Automatic Functor Assignment in the Prague Dependency Treebank*

Zdeněk Žabokrtský
Czech Technical University, Department of Computer Science
121 35 Praha 2, Karlovo nám. 13, Czech Republic
zabokrzt@cs.felk.cvut.cz

Abstract. This paper presents work in progress, the goal of which is to develop a module for automatic transition from analytic tree structures to tectogrammatical tree structures within the Prague Dependency Treebank project. Several rule-based and dictionary-based methods were combined in order to be able to make maximal use of both information extractable from the training set and a priori knowledge. The implementation of this approach was verified on a testing set, and a detailed evaluation of the results achieved so far is presented.

1 Introduction

The process of syntactic tagging in the Prague Dependency Treebank (PDT) is divided into two steps. The first step results in analytic tree structures (ATS), in which every word form and punctuation mark is explicitly represented as a node of rooted tree, with no additional nodes added (except for the root of the tree of every sentence). The second step results in tectogrammatical tree structures (TGTS), which approximate the underlying sentence representations according to [4]. In contrast to the ATSs, only autosemantic words have nodes of their own in TGTSs, information about functional words (prepositions, subordinating conjunctions, etc.) are contained in the tags attached to the autosemantics nodes. Figure 1 depicts an example of a TGTS.

Apart from slight changes in the topology of the input ATS (for instance, pruning of synsemantic nodes), the transition from ATSs to TGTSs involves the assignment of the tectogrammatical function (functor) to every node in the tree. There are roughly 60 functors divided into two subgroups (cf. [4]): (i) actants (ACTor, PATient, ADDRessee, EFFect, ORIGin) and (ii) free modifiers: TWHEN (time-when), LOCaction, MEANS, EXTent, BENeficiary, ATTRibute . . .).

At present, the topological conversion and the assignment of a few functors (e.g., ACT, PAR, PRED) are solved automatically by the procedure from Böhmová et al. [4]. However, most of the functors have to be assigned manually. The amount of labour involved in the manual annotation obviously slows down the growth of the PDT on

---

* I would like to thank my advisor Ivana Kruijff-Korbayová for her permanent support. I am also indebted to Alevtina Bémová for consultations about functor assignment, to Petr Pajas for the help with the data preprocessing and to Julia Hockenmaier for her useful comments.
the tectogrammatical level. Decreasing the amount of manual annotation has been the motivation for developing the more complex *automatic functor assignment system* (AFA) presented in this paper. Let us describe the starting position.

- No general unambiguous rules for functor assignment are known, human annotators use mostly only their language experience and intuition. We cannot reach 100% correctness of AFA since even the results of individual annotators sometimes differ.
- The annotators usually use the whole sentence context for their decision. It has not been measured how often it is really unavoidable to take the full context into account or how large the context must be.
- Preliminary measurements revealed that the distribution of functors is very non-uniform. The 15 most frequent functors cover roughly 90% of nodes. Conversely, there are hardly any examples for the least frequent functors.
- It would be very time consuming to test the performance AFA on randomly selected ATSs and find errors manually. Fortunately we can use the ATSs for which manually created TGTSs are already available, annotate them automatically and compare the results against the manually annotated TGTSs.
- The available TGTSs contain imperfect data. Some errors are inherited from ATSs, and functor assignments are in some cases ambiguous (nodes with more than one functor) or incomplete (some nodes have no functor yet).

## 2 Materials

*Training and Testing Sets.* When I started working on AFA, 18 TGTS files were available, each containing up to 50 sentences from newspaper articles. This was a sufficient amount of data for knowledge mining, which can improve the AFA’s performance. However, in order to reliably measure AFA’s correctness, it is necessary to have a separate data