Leopar: an Interaction Grammar Parser

Guy Perrier, Bruno Guillaume

LEOPAR: an Interaction Grammar Parser

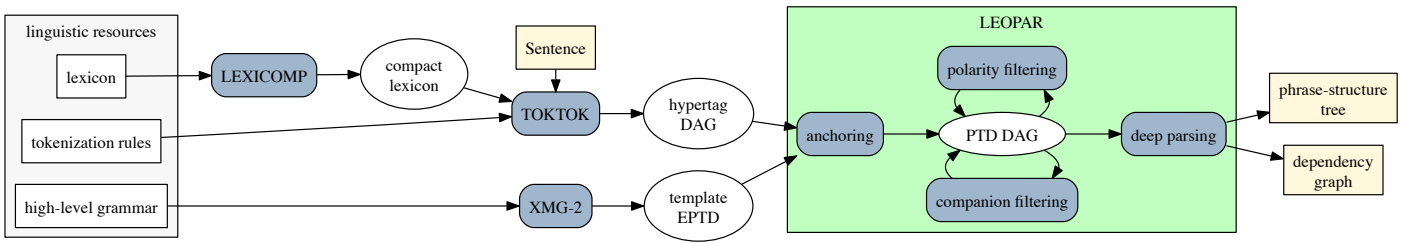
Guy Perrier and Bruno Guillaume
LORIA, Université de Lorraine, Nancy, France

1 Overview

LEOPAR[1] is a parser for natural languages which is based on the formalism of Interaction Grammars. The parsing principle (called "electrostatic parsing") consists in neutralizing opposite polarities: a positive polarity corresponds to an available linguistic feature and a negative one to an expected feature. The structures used in IG are underspecified syntactic trees decorated with polarities and called *Polarized Tree Descriptions* (PTDs)".

During the parsing process, PTDs are combined by partial superposition guided by the aim of neutralizing polarities: two opposite polarities are neutralized by merging their support nodes. Parsing succeeds if the process ends with a minimal and neutral tree.

The figure below describes LEOPAR and the toolchain around it (EPTD stands for *Elementary Polarized Tree Descriptions*).



2 Main modules

Anchoring: each lexicon entry is described with an *hypertag* (*i.e.* a feature structure which describes morpho-syntactic information on the lexical unit); in order to preserve tokenization ambiguities, tokenization is represented as a *Direct Acyclic Graph* (DAG) of hypertags; for each hypertag describing a lexical unit, the relevant EPTDs are build by instanciation of template EPTDs defined in the grammar.

Filtering: paths in the DAG produced by the anchoring represented the set of lexical selections that should be considered by the parsing process. In order to reduce the number of paths and so to speed up the deep parsing, the next step in LEOPAR is to filter out paths that are doomed to fail. Two kinds of filters are used: polarity filters remove lexical selections for which the set of polarities in not well-balanced and companion filters remove lexical selections for which it can be predicted (from knowledge on the template EPTDs) that some polarity will fail to find a dual polarity (called a companion) able to saturate it.

Deep parsing: the atomic operation used during deep parsing is the *node merging* operation. At each step, two dual polarities are chosen; the two nodes carrying these polarities are merged and tree description around the two are superposed. Of course, in case of dead-lock, backtracking is used to chose another pair of polarities.

Phrase-structure trees and dependency graphs: (E)PTD are tree descriptions which describes constraints on the phrase-structure tree. The parsing process aims to build a phrase structure tree which is a model the EPTDs chosen for each lexical unit. Dependency graphs are build from the phrase structure tree but also with information about taken from the parsing process itself.

[1] `http://leopar.loria.fr`