Alpino: A Wide Coverage Computational Grammar for Dutch

Gosse Bouma
Alfa-informatica, RUG

LOT Winterschool 2001
Overview

HPSG

- Rules,
- Lexicon.

The Alpino Grammar

- Rules,
- Using existing Lexical Resources,
- Extracting Subcategorization Frames.
Overview II

Parsing

- Unknown words, Robustness, Disambiguation.

Dependency Treebank

Conclusions and Future work
Algorithms for Linguistic Processing

- Efficient processing and disambiguation of natural language,
  - Develop wide-coverage Dutch grammar,
  - Evaluate coverage,
  - Evaluate disambiguation,
  - (Efficiency & Finite-state approximations).

Head-driven Phrase Structure Grammar

- Pollard & Sag (94), Sag & Wasow (99).
- **Signs:** Words and phrases are represented by bundles of linguistically motivated attribute-value pairs,
- **Lexicalist:** Words carry detailed information concerning their combinatoric properties,
- **Surface Oriented:** No movement or transformations,
- **Constraint-based:** A phrase is grammatical if it satisfies all constraints on this type of phrase.
Heads are important

• Information is shared between head-daughter and mother:
  – word class,
  – morphosyntactic info \((\text{CASE, VFORM, AGR})\),
  – (usually) semantics ,
  – (usually) \text{SLASH} \text{ (for unbounded dependencies)}

• Subcategorization properties of the head determines combinatoric properties.
dat Jan de kinderen aan het huiswerk *herinnert*

```
word

PHON herinnert

verb

HEAD VFORM fin

AGR sg3

SUBCAT ⟨PP[aan], NP[acc], NP[nom]⟩
```
Jan de kinderen aan het huiswerk herinnert
Verbal Head-Complement Rule

\[
\begin{align*}
&\text{HEAD} \quad \begin{cases} \text{1} \end{cases} \\
&\text{SUBCAT} \quad \begin{cases} \text{3} \end{cases} \\
&\text{DTRS} \\
&\text{HD-DTR} \\
&\text{NON-HD-DTRS} \quad \begin{cases} \text{2} \end{cases} \\
&\text{HEAD} \quad \begin{cases} \text{1} \end{cases} \quad \text{verb} \\
&\text{SUBCAT} \quad \begin{cases} \text{2} \mid \text{3} \end{cases} \\
&\begin{cases} \text{2} \end{cases}
\end{align*}
\]
Structures, Principles, and rules

• A rule of type (structure) $P$ must satisfy the constraints (principles) on $P$,

• Types form an inheritance hierarchy:
  – If $P'$ is a subtype of type $P$, it must satisfy the constraints on $P$. 
Headed Structures

• A Headed structure contains a head-daughter,

• Head-feature Principle: In headed structures, the HEAD features of the mother and head-daughter are shared (unified).

\[
hd\text{-phrase} \rightarrow \begin{bmatrix}
\text{HEAD} & \mathbb{1} \\
\text{DTRS} & \begin{bmatrix}
\text{HD-DTR} & \begin{bmatrix}
\text{HEAD} & \mathbb{1}
\end{bmatrix}
\end{bmatrix}
\end{bmatrix}
\]
Valence Structures

- A Valence structure is a headed structure,

- **Valence Principle**: The non-hd-dtrs list concatenated with the \textsc{subcat}-value of the mother equals the \textsc{subcat}-value of the hd-dtr.

\[
\begin{bmatrix}
\text{SUBCAT} & 1 \\
\text{DTRS} & \begin{bmatrix}
\text{HD-DTR} & \begin{bmatrix}
\text{SUBCAT} & 2 \\
\text{NON-HD-DTRS} & 3
\end{bmatrix}
\end{bmatrix}
\end{bmatrix} \leftarrow \text{concat}(3,1,2)
\]
The verbal head-complement rule is a valence structure,
More Structures

- A Head-Filler structure is a headed structure consisting of a head and a filler (used for WH-movement, relative clauses). The \textsc{slash}-value of the head and the filler unify.

