Ginseng: A Guided Input Natural Language Search Engine for Querying Ontologies

Abraham Bernstein, Esther Kaufmann, Christian Kaiser, Christoph Kiefer
Department of Informatics, University of Zurich, Switzerland
{bernstein, kaufmann, kiefer}@ifi.unizh.ch

1 Introduction

The Semantic Web presents the vision of a distributed, dynamically growing knowledge base founded on formal logic. Common users, however, seem to have problems even with the simplest Boolean expression [1]. So how can we help users to query a web of logic that they do not seem to understand? We address this issue by presenting *Ginseng, a **g**uided **i**nput **n**atural language **s**earch **eng**ine* for the Semantic Web.

2 The Ginseng Search Engine

Ginseng provides a quasi natural language querying access to any OWL knowledge base. The main difference between Ginseng and full natural language interfaces [2] is that Ginseng does not use any predefined vocabulary and does not try to interpret the queries (logically or syntactically). Instead, Ginseng "only knows" the vocabulary defined by the currently considered ontologies. All ontologies are stored in a Jena inferencing model (*OWL_MEM_RULE_INF*). The vocabulary is closed and the user has to follow it. This can limit the user's possibilities in general but ensures that every query can be answered. Note that the vocabulary grows with every additionally loaded ontology.

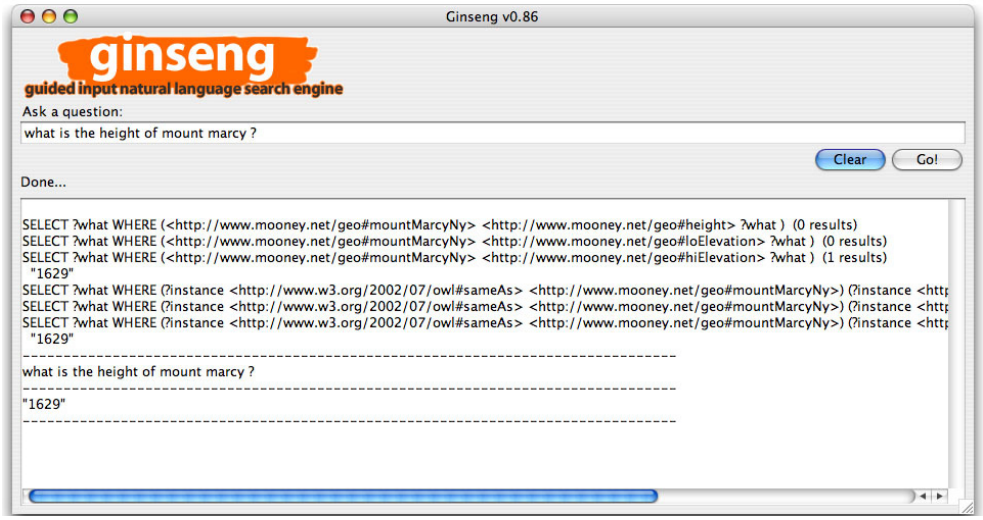

Fig. 1. The Ginseng user interface after executing a query

Ginseng allows users to query any OWL knowledge base using a guided input natural language which strongly resembles plain English. As shown in Fig. 1, the user enters the query into a free form entry field. When the user starts typing, the system predicts the possible completions of what the user enters (similar to completion suggestions in Unix shells) and presents the user with a choice popup box. While the user is in the middle of a word, the popup offers suggestions on how to complete the current word (Fig. 2). Obviously, the possible choices get reduced as the user continues to type. Ginseng, thus, guides the user through the set of possible queries while avoiding ungrammatical queries. When a query is completed, Ginseng translates it into SPARQL statements, executes it against the existing ontology model, and displays the generated SPARQL query and the result(s) of the query to the user (as shown in Fig. 1).

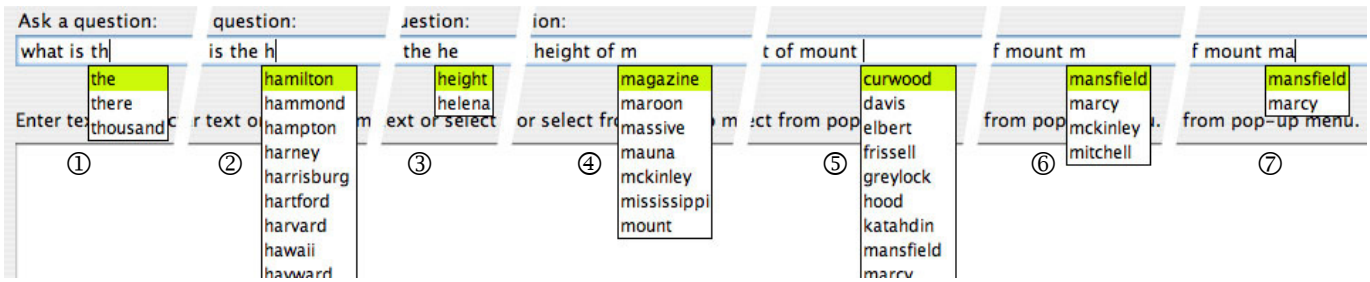

Fig. 2. Ginseng query-completion popup/choice window (numbers added)

