred with Querix (i.e., 22), whereas 17 checking queries were entered with NLP-Reduce, 3 with Ginseng, and none with Semantic Crystal. The number of checking queries almost inversely mirrors the increasing degree of formality from Querix/NLP-Reduce to Ginseng and Semantic Crystal, leading to the hypothesis that formality rather than naturalness may create a notion of confidence towards a system.

Additionally, we detected that 15 queries did not lead to a satisfying answer in NLP-Reduce due to typos. There were also 15 queries with typos in Querix, and 2 in Semantic Crystal. As such, the unbiased success and failure rates achieved by the test users with all queries and all interfaces are shown in Table 3. The results reveal that NLP-Reduce performs best with regard to correct answers from an objective point of view: the result, however, is not significant. Querix was even outranked by Ginseng, although Querix appeared to perform best from the users’ perspective. The ranking of the objective failure rate results remains the same as the biased failure rates.

We also examined the success and failure rates with relation to the time that was spent for reformulating and entering the query fragments. In Table 4 we see that, when relating the success and failure rates to the time it took users to reformulate and enter the query fragments, the significantly highest success rate was obtained with NLP-Reduce and the lowest with Semantic Crystal (p = 1.47e-16). Again, the results confirm that most time is required when working with the interface that shifts the query formulation burden to the user (i.e., makes it difficult to construct a query that the system can parse easily). The failure rates related to the time spent for entering the queries inversely reflect the same result (p = 1.84e-08). Consequently, the success rates of the more formal and, therefore, more precise query languages cannot balance out the additional time that is needed to compose queries with these query languages.

Using the Mixed Linear Regression Model analysis, we found that the order in which the interfaces were presented to a user slightly influenced the time that was spent on query reformulation: If a tool was presented last, users spent an average of 37.5 seconds more per query than if the tool was presented first (p = 0.019). This finding contradicts the general belief that users tend to become increasingly impatient as an experiment proceeds [54,58]. On the recorded desktop videos, it looked as if the users were eager to ‘get it right’ during the last iteration of the experiment.

<table>
<thead>
<tr>
<th></th>
<th>average success rate (unbiased)</th>
<th>average failure rate (unbiased)</th>
</tr>
</thead>
<tbody>
<tr>
<td>NLP-Reduce</td>
<td>68.75 %</td>
<td>25.00 %</td>
</tr>
<tr>
<td>Querix</td>
<td>59.38 %</td>
<td>15.10 %</td>
</tr>
<tr>
<td>Ginseng</td>
<td>63.54 %</td>
<td>36.46 %</td>
</tr>
<tr>
<td>Semantic Crystal</td>
<td>54.86 %</td>
<td>44.09 %</td>
</tr>
<tr>
<td>p-value</td>
<td>0.072</td>
<td>1.36e-08</td>
</tr>
</tbody>
</table>

Table 3
Success and failure rates for queries entered by the users from an objective point of view meaning that the interfaces actually generated correct or false answers. These “unbiased” results measure success using the ground truth of the data set. The p-value was calculated by a single factor ANOVA with four levels. (The rates do not add up to 100% due to typos by the subjects.)
Table 4
Results of the average number of successful and failed queries per minute. A successful query retrieves a satisfying answer, whereas a failed query does not retrieve a satisfying answer from the subject’s point of view. The p-value was calculated by a single factor ANOVA with four levels.

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 did: With every year a user’s age grows, the average time to reformulate a query increases by 3.30 seconds (p = 0.010).

Table 5
Results of the System Usability Score questionnaires. The SUS 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. The p-value was calculated by a single factor ANOVA with four levels.

Table 6
Results of the comparison questionnaires, in which the test users indicated which interface and query language (QL) they liked best as well as which interface and query language they liked least.

Seeing the SUS-scores, the results of the comparison questionnaires are no surprise. 66.67% of the users liked the Querix interface best and only 2.8% liked it least (columns 1 and 2 in Table 6). Querix obtained almost the same feedback for its query language (QL) in particular, this time reaching statistical significance with p = 0.0075 (columns 3 and 4 in Table 6). All of these results are highly significant as shown by a χ² test.