- A head-modifier structure is a headed structure consisting of a head and a modifier. The \textsc{mod}-value of the modifier unifies with the head. The semantics of the mother is determined by the modifier.
The Alpino Grammar

• HPSG-style Grammar,
• Lots (100+) of rather specific rules,
• Lexicon (150K+ entries) based on CGN/Celex and Parole,
• Unbounded Dependencies, Verb Clusters, & Recursive Constraints.
Grammar Rules

- Rules have a Context-Free Backbone,
- Rules inherit constraints from Structures,
- `rule(RuleName, Mother, ListOfDaughters)`
**PP → P NP**

\[
\text{rule}(pp\_p\_np, \text{PP}, \text{[ P, NP ] }): - \\
\quad \text{PP} \Rightarrow pp, \text{P} \Rightarrow p, \text{NP} \Rightarrow np, \\
\quad \text{P:ppost} \Rightarrow \text{no}, \\
\quad \text{NP:nform} \Rightarrow \neg \text{er}, \\
\quad \text{hd\_comp\_struct}(\text{P}, \text{NP}, \text{PP}).
\]

\[
\text{rule}(pp\_p\_np, \text{PP}, \text{[ P, NP ] }): - \\
\quad \text{PP} \Rightarrow pp, \text{P} \Rightarrow p, \text{NP} \Rightarrow np, \\
\quad \text{P:ppost} \Rightarrow \text{no}, \\
\quad \text{NP:nform} \Rightarrow \neg \text{er}, \\
\quad \text{hd\_comp\_struct}(\text{P}, \text{NP}, \text{PP}).
\]
Other PP rules

%% pp --> p n, where p in \{ter,per\}
grammar_rule(pp_p_n, PP, [ P, N ] ) :-
    PP => pp, P => p, N => n,
    hd_comp_struct(P,N,PP).

%% pp --> p np part (naar Assen toe)
grammar_rule(pp_p_np_part, PP, [ P, NP, Post ] ) :-
    PP => pp, P => p, NP => np,
    NP:nform => ~er,
    hd_comp_struct(P,NP,PP),
    Post => part,
NP Syntax (17 rules)

np → det n       het kind
np → n           water, kinderen
n → n n          premier Kok
n → n sbar       feit dat ...

n → n te_vp      kans (om) te winnen ...

n → adj n        aardig kind
n → n pp         kind van de rekening
n → n rel        kind dat slaapt
n → n adj        iets lekkers
n → adj          (de) groene
np → np adv      de baas zelf
np → np , np ,   Jan, de baas,
...    ...
VP Syntax (21 rules)

\[
\begin{align*}
vproj & \rightarrow \ v_{\text{arg}} \ vc \quad (\text{dat) hij een boek wil lezen} \\
vproj & \rightarrow \ vc \quad (\text{Jan zou) } [\epsilon \text{ willen slapen}] \\
v c & \rightarrow \ v \ vc \quad (\text{dat hij) zou willen bellen} \\
v c & \rightarrow \ \text{part} \ vc \quad (\text{dat hij) op zou bellen} \\
v c & \rightarrow \ vb \quad (\text{dat het kind) slaapt} \\
v b & \rightarrow \ v \quad (\text{dat het kind) slaapt} \\
v & \rightarrow \ \epsilon \quad (\text{het kind wil een boek) } \epsilon \ (\text{lezen}) \\
\ldots & \ldots \\
\end{align*}
\]
Rule Coverage

- **Sentence types:** declaratives, yes/no & \( \text{wh} \)-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, ...).
The Lexicon

• HPSG is driven by the lexicon,
  – Especially subcategorization requirements,

• Lexical entries are defined using Inheritance,

• Entries extracted from Celex and Parole,
  – Including subcategorization frames.
Inheritance for Lexical Entries

- ‘aanrekenen’ 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.
Example Lexical Entry

