Lexical disambiguation with polarities and automata

Guillaume Bonfante, Joseph Le Roux and Guy Perrier
Universités de Nancy

Abstract

We propose a method for lexical disambiguation based on polarities for Interaction Grammars (IGs). We also show how to cope with coordination.

Introduction

We deal with lexical disambiguation using lexicalized IGs(Perrier, 2004). A grammar is defined by a lexicon which associates to every word of the language a set of lexical items specifying the grammatical behaviors the word. There is a complexity issue: the number of ways of associating each word of a sentence with a corresponding item is the product of the number of lexical entries of each word.

Using automata to represent this exponential is not a new idea. (Joshi and Srinivas, 1994) use probabilities to filter the most probable selections. But this method, based on n-grams, can miss valid selections and our concern is to keep the good ones.

In IGs lexical items have polarities like in Categorial Grammars (CGs) (Morrill, 1994). Lexical items may be seen as bags of polarized features and this simplification, as an abstraction. In the abstract grammar, parsing amounts to counting polarities and we use it to filter the initial grammar. The grounds of this method, from (Bonfante et al., 2004), lie in a homomorphism from the initial grammar to the abstract grammar: every parse in the former is transposed in a parse in the latter. The presented method can be applied to any formalism provided it can be polarized, see (Kahane, 2004).

1 Interaction Grammars

Here is a very simplified presentation of IGs.

1.1 Tree descriptions

IGs are a grammatical formalism using *underspecification* and *polarity*. Underspecification applies to syntactic trees, and is expressed using *tree descriptions*. A tree description represents trees sharing some properties. Polarized features decorating nodes express valences: positive (resp. negative) features represent available (resp. expected) resources.

Syntactic composition resembles an electrostatic process and consists in superposing tree descriptions while respecting polarities: a negative feature must encounter a dual positive feature to be neutralized. Parsing builds a completely specified tree, called the minimal neutral model.

1.2 Polarized features

Tree description nodes are associated with feature structures. A feature is a triple (f, p, v) such that:

- f is a feature name taken from a finite set $\mathcal{F}$;
- v is a finite disjunction $(v_1|\ldots|v_n)$ of atoms v_i;
- p is a polarity from $\{\rightarrow, \leftarrow, =\}$, whether it is positive, negative or neutral.

The labelling of a node N is written $N : (f, p, v)$.

2 Polarity automata

In this section, we define a criterion to filter valid descriptions. Then, we implement it with automata.

2.1 Global neutrality of valid descriptions

Definition 2.1 *Given a description D, $f \in \mathcal{F}$, a value v_1 and a node $N \in D$, p_N is defined as follows:*

1. *If $N : (f, \rightarrow, v_2)$ such that $v_2 \subseteq v_1$, then $p_N(f, v_2) = \{+1\}$.*
2. *If $N : (f, \rightarrow, v_2)$ such that $v_2 \cap v_1 \neq \emptyset$ and $v_2 \cap v_1 \neq v_1$, then $p_N(f, v_2) = \{0, +1\}$.*
3. *If $N : (, f\ \leftarrow, v_2)$ such that $v_2 \subseteq v_1$, then $p_N(f, v_2) = \{-1\}$.*
4. *If $N : (f, \leftarrow, v_2)$ such that $v_2 \cap v_1 \neq \emptyset$ and $v_2 \cap v_1 \neq v_1$, then $p_N(f, v_2) = \{0, -1\}$.*
5. *In all other cases, $p_N(f, v_2) = \{0\}$.*

Given a description D with nodes $N_1, \ldots, N_k$, a feature $f \in \mathcal{F}$ and a value v, the the function p_D is defined as follows: $p_D(f, v) = \{n \in \mathbb{Z} \mid n = n_1 + \cdots + n_k$ with $n_1 \in p_{N_1}(f, v), \ldots,\ n_k \in p_{N_k}(f, v)\}$.

Proposition 2.1 *The set of integers returned by the function p_D is always a $\mathbb{Z}$ interval.*

Now, we are able to give a *global neutrality criterion* (GNC) verified by all descriptions that have neutral models, the *saturated descriptions.*

Proposition 2.2 *If a description D is saturated then for every $f \in \mathcal{F}$ and value v , $0 \in p_D(f, v)$.*

Corollary 2.1 *If a description D is valid, for every f and v, $0 \in p_D(f, v)$.*

Descriptions that do not respect GNC can be discarded.

2.2 Deterministic polarity automata

Let $w_1 \dots w_n$ be sentence to parse with an IG G given by its lexicon Lex_G. For any $1 \leq i \leq n$, the word w_i is associated by Lex_G with a set of descriptions $Lex_G(w_i) = \{D_{i,1} \dots D_{i,k_i}\}$. A *lexical selection* is a sequence $D_{1,s_1} \dots D_{n,s_n}$, where $D_{i,s_i} \in Lex_G(w_i)$. For any $f \in \mathcal{F}$ and value v, the automaton $A(f, v)$ is defined as follows:

- States are pairs (i, p), where i represents the position between w_i and w_{i+1} and p is an interval of $\mathbb{Z}$ which represents the counting of polarities.
- Transitions have the form $(i, p) \xrightarrow{D_{i+1,s_k}} (i+1, q)$, where q is a $\mathbb{Z}$ interval of the sums of any element of p added to any element of $p_{D_{i+1,s_k}}(f, v)$.
- The initial state is $(0, \{0\})$ and accepting states are (n, p) such that $0 \in p$.