Even though 60.42% of the users disliked Semantic Crystal as query interface when comparing it to the other NLIs, a surprising portion of 14.58% assessed Semantic Crystal as favorite interface. The graphically displayed knowledge base was explicitly found helpful by five users. Only 12.50% liked NLP-Reduce best and 6.25% Ginseng. With respect to the query language, the results are different: here, the query language of Semantic Crystal received the lowest rating (4.17%), and the query languages of NLP-Reduce (18.75%) and Ginseng (16.50%) were clearly preferred, showing the same ranking as the results of the SUS scores.

When viewing the results of query language liked least (column 4 in Table 6), the keyword-based QL provided by NLP-Reduce was disliked twice as much (25.00%) than the controlled query language of Ginseng (12.50%). 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 control that can enable guidance.

The Mixed Linear Regression Model analysis showed that with each second spent more with a system, the SUS score dropped by 0.06 (p = 1.79e-09), whereas the number of queries used, the suc-
cess/failure rates, and the order of the queries did not influence the SUS ratings. It seems that the factor time is a very important issue for casual end-users when judging a user interface. The order in which the interfaces were presented to the user also had 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 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 questionnaires in order to indicate their motivations for their choices of the best and least liked interfaces, the most often-named comments for each interface were the ones presented in Table 7. The number of times a comment was given is indicated in parentheses.

Obviously, NLP-Reduce and Querix were deemed as simple to use, but NLP-Reduce’s answer presentation which includes complete URIs was disliked. Although the use Ginseng was easily learnt, it was considered to be too restrictive and, therefore, too complicated because the users were obliged to follow a fix vocabulary and prescribed sentence structures. The comments for Semantic Crystal are controversial as expected: While some users liked the graphical display of what was possible to ask and were intrigued by the different approach, most subjects rated it too complicated and too time-consuming.

In addition, the comparison questionnaire specifically asked the test users for which query language they liked best/least and why. The comments that were given for each query language are listed in Table 8. Again, the number of times a comment was given is indicated in parentheses.

The comments most often given for the query languages of the four systems are consistently contradictory. While the use of keywords in NLP-Reduce was rated positively by some users, for others NLP-Reduce’s query language was unclear. Most users liked the every-day and clear language of Querix, whereas two subjects found it cumbersome that one has to enter complete sentences. Ginseng’s query language was deemed both assisting as well as controlling. Finally, the graphical query composition language of Semantic Crystal was appealing to some users by its playful character, but most subjects clearly expressed an aversion to the query language because it is too complicated and laborious.

The following comments were found striking enough to list them individually, even though they only appeared once:

(i) NLP-Reduce is too formal.
(ii) The language is lost in NLP-Reduce.
(iii) It is not clear what language can be used in NLP-Reduce.
(iv) With NLP-Reduce, I can use normal speech patterns.
(v) NLP-Reduce’s language is too unrestricted to give confidence.
(vi) No structured queries in NLP-Reduce.
(vii) Querix has clear sentence structures. Its language is everyday language.
(viii) Semantic Crystal is fun, but too laborious for every day.
(ix) Semantic Crystal is more difficult to use than a system allowing sentences.
(x) The language of Semantic Crystal is very natural.
(xi) Ginseng and Semantic Crystal appear innovative, but too restrictive. NLP-Reduce is too relaxed. Querix is a good compromise.

Noticeably, there are again conflicts in the comments with regard to which query language is regarded as formal and which as natural. Consider the first (no. 1) and the tenth comment (no. 10), for example. They argue exactly the opposite of where we placed the query languages in the Formality Continuum indicating that there may be different notions of formality/structuredness and naturalness/freedom (as already mentioned in Footnote 4). Furthermore, NLP-Reduce is highly controversial: while some declare its query language to be confusing (no. 2, 3, 5), others find it very natural (no. 4). We can count five comments (no. 2, 3, 5, 6, 11) asking for more structure in the query language of NLP-Reduce; these are prime examples of the habitability problem.

