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

$$\texttt{rule}(\text{pp\_p\_np}, {}_{pp}\begin{bmatrix} \text{wh} & \boxed{1} \\ \text{slash} & \langle\ \rangle \\ \text{prep} & \boxed{2} \\ \text{ppost} & no \\ \text{dt} & \boxed{3} \end{bmatrix}, \langle {}_{p}\begin{bmatrix} \text{sc} & \langle \boxed{4} \rangle \\ \text{wh} & \boxed{1} \\ \text{prep} & \boxed{2} \\ \text{ppost} & no \\ \text{dt} & \boxed{3} \end{bmatrix}, \boxed{4}_{np}\begin{bmatrix} \text{nform} & \neg er \end{bmatrix} \rangle).$$

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).

$$\texttt{lex}\Bigg(\begin{bmatrix}
\text{deps} & \boxed{1}\langle \underset{np}{\begin{bmatrix} case & acc \\ nform & norm \end{bmatrix}} \rangle \\
\text{subj} & \boxed{2} \underset{np}{\begin{bmatrix} agr & sg\&thi \\ case & nom \\ nform & norm \end{bmatrix}} \\
\text{sc} & \boxed{3} \\
\text{parts} & \langle \underset{part}{\begin{bmatrix} part & toe \end{bmatrix}} \rangle \\
\text{vform} & fin \\
\text{slash} & \boxed{4} \\
\text{hebben_zijn} & hebben
\end{bmatrix}_{v}\Bigg) \text{ :- realize-args}(\langle \boxed{2} \mid \boxed{1} \rangle, \boxed{3}, \boxed{4}).$$

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

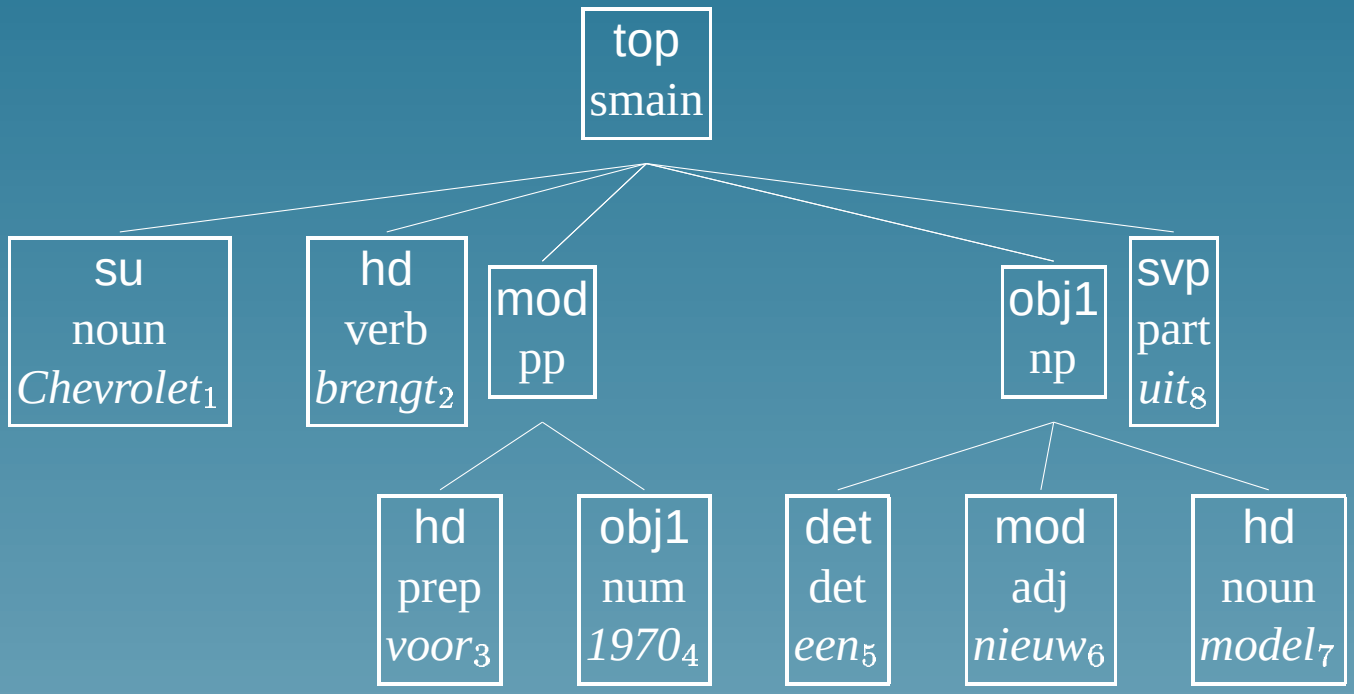
- Provide a grammar independent level of representation,
- Suitable for (relatively) free-word order languages,
- Lexical Dep Relations are useful for data-driven, statistical, parsing (Collins 98),
- We adopt annotation format for Dutch developed in CGN project.