Every accepting path in $A(f, v)$ from the initial state represents a lexical selection that verifies GNC. Other paths can be deleted. So, it is necessary for a correct lexical selection to be recognized by all polarity automata, for every f and v.

Hence, the intersection of polarity automata contains the good solutions. Furthermore, if some (bad) lexical selection is not contained in all initial automata it will disappear from the intersection. Actually, this process pursues the filtering.

2.3 Selection of feature values for filtering

If a value appears neither with a positive nor negative polarity in any descriptions, the automaton will not filter. So, the first optimization is to consider only values with active (positive or negative) polarity within some descriptions. This set is called S_{pol}.

Then, the size of the automaton depends on the choice of the value: the non-determinism is coded in the intervals of the states. If $v \subseteq v'$ we know that $p_D(f, v) \supseteq p_D(f, v')$. Thus the automaton for v will be larger than the one for v'. As a consequence, we order feature values with the partial order $\subseteq$.

Let us pay attention to maximal values for that order in S_{pol}. There are two cases. First $v \cap v' = \emptyset$. Such a value does not remove non-determinism. Second, $v \cap v' \neq \emptyset$. In that case, $p_D(f, v \cup v')$ may be deterministic *even if* $p_D(f, v)$ and $p_D(f, v')$ are not: $A(f, v \cup v')$ is smaller than $A(f, v)$ and $A(f, v')$.

As a conclusion, we add to S_{pol} any feature value $v1 \cup v2$ such that $v_1, v_2 \in S_{pol}$ and $v_1 \cap v_2 \neq \emptyset$ until we reach a fix point.

3 Refinements

Coordination is a linguistic phenomenon that shows so much ambiguity that GNC is not sufficient but we can take advantage of the syntactic modelisation.

3.1 Syntactic modelisation of coordination

Our modelisation of coordination is inspired from CGs, see (Morrill, 1994) for details. To be conjoinable two segments must be on the left and on the right of a coordination and have the same active polarities (called Δ). The description associated with a coordination neutralizes the conjoins and offers to the rest of the sentence the same active polarities.

3.2 Coordination and polarities

We note δ the particular value of Δ for a feature f and a value v.

We first remark that to conjoin 2 segments $D_h, \dots, D_{i-1}$, and $D_{i+1}, \dots, D_j$, $\sum_{n=h}^{i-1} p_{D_n}(f, v) = \sum_{n=i+1}^{j} p_{D_n}(f, v) = \delta$. Moreover, if D_i is the description associated with a coordination, then $p_{D_i}(f, v) = -\delta$, and the global polarity of the whole segment is $\sum_{n=h}^{j} p_{D_n}(f, v) = \delta$.

Consequently, for any f, any v and a sentence with a coordination at position i we have the following invariants: $\sum_{n=1}^{i} p_{D_n}(f, v) = \sum_{n=1}^{h-1} p_{D_n}(f, v)$ and $\sum_{n=1}^{j} p_{D_n}(f, v) = \sum_{n=1}^{i-1} p_{D_n}(f, v)$.

3.3 Automata for coordination

We want to assert these invariants on automata. We notice that $\sum_{n=1}^{i} p_{D_n}(f, v)$ is the second projection of the state $(i+1, \mathcal{I})$ of $A(f, v)$. Hence our invariants can be applied on states. For every transition t labelled with a coordination from (i, p) to $(i+1, q)$ in $A(f, v)$ we check that: (1) there exists (h, q') in the path from the initial state to (i, p) with $q' = q$ and (2) there exists (k, p') in the path from $(i+1, q)$ to a final state with $p' = p$. If these states cannot be found, the transition t should be removed.

4 Conclusion

We presented a symbolic method for lexical selection. We used IGs but this method can be extended to other formalisms, see (Bonfante et al., 2004).

Our method is parameterable: possible choices are using feature values or maximal values over them, using coordination heuristic or not. This can lead to a fine grained approach of lexical selection where parameters are triggered by some words in the input or by the lengths of the sentences.

References

G. Bonfante, B. Guillaume, and G. Perrier. 2004. Polarization and abstraction of grammatical formalisms as methods for lexical disambiguation. In *20th Conference on Computational Linguistics, CoLing'2004, Genève, Switzerland*, pages 303–309.

A. Joshi and B. Srinivas. 1994. Disambiguation of super parts of speech (or supertags) : Almost parsing. In *COLING'94, Kyoto*, pages 154–160.

S. Kahane. 2004. Grammaires d'unification polarisées. In *TALN2004, Fès, Maroc*, pages 233–242.

Glyn Morrill. 1994. *Type Logical Grammar: Categorial Logic of Signs*. Kluwer, Dordrecht.

G. Perrier. 2004. La sémantique dans les grammaires d'interaction. *Traitement Automatique des Langues*, 45(3):123–144.