Comment no. 8 about Semantic Crystal (“fun, but too laborious for every day”) raises the issue that the usefulness of a query interface may depend on how often the interface is used and, to carry the idea a bit further, for which tasks.

The structure that is imposed by Querix’ language seems to be accepted by end-users (comments no. 7 and 11). They may experience the structure of natural language sentences as natural and flexible enough, they do not even perceive it as structure. However, a remarkable number of users do notice the structure and appreciate it as assistance, therefore supporting our Habitability Hypothesis.
positive comments on the interface | negative comments on the interface
--- | ---
NLP-Reduce  | + the simplest interface (5)  
  | + similar to common search engines (2) 
  | - bad presentation of results (5) 
  | - too relaxed (2)
Querix | + simple to use (19) 
  | + free and unrestricted query language (7) 
  | + clear and good answers (4)
Ginseng | + simple (3) 
  | + comprehensible (2) 
  | - too restrictive (4) 
  | - too complicated (3)
Semantic Crystal | + graphical display of elements (5) 
  | - too complicated (18) 
  | + different (3) 
  | - too laborious (7)

Table 7
The comments most often named by the test users for each query interface. The numbers in parentheses indicate how often the comment was given.

positive comments on the query language | negative comments on the query language
--- | ---
NLP-Reduce  | + I can use keywords (5) 
  | + no thinking required (2) 
  | + robust to input (2) 
  | - query language not clear (4) 
  | - no superlative forms (3)
Querix | + I can use my language (8) 
  | + simple to use (5) 
  | + clear language (2) 
  | - one has to enter complete sentences (2)
Ginseng | + assisting (4) 
  | + simple (3) 
  | - too restrictive (3)
Semantic Crystal | + playful (2) 
  | - too laborious (7) 
  | - too complicated (5) 
  | - cumbersome (4)

Table 8
The comments most often named by the test users for each query language in particular. The numbers in parentheses indicate how often the comment was given.

Statement no. 11 “Ginseng and Semantic Crystal appear innovative, but too restrictive. NLP-Reduce is too relaxed. Querix is a good compromise.” nicely summarizes and confirms the concept of our Formality Continuum.

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 allowing full English questions with a limited set of sentence beginnings was judged to be the most useful and best-liked query interface. This finding partially contradicts another usability study investigating different query languages and showing that students generally preferred keyword-based search over full-questions search [61]. 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. Note that the contradiction is only partial, as our users perceived Querix to exhibit superior performance over NLP-reduce (which it did not objectively). Hence, the subjects in both studies seem to follow the heuristic of using the system for which they get the better query performance. The qualitative results mirror these findings.

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, meaning that the query language of NLP-Reduce actually offers more freedom and less control than Querix. Additionally, the beginnings of the sentences that are accepted by Querix are limited to a set of six sentence beginnings, which restricts its query language even more. We can think of two reasons for the comment:

- 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.
- 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.

Indeed this interpretation is supported by the explicit statement that users liked Querix as they can use “their query language” and they perceive this language to be “clear”.

The analysis of the results reveal a divergence between the perceived and the actual correctness of answers. Systems such as Querix generating natural language answers and engaging users in some kind of natural language feedback or clarification dialog apparently lead to the impression that the interface “understands” the user, therefore creating confidence towards the returned answers. We think that this is one of the reasons that our subjects rated Querix best with regard to the SUS score as well as by directly naming the system they liked best. Though NLP-Reduce exhibited a better (but not significant) objectively successful retrieval performance, it was rated less favorably than Querix.

