A query tool for syntactically annotated corpora

Laura Kallmeyer
SFB 441, University of Tübingen
lk@sfs.nphil.uni-tuebingen.de

In this talk I present a query tool for treebanks, more precisely for the German Verbmobil treebank (Stegmann et al. 1998). The query tool allows to search for terminals, node and edge labels and binary relations immediate dominance, dominance and linear precedence.

Architecture of the tool The tool consists of an initializing component, a query component and a relational database. First, the intializing component processes the trees of the corpus to prepare them for searching. It extracts pairs of nodes and stores them in the relational database. The query component takes the input queries, translates them into SQL queries and passes them to the database. The corpus search itself is done by the database management system, and the database management system also takes over the task of creating indexes on the tables in the database.
As database management system I have chosen mysql and the query tool itself is implemented in Java using JDBC (Java Database Connectivity) as application programming interface.

The initializing component The initializing component extracts nodes together with information on terminal strings, node and edge labels and dominance and linear precedence relations from annotation structures in the corpus. The way this is done depends on the data structure underlying the annotation. The Verbmobil annotation structures are not finite trees since two nodes are not necessarily related by a dominance or linear precedence relation. Dominance is computed as reflexive transitive closure of immediate dominance which is directly available from the corpus. Linear precedence is obtained from immediate precedence between leaves (terminal strings) as follows. First, the transitive closure is built. Then, for all nodes a, b such that neither of them dominates the other: If c precedes d for all c dominated by a and d dominated by b, then linear precedence between a and b is added.
Pairs of nodes are then stored in the database together with label, edge and dominance/linear precedence information.

The query component The query language is a quantifier free first order logic. Expressions are built using variables over nodes, unary predicates for terminal strings and node and edge labels and binary relations immediate dominance, dominance and linear precedence. From the input query, the query component first extracts pairs of variables occurring together in single conjuncts. Then, depending on these pairs, the query is translated into an SQL query that is passed to the database.

Current state of the work The work presented here is work in progress. Currently, the database has been designed and the initializing component has been implemented. The implementation of the query component is supposed to be finished in May 2000. Further work that needs to be done is the implementation of a web-based user-interface.

Conclusion I propose a query tool for treebanks that stores node pairs from a corpus in a relational database and then searches the corpus by executing SQL queries on the database. The advantage of such an approach is that the indexing functionalities offered by the database system directly affect the performance of searching the corpus.

References

Stegmann, R., Schulz, H. and Hinrichs, E. W.: 1998, Stylebook for the German Treebank in VERBMOBIL, Eberhard-Karls Universität Tübingen.