PHON  rekent

DEPS  \[1\left\langle \begin{bmatrix} \text{CASE} & \text{acc} \\ \text{NFORM} & \text{norm} \end{bmatrix}, \begin{bmatrix} \text{AGR} & \text{sg} \\ \text{CASE} & \text{nom} \\ \text{NFORM} & \text{norm} \end{bmatrix} \right\rangle\]

SUBCAT  2

PARTS  \left\langle \begin{bmatrix} \text{PART} & \text{aan} \end{bmatrix} \right\rangle

VFORM  fin

SLASH  3

PERFECT  hebben

&  \text{argument realization}(1,2,3)
Lexical Types

- **Verbs (30+ types):**
  - intrans, trans, bi-trans, pp-complement, inherently reflexive, \textit{(te,om-)} vp-complement, \textit{(sbar,wh-)}s-complement, sbar-subject, het-subject, modal, aux, ...

- **Nouns:**
  - count, mass, proper name, pronoun (refl, relative, er, het), title, temporal, s-complement, vp-complement
Lexical Coverage

- **Alpino Lexicon**: 150K+ inflected entries,

- From **Celex**: 33K lemma’s for nouns, adjectives, adverbs, etc.

- From **Parole**: 800 nouns with special subcategorization frames,

- **Hand-coded**: 800 lemma’s (auxiliaries, modals, complementizers, determiners, pronouns, ...)

- **Celex & Parole**: 11,700 verbal lexical entries with subcategorization
## Dependency Frames

<table>
<thead>
<tr>
<th>Dependency Frame</th>
<th>Overlap</th>
<th>Clx only</th>
<th>Prl only</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>[SU:NP][OBJ1:NP]</td>
<td>1810</td>
<td>1211</td>
<td>240</td>
<td>3261</td>
</tr>
<tr>
<td>[SU:NP]</td>
<td>257</td>
<td>1697</td>
<td>42</td>
<td>1996</td>
</tr>
<tr>
<td>[SU:NP][PC:PP]</td>
<td>337</td>
<td>541</td>
<td>273</td>
<td>1151</td>
</tr>
<tr>
<td>[SU:NP][OBJ1:NP][PC:PP]</td>
<td>129</td>
<td>375</td>
<td>308</td>
<td>812</td>
</tr>
<tr>
<td>[SU:NP][VC:S]</td>
<td>103</td>
<td>136</td>
<td>103</td>
<td>342</td>
</tr>
<tr>
<td>[SUP:NP][OBJ1:NP][SU:CP]</td>
<td>7</td>
<td>247</td>
<td>5</td>
<td>259</td>
</tr>
<tr>
<td>[SU:NP][OBJ2:NP][OBJ1:NP]</td>
<td>65</td>
<td>171</td>
<td>28</td>
<td>264</td>
</tr>
<tr>
<td>[SU:NP][SE:NP][PC:PP]</td>
<td>65</td>
<td>62</td>
<td>102</td>
<td>229</td>
</tr>
<tr>
<td>[SU:NP][SE:NP]</td>
<td>49</td>
<td>137</td>
<td>65</td>
<td>251</td>
</tr>
<tr>
<td>[SU:NP][VC:VP]</td>
<td>10</td>
<td>16</td>
<td>37</td>
<td>63</td>
</tr>
</tbody>
</table>
Subcategorization Coverage

- Corpus of 400 short-sentences from Eindhoven-corpus,
- Syntactically annotated with (CGN-style) dependency trees.
- Of the verbs in the treebank (480) 80% is included in the Alpino lexicon with the appropriate dependency (subcategorization) frame.
Parsing and Robustness

- Parser
- Lexical Analysis and Unknown Words
- Robustness
- Parse Selection and Disambiguation
• Left-corner Chart Parser,

• Packed representation of the chart,

• Delayed evaluation for recursive constraints.