Therefore, retrieval performance seems not to be the primary criterion that creates confidence towards an interface in general. The preference for Querix and its full-sentence query language was, however, extremely significant. Hence, we doubt that the impact of Querix’s interactive nature solely explains its 20 point lead. Indeed, the large number of other positive qualitative comments (26) and the lack of any negative ones about the user interface (Table 7) as well as the large number of positive comments explicitly regarding Querix’s query language (Table 8) indicate that the interactive style at best mitigated Querix’s dominant SUS score. As such, the result of the study strongly suggests that Querix was the best-liked and best-rated query interface.

Although the success rate that was achieved by the subjects with Semantic Crystal was the lowest of the four interfaces, we actually think that this is a good result when considering that our subjects used the interface for the first time, that they were completely unfamiliar with ontology and SPARQL issues, and that “they were thrown in the deep end of query composing tasks” with Semantic Crystal (after very brief instructions, which were even given on paper and not by a live system demo). Furthermore, though Semantic Crystal was assessed as difficult and laborious to use, some users pointed out the big advantage of graphically displayed knowledge bases and queries. Consequently, we should consider interfaces to Semantic Web data that offer a combination of graphically displayed as well as keyword-based and full-sentence query languages. A user could then choose between different querying possibilities. And we might have to think of adequate natural language answer generation components [2], which seem to increase a user’s trust in a system and the overall user satisfaction.

To get back to our Habitability Hypothesis, we first want to recall it: The Habitability Hypothesis proposes that query interfaces to the Semantic Web should impose some structure on the casual end-user to guide the query formulation process, but not overly control the user with an excessively formalistic language, therefore, alienating the user. As such, the best solutions for casual or occasional end-users should lie somewhere in the middle of the Formality Continuum (cf. Figure 2 on page 4), therefore easing the query formulation task without complicating it.

The usability study supported the hypothesis in terms of the perceived (or biased) success rate of users, lowest perceived (or biased) as well as actual (or unbiased) failure rate, the SUS score, the interface preference, and the query language preference. In all of these results Querix, which requires the structure of full, grammatically correct English sentences with a limited set of sentence beginnings significantly outperformed the other three interfaces when evaluated by 48 casual end-users. The structure that is imposed by Querix, however, is not perceived as a formal or restricting structure but as a natural, guiding structure, which is plausible, because the structure of natural language is not noticed in everyday use either. Note, however, that this success comes at a “price”: entering Querix queries is significantly slower than with NLP-Reduce. On the other hand, the structure that was imposed by Ginseng was evidently too restrictive as evidenced by the slower execution time.
The lack of support for the hypothesis in the actual (or unbiased) success rate is puzzling given the significant and strong support in terms of the average actual (or unbiased) failure rate. This divergence between the user’s preferences and failure rates on one side and the success rates on the other side clearly requires more investigation. Nonetheless, from a user preference point of view and given a certain speed preference tradeoff, the best solutions for casual end-users seem to lie towards the middle, but certainly on the natural side of the Formality Continuum.

3.4. Limitations and Future Work

Whilst our results suggested confirming the habitability hypothesis 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 and their results have been presented in [42].

Concerning valid and even more fine-grained 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 query interfaces was different. As mentioned this has likely mitigated Querix’s lead in the SUS score. Given the qualitative statements we doubt that the impact is so profound that it solely explains Querix’s lead.

As always in benchmarking experiments involving human-computer interaction there might have been issues with our operationalization of the hypotheses. For example, our choice of Semantic Crystal as the formal query tool rather than some other formal notation may have prompted users to dislike it as they might be adverse to multi-colored graphs. Thus, we would like to motivate further studies with different settings and users to verify our results.

Also, our evaluation was limited to answering factual question answering. While there are other types of querying we believe that this limitation is prudent for a user evaluation, as the introduction of different types of queries would further complicate any generalization. Nonetheless, the limitation to these kinds of queries may conceal an interaction between the types of queries asked and the appropriate tool to use. Consider the contrast between answering factual queries and explorative information exploration: is the best interface for the first task also good for the second and vice-versa? There is good reason to believe that this is not the case, as information exploration is oftentimes a more serendipitous task and may require a different kind of query approach and answer presentation. Unfortunately, the data we gathered is not suitable to shed light on this issue, which we hope to explore in the future.