Demo at 2006 Jena User Conference, Bristol, UK, May 2006.

Ginseng's architecture has three parts: a partially dynamically generated *multi-level grammar*, an *incremental parser*, and the *ontology-access layer*.

The **multi-level grammar** consists of a *static part* specifying the generally possible query sentence structures and a *dynamic part* which gets generated from the used ontologies. The *static grammar rules* provide the basic sentence structures and phrases for English questions. It handles general question structures such as "What are the capitals of the states that border Nevada?" as well as closed questions (e.g., "Is there a city that is the highest point of a state?" typically resulting in an answer of "yes" or "no") or questions resulting in numbers (e.g., "How many rivers run through Texas?"). Furthermore, it provides sentence construction rules for the conjunction or disjunction of two or more sentence parts. The static grammar consists of about 120 mostly empirically constructed domain-independent rules.

The *dynamic grammar rules* get generated from the loaded OWL ontologies. When starting Ginseng, all ontologies in a predefined search path are loaded into Jena. For each ontology the necessary dynamic rules are generated to extend the static part of the grammar. Basing on Jena, Ginseng essentially loads each ontology into an inferencing model and generates a grammar rule for each class (*OntClass.class*), individual (*Individual.class*), object property (*ObjectProperty.class*), and data type property (*DatatypeProperty.class*) by the use of the corresponding list-methods of the ontology model class (*OntModel.class*). These rules provide the elements that are displayed as possible choices in the popup boxes (examples 2 to 7 in Fig. 2).

Ginseng also allows that synonyms of the labels (*rdf:label*) used in the ontology models can be included by annotating the ontology with Ginseng tags (from the ginseng namespace). As such, Ginseng also generates a dynamic grammar rule for each synonym. While such annotations aren't necessary for Ginseng to run correctly, they do extend the vocabulary of Ginseng and increase its usability. Additionally, they reduce the limitation that Ginseng's approach, to some extent, depends on the choice of vocabulary when the ontology was built; the more meaningful the labels of an ontology, the wider and more useful the vocabulary provided by Ginseng.