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.

$$\texttt{lex}\Bigg(\begin{bmatrix}\textsf{deps} & \underset{pred}{\Big\langle}\begin{bmatrix} dt & \boxed{1}\end{bmatrix}, \underset{np}{\begin{bmatrix} case & acc \\ dt & \boxed{2}\end{bmatrix}}\Big\rangle \\ \textsf{subj} & \underset{np}{\begin{bmatrix} dt & \boxed{3}\end{bmatrix}} \\ \textsf{dt} & \underset{vdom}{\underset{v}{\begin{bmatrix} hwrd & vind \\ postag & verb \\ cat & inf \\ su & \boxed{3} \\ obj1 & \boxed{2} \\ predc & \boxed{1} \\ mod & \langle\rangle \end{bmatrix}}}\end{bmatrix}, \textsf{vinden}\ \Bigg).$$

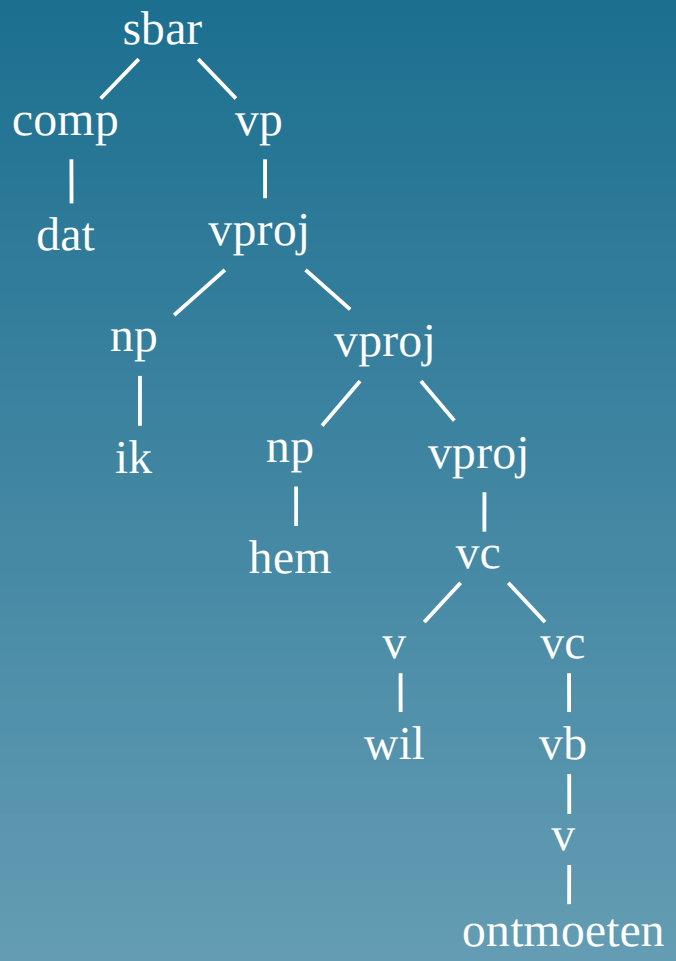
Dependency Tree

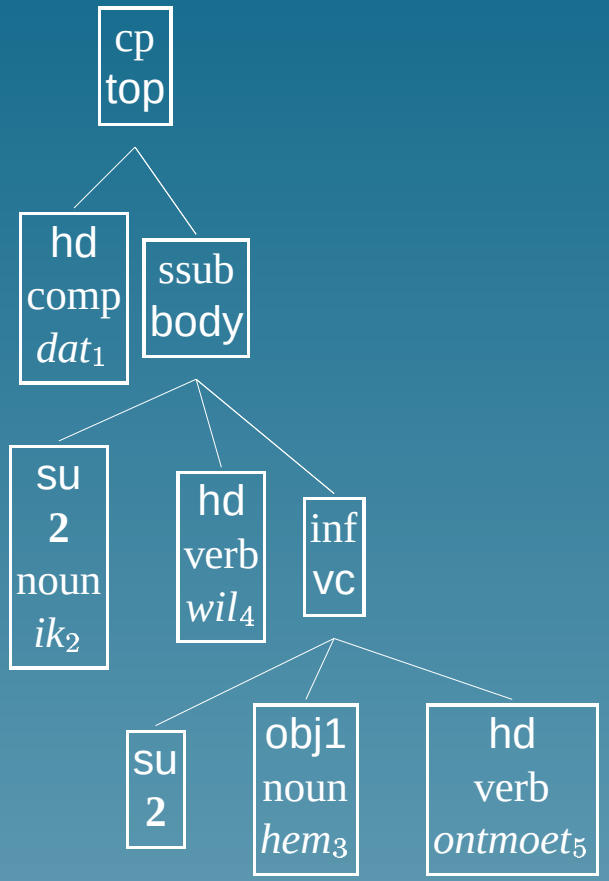


Chevrolet brengt voor 1970 een nieuw model uit :

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.


sbar
comp
vp
dat
vproj
np
vproj
ik
np
vproj
hem
vc
v
vc
wil
vb
v
ontmoeten


cp
top
hd
comp
dat1
ssub
body
su
2
noun
ik2
hd
verb
wil4
inf
vc
su
2
obj1
noun
hem3
hd
verb
ontmoet5

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 ⟨HdWrd DepRel DepHdWrd⟩, e.g.

⟨dat	body	wil ⟩
⟨wil	su	ik ⟩
⟨wil	vc	ontmoet⟩
⟨ontmoet	su	ik ⟩
⟨ontmoet	obj1	hem ⟩

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...