Transportable Natural Language Interfaces for Taxonomic Knowledge Representation Systems

Lutz Prechelt, Finn Dag Buø, Rolf Adams
(prechelt|finndag|adams@ira.uka.de)
Institut für Programmstrukturen und Datenorganisation
Universität Karlsruhe, W–7500 Karlsruhe, Germany

Appeared in the proceedings of the
Ninth IEEE Conference on
Artificial Intelligence for Applications
Orlando, Florida, March 1-5, 1993

A general approach is presented for building transportable natural language interfaces for question answering systems based on a KL-ONE-like knowledge representation. An example system, YAKR*, is described: The YAKS knowledge representation of concepts and relations is annotated with minimal syntactic information to generate a semantic case frame grammar with inheritance of cases. The generated grammar directs a case frame parser, which processes written input into instantiated case frames. These instantiatiations are easily translated into knowledge base queries. The same method is applicable to other object-oriented knowledge bases and other parsing techniques. The original contribution of this work is to show an approach with which natural language interfaces can with loweffort be adapted to work with any new knowledge base: While most other systems require a complete model of the domain for the natural language interface knowledge representation, we derive most of this information from the application's knowledge base. This technique reduces the amount of work needed to create the interface to about additional 15 percent after building the knowledge base for the application kernel.*

AI topic: knowledge acquisition, knowledge representation, natural language interface
Language/Tool: Unix, C++
Status: implementation complete, evaluation in progress
Effort: about 4 person years
Impact: quick development of restricted natural language interface for certain classes of knowledge-based applications (15 % additional knowledge acquisition work).

1 Introduction

There have been several attempts to create natural language interfaces for databases [15, 10, 5
of these natural language interfaces t
transportable, that is, they
domains by changing
dependent knowle
nal datab
e

express facts) and assertional knowledge (facts about *individuals (instances)* in the application domain).

The terminological knowledge consists of concept definitions and role definitions. A *concept* can be thought of as an abstract set of individuals. The concrete individuals that belong to a concept are called the *instances* of that concept. A *role* is a binary relation from a concept A to a concept B, i.e., a set of of instances. A is called the *domain* of t is called the *range* of the role. C defined with constructor of the set of all res and small, but suffice to construct all information the natural language interface needs. In this section we describe the type of information the annotations contain and how it is used in YAKR to generate the case frames. See [6, 14] for a detailed description.

There are two main types of information present in the annotations: (1) information about individual words
and (2) information about grammatical constru

The word information associates each
with its natural language sync
simple phrases that rep
variety of the
concep

Only the words Zugriff and Lesen need to be in the dictionary, the compound is algorithmically broken into these components.

4.2 Grammatical Construction Annotations

All the above annotations merely describe phrases that represent individual concepts; no case frames are built from them. The source of cases for the case frames are the roles. This is where the information about gra matical constructions is used, which tells to insert where. Similar Annotat to generate the informatio other natural lan

As an exampl an act

that has to be used in a query if the case containing it has been filled. Most of the instantiations can simply be processed in a top-down manner, only the instantiations for certain grammatical constructions require some more complicated processing. In any case, this processing is purely mechanical. No further semantic processing is necessary. Most ambiguities that remai
after parsing need not be explicitly resolv
since the wrong interpretations re
return no answer at all.

When a parsing techn
sing shall be u
gets

proach, the lexicon can be reduced to a dictionary (without semantic information); the semantic information can be derived via the annotations. The conceptual schema consists of sort information and constraints on the arguments of nonsort predicates. With our approach, no such schema is necessary at all; the information can completely be deduced from the knowledge base itself. The database schema consists of info tion that enables the mapping of the inte presentation to a query expresse language. With our appr mapping can be d derived f on

T

[9] Keith E. Gorlen, Sanford M Orlow, and Perry S. Plexico. *Data Abstraction and Object-Oriented Programming in C++*. Wley, Chichester, 1991.

[10] B. Grosz, D. Appelt, P. Martin, and F. Pereira. TEAM: An experiment in the design of transportable natural language interfaces. *Artificial Intelligence*, 32(1987):173–243, 1987.

[11] IEEE Computer Society. *The Seventh Conference on Artificial Intelligence Applications*, Mami Beach, Florida, February 1991. IEEE Computer Society Press.

[12] Matthias Ott. Modellierung von UNIX-Kommandos mit YAKR. Studienarbeit, Universität Karlsruhe, Institut für Programmstruktu
und Datenorganisation, D-7500 Karlsru
many, November 1992.

[13] Lutz Prechelt. Ein Fallscha
Deutsch. Master's th
strukturen u
Karls

[14]