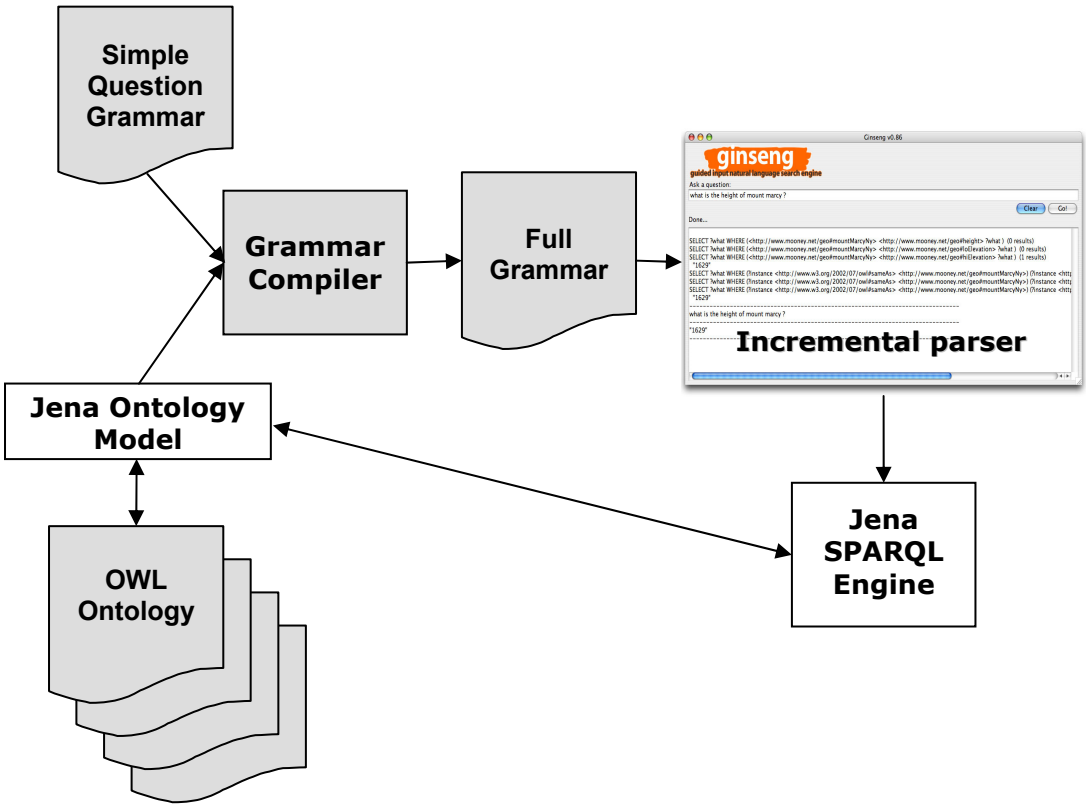

Fig. 3. Ginseng's architecture

The full grammar is then used by the **incremental parser** in two ways: First, its specifies the complete set of parsable sentences, which can be used to provide the user with alternatives during the entering of queries as described above and prevents incorrect queries. Second, the grammar contains information on how to construct the SPARQL queries. Thus, a complete parse tree is also used to generate the resulting SPARQL statements.

Finally, Jena's SPARQL interpreter is used as **ontology access layer** to execute the query. When the execution of a query is completed, Ginseng displays the generated SPARQL query and the result(s) to the user.

3 Preliminary Evaluation

To evaluate the power of Ginseng's grammar we ran 880 queries from the geographical Mooney knowledge base [3]. We found that Ginseng could execute 40% of the queries right out of the box. After removing ungrammatical queries (e.g., "Which state is the largest city in Montana in?") and queries that cannot be formulated in SPARQL yet, for example queries that require aggregate functions (e.g., "How many cities named Austin are there in the USA?"), Ginseng parsed 67.1% of the queries without any modification of the grammar.

The queries that could be parsed resulted in a precision of 92.8% and a recall of 98.4%. The excellent recall shows nicely that Ginseng can capture the *Gestalt* of a natural language query without any complicated logic processing. The slightly worse precision is probably a result of Ginseng's limited ability to actually "understand" the queries: it "only" extracts part of the information in the natural language. Yet, the overall retrieval performance (F-measure = 0.955) reconfirms the power of Ginseng's simple design.

4 Limitations and Future Work

We can think of four limitations to our work. First, and most obviously, Ginseng cannot process all natural language questions due to its construction. Nevertheless, as a usability evaluation with 20 subjects showed, this limitation didn't seem to bother the users seriously (we do not discuss this usability study here due to space limitations). Second, we ran the 880 queries from only one of the three Mooney knowledge bases [3]. We plan to include the other two knowledge bases in Ginseng. The complete corpus of 1770 questions might allow us to learn the set of static grammar rules (instead of creating them manually). Third, in its current state SPARQL does not provide aggregate functions yet. Ginseng, therefore, doesn't answer questions such as "What is the number of rivers in California?" even though such questions can be entered and parsed. We hope that SPARQL will soon be enabled with this valuable feature as well as other new features (e.g., GROUP BY). Last, even though Ginseng could execute 67.1% of the geographical questions, we believe that we can clearly improve this result by applying query expansion methods. To achieve this goal, we plan to emphasize and improve the integration of synonyms from WordNet for classes, individuals, and properties in the generation process of the dynamic grammar rules, which would automatically lead to a larger vocabulary that could be queried by Ginseng.

5 Conclusions

Based on the structure of arbitrary OWL ontologies, Ginseng relies on a simple, dynamically extendable grammar that can be used to parse quasi-natural English queries. The result is a simple and computationally cheap but highly adaptive approach to guided English ontology querying – a potentially important component for bridging the gap between real-world users and the logic-based underpinnings of the Semantic Web.

References

1. Spink, A., et al.: Searching the Web: The Public and Their Queries. Journal of the American Society for Information Science and Technology 52/3 (2001) 226-234
2. Androutsopoulos, I., Ritchie, G.D., Thanisch, P.: Natural Language Interfaces to Databases - An Introduction. Natural Language Engineering 1/1 (1995) 29-81
3. Tang, L.R., Mooney, R.J.: Using Multiple Clause Constructors in Inductive Logic Programming for Semantic Parsing. ECML-2001. Freiburg, Germany (2001) 466-477