Transportable Natural Language Interfaces for
Taxonomic Knowledge Representation System

Lutz Prechelt, Finn Dag Buo, 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

1 Introduction

There have been several attempts to create natural language interfaces for databases [15, 10, 5]. Of these natural language interfaces some are transportable, that is, they can 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 15 percent after building the knowledge base for the application kernel.

A1 topic: knowledge acquisition, knowledge representation, natural language interface
Language/Tools: 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).
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 instances. A is called the domain of the role. C is called the range of the role. C is defined with constructor of the set of all results.
and small, but suffice to construct all information the
natural language interface needs. In this section we
describe the type of information the annotations con-
tain and how it is used in YAR 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 syn-
se simple phrases that repr
variety of the
concept
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 grammatical constructions is used, which tells us to insert where. Similar Annotations to generate the information of other natural languages.

As an example, 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 remain after parsing need not be explicitly resolved, since the wrong interpretations resolve to return no answer at all.

When a parsing technique shall be used,
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 information that enables the mapping of the internal representation to a query expression language. With our approach, the mapping can be derived from the schema.


[14]