We limited ourselves to four interfaces and four queries for each query tool 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 controlled usability studies [40,47,61]), 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.

In the same spirit it could be argued that we should have chosen off-the-shelf NLI’s rather than systems that we developed ourselves. The main reason for developing our own systems was that we could develop their functionality to deliberately place them at a specific point along the formality continuum. This would have been difficult with a system from another provider, as those systems usually aim at providing the best possible solution rather than a solution limited to specific kinds of functionality. Consequently, whilst we agree that a general evaluation of NLIs for Semantic Web data would have been served better with employing existing tools, our goal – exploring the precise kind of query language support most suitable for the task – significantly profited from the use of tools with precisely controlled query languages.

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 [74], we will, therefore, develop an interface for casual end-users that offers a combina-

---

14 We would like to thank the anonymous reviewer for pointing out this issue.
tion of a graphically displayed and a natural language query language which embeds both keywords as well as full sentences. A user can then choose which query language to use according to personal preferences or different information seeking tasks as pointed out by the test user comment no. 8 on page 15. A thorough evaluation of this new interface in a real-world setting (i.e., a real Web interface to a highly frequented Web site) would point out its usefulness. Implementing these ideas will be our future undertakings.

Last but not least, our study does not address the problem of accessing multiple ontologies simultaneously. Indeed, it assumes that any matchmaking issues between ontologies are resolved and that the NLI has access to a well-aligned set of ontologies. In a real Semantic Web environment, alignments between ontologies may be brittle, which could have an impact on the NLI’s performance. When generalizing our findings one may have to take alignment complications into consideration. Given the focus of the study on query languages and user perception rather than retrieval performance we believe that it is a reasonable assumption to be made.

4. Related Work

NLIs have repeatedly been developed since the 70s, but oftentimes with moderate success [3,18,53,71], 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. Besides, more and more people from a wide population access information stored in a variety of formal repositories through Web browsers, PDAs, cell phones etc. Around 2000, researchers have again started to address the task of building NLIs and a number of well-performing NLIs to databases emerged [35,1,25,57,36,51,30,37]. Considering the difficulties with full NL, it seems comprehensible that restricted natural language or menu-guided interfaces have been proposed by some approaches [7,17,52,71,64,31,38,67,46]. The popularity of the Semantic Web created several NLIs that provide access to ontology-based knowledge bases [41,24,33,49,26,73,30,20,31,38,46]. A large number of these systems base on controlled natural language [12,11,30,31,38,46] and, mostly, propose an alternative editing format to OWL [9,31,38,46]. One system proposed a conversational interface to Semantic Web services rather than the knowledge base itself mediated through IRC bots [33].

Most of the projects in the area of NLIs and, therefore, the evaluations of the systems mainly focus on retrieval performance and/or the portability dimension. Our work emphasizes the usability dimension, which is only investigated by some studies (e.g., [31,37,38,46]). Most of these studies, however, are limited to comparing their own approach either in isolation against some absolute measure (e.g., [37]) or against some formal query language (e.g., [31,38,46]). Our study, in contrast, compares four different query interfaces – one of which being formal, the others being similar to some of the related work – with the goal of eliciting the the core features making an NLI’s successful for casual users.

As a representative sample we will now discuss four recent NLI projects that conducted a usability study: ORAKEL [19,20], Squirrel [26], CHEST [61], and the tourism platform by Dittenbach et al. [25].

ORAKEL by Cimiano is a portable NLI to structured knowledge bases that is ontology-based in two ways [20]. 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 27 users (26 computer scientists and one graphic designer) 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. is a search and browse interface to semantically annotated data [26]. 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
The domain as well as the linguistic relationships between the concepts of the domain. The ontology also contains parametrized SQL fragments that are used to build the SQL statements representing the natural language query. A lightweight grammar is used to analyze the structure of a question and to combine the SQL statements in order to obtain one coherent SQL query that can be executed over the database. The interface was integrated into the Tiscover platform\footnote{http://www.tiscover.at/} and online for ten days. The goal was to collect a broad range of questions and to find out what users really wanted in an unsupervised field test. 1425 unique queries were collected in both languages German and English.

