Alpino: A Wide Coverage Computational Grammar for Dutch

Gosse Bouma and Gertjan van Noord
Alfa-informatica, RUG

CLIN 00
Overview

- PIONIER project ‘Algorithms for Linguistic Processing’
- The Alpino grammar,
- Lexical resources,
- Construction of Dependency Trees,
- Treebank and evaluation,
- Future work.
Algorithms for Linguistic Processing

• Efficient processing and disambiguation of natural language,

• Develop wide-coverage Dutch grammar,

• Study disambiguation techniques,

• Evaluate coverage & disambiguation,

• (Efficiency & Finite-state approximations).
Grammar

- Lexicalized (HPSG-style) grammar,
- Extension of the NWO-TST (OVIS) grammar,
- Added rules for written language,
- Incorporated lexical entries based on Celex and Parole.
Rule Coverage

- Sentence types: declaratives, yes/no & WH-questions, topicalization, imperatives, subordinate clauses,

- NPs: relatives, sbar-complements, titles (minister zalm), measure phrases, temporal expressions, ...

- VP syntax: NP, PP, VP, SBAR complements, predicative phrases, adjuncts, verb clusters, particles, passives.

- Coordination of maximal projections (NP, PP, S, ...).
Inheritance in Rule Definitions

- 114 rules

- `pp --> p np is-a head-comps-struct`.

- `head-comps-struct` is-a `headed-struct`.

- `headed-struct` satisfies
  - `head-feature principle`,
  - `valence principle`,
  - `filler principle`. 
Example Rule

\[ \text{rule}(\text{pp}_p\_\text{np}, \begin{bmatrix} \text{wh} & 1 \\ \text{slash} & 2 \\ \text{ppost} & \text{no} \\ \text{dt} & 3 \end{bmatrix}, \begin{bmatrix} \text{sc} & \langle 4 \rangle \\ \text{wh} & 1 \\ \text{prep} & 2 \\ \text{ppost} & \text{no} \\ \text{dt} & 3 \end{bmatrix}, \begin{bmatrix} \text{nform} & \text{\neg er} \end{bmatrix}) \].
Inheritance for Lexical Entries

- ‘toerekenen’ is-a trans-particle-verb
- trans-particle-verb is-a trans-verb
- trans-verb is-a np-subj-verb
- np-subj-verb is-a verb
- verb is-a lexical-sign.
- lexical-sign satisfies argument-realization.
Recursive Constraints & Co-routining

- Slash-introduction defined as a constraint on mapping from `DEPENDENTS (and SUBJ)` to `SUBCAT and SLASH` (Bouma, Malouf, Sag, 2001).

- Verb-raising verbs defined using argument-inheritance (append of `SUBCAT-lists`) (Bouma and van Noord, 97),

- Co-routining is used for implementation of such constraints (van Noord and Bouma, 1994).
\[
\text{realize-args}(\langle 2 | 1 \rangle, 3, 4).
\]
Lexical Resources

- Wide-coverage of lexicalist grammars requires detailed lexical info,

- We use existing lexical resources (Celex & Parole) to obtain morphological and subcategorization info.

- Currently, the system has approx. 150K (inflected) lexical entries.
Lexical Resources

- **Celex:**
  - 33K lemma’s for nouns, adjectives, adverbs, etc.,
  - 5800 lemma’s for trans & intrans (particle) verbs.

- **Parole:**
  - 1600 verbs with subcat-frames not covered by Celex,
  - 800 nouns with special subcat properties.

- **“Hand”:**
  - 800 hand-crafted lemma’s,
  - 4K proper names occurring in Eindhoven corpus.
Treebank

• A syntactically annotated corpus is useful for:
  ★ Grammar Debugging,
  ★ Evaluation,
  ★ Collection of statistical info.

• Using current grammar directly has disadvantages:
  ★ Grammars change,
  ★ Annotation is difficult for strings outside coverage,
  ★ Hard to compare with other systems,
Dependency Trees

- Provide a grammar independent level of representation,
- Suitable for (relatively) free-word order languages,
- Lexical Dep Relations are useful for data-driven, statistical, parsing (Collins 98),
- We adopt annotation format for Dutch developed in CGN project.
Head-driven DT construction

- Data-structure: feature for each Dep Rel,
- A lexical head subcategorizes for a specific set of dependents, each linked to a specific Dep Rel,
- In head-comps-structures, Dep Tree can simply be shared between mother and head.
\[
\text{deps} \left\langle \begin{array}{c} \text{pred} \\
\text{subj} \\
\text{lex} \\
\text{dt} \\
v \\
vdom \\
\end{array} \right\rangle \otimes \begin{array}{c} \text{case} \\
\text{acc} \\
\text{hwrk} \\
\text{vind} \\
\text{postag} \\
\text{verb} \\
\text{cat} \\
\text{inf} \\
\text{su} \\
\text{obj1} \\
\text{predc} \\
\text{mod} \\
\end{array} \right\rangle
\]

\[
\left\langle \begin{array}{c} \text{pred} \\
\text{subj} \\
\text{lex} \\
\text{dt} \\
v \\
vdom \\
\end{array} \right\rangle \text{, viden }\)
Chevrolet brengt voor 1970 een nieuw model uit:
Phrase Structure and Dep Trees

• DT-construction in the grammar:
  ★ coordination (not a regular headed-struct),
  ★ unbounded dependencies (not lexically headed),
  ★ modification (no lexical treatment of adjuncts).

• Structure of Dep Tree not always isomorphic to syntactic tree.
  ★ Example: Crossing Dependency Constructions.
Building a Treebank

- Thistle: editor for linguistic objects (Calder, 2000),
- Define a Thistle SPEC (XML DTD) for Dep Trees,
- Initial trees constructed with Alpino,
  - Parse input string,
  - Select (manually) best parse,
  - Store corresponding Dep Tree as XML
- Use Thistle to edit and correct parse results,
Using the Treebank

- Grammar Evaluation based on Dep Rel’s between lexical Heads (Carroll et al, 1999),
- Dep Tree defines as set of $\langle$HdWrd DepRel DepHdWrd$\rangle$, e.g.

  $\langle$dat body wil $\rangle$
  $\langle$wil su ik $\rangle$
  $\langle$wil vc ontmoet $\rangle$
  $\langle$ontmoet su ik $\rangle$
  $\langle$ontmoet obj1 hem $\rangle$
Using the Treebank

- Parse results can be scored for precision and recall using lexically headed dependency relations,
- Useful during grammar development,
- Probabilities for lexical dependency relations can be estimated by parsing (unannotated) text,
- These can be used for disambiguation (i.e. to rank parse-results).
Conclusions

- Coverage: Combination of lexicalist HPSG-style grammar with existing lexical resources,
- Head-driven construction of Dependency Trees,
- Treebank construction,
- Grammar evaluation.
Future Work

• Expand syntactic coverage,

• Expand lexicon (use CGN lexical resources...).

• Expand treebank,

• Create parse selection tool for manual annotation,

• Build a statistical disambiguation model...