Features and Tags

Sven Naumann

Linguistische Datenverarbeitung, Universität Trier, D–54286 Trier

Summary: In many natural language processing systems (NLP-systems) the analysis of linguistic data starts with the identification of word classes (tagging). This paper describes the knowledge–based tagger FTAG which was developed as part of a system for the acquisition of syntactic knowledge from corpora.

1. Overview

Since machine-readable corpora became available in the early sixties quite a number of different tagging systems have been proposed. FTAG is a knowledge–based tagging system (tagger from now on) which was designed as a flexible tool for the linguistic analysis of medium–sized corpora*. The following two sections describe the tag set used in FTAG and the approach taken to make sure that each word form gets a unique tag. Section 4 gives a detailed impression of the system architecture and the tagging algorithm. Finally, first results and necessary improvements are discussed.

2. Tags

Unlike most other taggers FTAG uses complex feature structures instead of a fixed set of simple tags. Using feature structures as tags offers several advantages:

1. As feature structures have turned out to be the most popular data structure for presenting linguistic information in NLP–systems, a tagger which uses feature structures can be easily combined with those systems.

2. Feature structures can be extended dynamically: Information produced by different modules of an NLP–system can be combined quite naturally in a single feature structure; i.e. the morpho–syntactic information generated by the tagger can be completed by information which come from the syntax or the semantic module.

3. Applications may differ with respect to the granularity of (lexical) analysis required. Feature structures make it possible to change the size of the tag set quite easily: If all features (and their values) are considered as relevant for determining the identity of two tags, maximum granularity is achieved. To reduce the size of the tag set, it is sufficient to mark a certain subset of features as irrelevant in this respect.

*At the moment, there is only a german version of FTAG. But the modular structure of the system makes is easy to adapt it to different languages.
4. Feature structures allow a flexible treatment of ambiguity, which is one of the main problems in tagging: If \( C = \{C_1, \ldots, C_n\} \) \((n > 1)\) is a set of candidate tags for a word form \( w \), a unique tag for \( w \) can be formed either by simply disregarding all features whose value are not determined unambiguously (under specification) or by using logical connectives (such as disjunction and negation) to combine the conflicting information within one feature structure\(^1\).

FTAG uses typed feature structures which contain at least the following features:

\[
\begin{array}{c}
\text{PHON} & \text{word form} \\
\text{SYN} & \text{HEAD} \\
\text{LEX} & + \\
\end{array}
\begin{array}{c}
\text{MAJ} & \text{category} \\
\text{TYP} & \text{subclass} \\
\end{array}
\]

If \( w \) is a word form and \( f \) a feature structure generated by FTAG as tag for \( w \), then the PHON-feature in \( f \) takes \( w \) itself as value. MAJ(or) determines the word class (verb, noun, adjectiv, \ldots) of \( w \) and TYP the subclass \( w \) belongs to. The feature specification \(<\text{LEX} +>\) marks \( w \) as a lexical item. Depending on the values of MAJ and TYP further feature specifications may be added. Features either take simple objects called atoms (e.g. PHON, LEX, etc.) or feature structures (e.g. SYN) as values. To minimize computational demands, the use of disjunction and negation is restricted to atomic values in FTAG.

**Example (1)**

Definite articles in German typically show a high degree of ambiguity concerning gender, number and case. There are, for example, eight readings for *die*:

- feminine \{nominative\} singular
- feminine \{accusative\}
- masculine \{nominative\} plural
- neuter

Instead of generating eight candidate tags, FTAG proposes two tags:

\[
\begin{array}{c}
\text{PHON} & \text{die} \\
\text{SYN} & \text{HEAD} \\
\text{LEX} & + \\
\end{array}
\begin{array}{c}
\text{MAJ} & \text{article} \\
\text{TYP} & \text{definite} \\
\text{case} & \text{nom} \lor \text{acc} \\
\text{gen} & \text{fem} \\
\text{num} & \text{sg} \\
\end{array}
\]

\(^1\)In some cases it may be still necessary to assign \( w \) more than one tag.