In 57.05\% of the queries, users formulated grammatically correct and complete queries, whereas only 21.69\% used the interface like a keyword-based search engine. The remaining queries (21.26\%) were question fragments such as “double room for two nights in Vienna.” It is reported that the users accepted the NLI and were willing to type more than just keywords to search for information; some queries even consisted of more than one sentence. The authors assume that users are more specific formulating a query in natural language than with keywords, a conclusion we can confirm on the basis of our controlled usability experiment.

In general, the study shows that the complexity of natural language questions is relatively low, i.e., the number of concepts that are combined in queries is low (the average number of relevant concepts occurring in the queries was 3.41 compared to a median of 2 in web searches\cite{65}), and the questions that are formulated on the basis of combining concepts are of simple syntactical manner. The main conclusion drawn from the study is that NLIs are especially useful in case of inhomogeneous user groups as with the tourism platform. Hence, the complexity of the sentences expressing the user’s information need is tractable with shallow language processing techniques. Motivated by these findings, we more than ever tried to keep our NLIs simple in design and avoid complex configurations by restricting the query language to some extent, since users tend to not enter complex questions and even appreciate the guidance of a controlled or restricted natural language.

The approaches in the field of NLIs nicely show that they can successfully tackle the performance and portability dimension. As such, they comple-
ment our findings, which focus on the usability and usefulness dimension. Some of the approaches that investigate usability confirm our findings that NLIs are useful in a casual end-user’s point of view and particularly useful for heterogeneous user groups. However, only very few recent usability studies concerning NLIs to ontology-based data exist [37, 31, 38, 46], studies benchmarking different NLIs are, as far as we could ascertain, not inexist-ent in the field of the Semantic Web. Hence, more work is needed regarding NLIs to Semantic Web data and further comprehensive usability studies to investigate the casual end-users’s perspective.

5. Conclusions

Natural language search interfaces hide the for-mality of an ontology-based knowledge base as well as the executable query language from end-users by offering an intuitive and familiar way of query for-mulation. For the successful use of an NLI, however, users need to know what is possible to ask, since these systems still need carefully developed query statements. This paper proposed the Formality Continuum, which suggests that we can achieve supe-rior user support by imposing some restrictions on the user’s natural language input to guide the query formulation process. The goal was to investigate to what extent the query language should be restricted or controlled and, additionally, if NLIs are actually assessed as useful from the casual end-users point of view. In contrast, most studies concerning NLIs to structured data aim at achieving high-quality re-trieval performance and transportability.

Focusing on the usability aspect, we conducted a study with 48 real-world users and four interfaces featuring four different query languages. The results showed that a full-sentence query option with a limited set of sentence beginnings was significantly preferred to keywords, a menu-guided, and a graphi-cal query language. As such, the best solutions for casual or occasional end-users lie towards the mid-dle, but on the natural side of the Formality Continuum. The structure of natural language was perceived as familiar, convenient, and guiding, also al-lowing more semantically refined queries than just keywords. NLIs offering an adequately guiding query language can, therefore, be considered to be indeed useful for casual or occasional end-users and espe-cially for heterogenous user groups. We believe that our study generally shows the potential of NLIs for end-user access to the Semantic Web or other data repositories and provides some evidence for the use-fulness as end-user query languages in general, pro-viding a chance to offer the Semantic Web’s capa-bilities to the general public.

6. Acknowledgments

The authors would like to thank Ray Mooney and his group from the University of Texas at Austin for having generously supplied the data set, on which the usability study was based (http://www.cs.utexas.edu/users/ml/nldata.html). We would also like to thank the anonymous reviewers for their extremely helpful comments.

References

[7] URL http://dx.doi.org/10.1080/10447310802205776