• Speed (first/best parse only): 0.5-1 sec for short sentences, 8-9 for newspaper text, 1 Month of Volkskrant (approx. 1.5 mln words) takes a week....
Lexical Analysis and Unknown Words

- **Filtering**: Add chart-item for a verb selecting a PP\(\langle prep \rangle\) only if \(prep\) is in the input string as well,
- Similarly for verbs with a separable particle (\(bel\ hem\ op\)).
- Try to **guess** category of Unknown Words:
  - Capitalized at begin of sentence,
  - Proper Names (\(Jan\ de\ Vries\)),
  - Try to find a known suffix (\(ongevallenverzekering\)),
  - Adjectival Proper Names (\(Amerikaanse\)),
  - Assume it is a Noun.
  - To do: Use POS-tagger for lexical analysis.
Robustness

• (If no full parse is possible),
  – try to find the minimal number of maximal projections in the chart,
  – covering the maximal number of words in the input.
Parse-Selection

• Full parses are preferred over partial parses (robustness),

• Some constructions and lexical entries have a penalty,
  – Non-subject-topic, vp-van-pp, deverbalized-noun, *uit*, *in* as verb, *een* as num,....
  – Parses with lowest penalty are preferred,
  – Hand-coded....

• *(Preferences based on statistical score for dependencies between head words).*
The Alpino Treebank

• Wide-coverage grammar development requires:
  – Debugging aids,
  – Some evaluation metric,
  – (Statistical component)

• Treebank:
  – Syntactically annotated corpus,
  – Signals parse-failures,
  – Can be used for evaluation,
  – Can be used to collect and evaluate statistical info
Dependency Trees

• Provide a grammar independent level of representation,

• Suitable for (relatively) free-word order languages,

• Lexical Dep Relations are useful for disambiguation (Collins 98),

• We adopt annotation format for Dutch developed in CGN project.
Head-driven DT construction

- Add attribute for each Dependency Relation,
- A lexical head subcategorizes for a specific set of dependents, each linked to a specific Dep Rel,
- In head-comp-structures, Dep Tree is shared between mother and head.
Dependencies in Lexical Entries

Wij *vinden* haar fink.

\[
\text{PHON} \quad \text{vinden}
\]

\[
\text{DEPS} \quad \langle \left[ \text{DT} \quad \text{1} \right], \left[ \text{CASE} \quad \text{acc} \right], \left[ \text{CASE} \quad \text{nom} \right] \rangle
\]

\[
\text{DT} \quad \left[ \begin{array}{c}
\text{HWRD} \quad \text{vind} \\
\text{POSTAG} \quad \text{verb} \\
\text{CAT} \quad \text{inf} \\
\text{SU} \quad \text{3} \\
\text{OBJ1} \quad \text{2} \\
\text{PRED1} \quad \text{1}
\end{array} \right]
\]

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:
  – Crossing Dependency Constructions.
Ik wil hem ontmoeten.
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 using a chart-based selection tool,
  - Store corresponding Dep Tree as XML

- Use Thistle to *edit and correct* parse results,
Current Contents

- 130 examples from CGN documentation,
- 500 short sentences from Eindhoven corpus,
- 40 longer sentences from Eindhoven corpus,
- 18 CGN examples (for LOT evaluation).
- approx 4000 words total....
Using the Treebank

- **Grammar Evaluation** based on Dep Rel’s between lexical Heads (Carroll et al, 1999),

- **Dep Tree** defines as set of \( \langle \text{HdWrd DepRel DepHdWrd} \rangle \), e.g.

\[
\langle \text{dat body wil} \rangle \\
\langle \text{wil su ik} \rangle \\
\langle \text{wil vc ontmoet} \rangle \\
\langle \text{ontmoet su ik} \rangle \\
\langle \text{ontmoet obj1 hem} \rangle
\]

- Parse results can be scored for precision and recall.
Conclusions and Future Work

So far:

- HPSG-style grammar with detailed rule-base,
- Detailed lexical information from existing resources,
- Robust Parsing and Disambiguation,
- Dependency Treebank for evaluation.

To do:

- Extend Rule-base,
- Corpus-based acquisition of subcategorization frames,
- Part-of-Speech tagger for lexical analysis,
- Expand Treebank,
- Statistical models for disambiguation.