Robust Parsing, Error Mining, Automated Lexical Acquisition, and Evaluation

Gertjan van Noord
University of Groningen
Robust Parsing

- Different topic: Robustness of Software
- Focus on Parsing

*a smooth degradation in the performance of a system when faced with unexpected input*
Approach in Alpino

- Produce something *useful* for unexpected input
  - unknown words
  - partial results from parser

- Expect a lot
  - Error Mining
  - Automated Lexical Acquisition
Overview

- Background: Alpino parser for Dutch
- Partial parse results
- Error Mining
- Automated Lexical Acquisition
- Evaluation of Robust Parsing
Background: Alpino

- Origin: rule-based parser in OVIS
- Grammar and lexicon
- Lexical analysis
- Syntactic analysis
- Parse Selection
- Dependency Structures; Treebanks; Evaluation
• Spoken Dialogue System for Timetable Information

• Robustness very important

• Comparison with DOP (Scha and Bod)

<table>
<thead>
<tr>
<th></th>
<th>WA</th>
<th>CA</th>
<th>cpu</th>
</tr>
</thead>
<tbody>
<tr>
<td>DOP</td>
<td>76.8</td>
<td>75.2</td>
<td>7011</td>
</tr>
<tr>
<td>rule-based</td>
<td>84.1</td>
<td>83.0</td>
<td>5524</td>
</tr>
</tbody>
</table>
The Alpino Grammar

- Grammar containing $\geq 600$ rules
- Lexicon ($\geq 100K$ words) based on Celex and Parole
- Grammar and lexicon organized as inheritance hierarchies
- output: CGN dependency structures
Grammar

- Linguist: capture generalizations; state general constraints only once
- Parser: specific rules; state as much constraining information as possible
- Constructionalist HPSG
Grammar Rules

- Grammar rules are instantiations of various *structures*
- ⋆ hd-compl-struct
  ⋆ hd-det-struct
  ⋆ hd-mod-struct
  ⋆ hd-filler-struct
  ⋆ hd-extra-struct
  ⋆ . . .
- *and* include very specific information
Grammar Structures

- Grammar structures are organized in an inheritance network
- Structures are associated with various principles
  - head-feature-principle
  - valence-principle
  - filler-principle
  - extraposition-principle
  - . . .
The Alpino Lexicon

- Lexical information is crucial
  - subcategorization frames

- The lexicon is a mapping from words to tags
  - Compact representation (*perfect hashing* FSA)

- Each *tag* is mapped to (one or more) signs

- This mapping is organised in inheritance network

- Tags combine lexical information and inflection
Lexical Analysis

- Lookup each word in (full-form) lexicon
- Treat unknown words
- Filter irrelevant tags
  - Cooccurrence restrictions filter impossible tags
  - HMM-tagger filters unlikely tags
Example

Mercedes zou haar nieuwe model gisteren hebben aangekondigd
Mercedes would have announced its new model yesterday

- 132 tags (including 2 for unknown word Mercedes)
- 22 tags survive rules
- 18 tags survive HMM-tagger
- 18 tags $\rightarrow$ 72 signs (vs. 1025 signs)
Parser

- Left-corner Parser with Memoization and Goal Weakening
- Delayed evaluation for recursive constraints
- Parse Forest: compact representation of all parses
Extract Best Parse

- Some function which scores parse trees
  - log linear model
  - trained on Alpino treebank (150,000 words)

- Naive solution: too inefficient

- Extend the function to partial parse trees

- Best-first search, with beam
Dependency Structures

- Provide a *grammar independent* level of representation
- Suitable for free-word order languages
- Lexical dependency relations are useful
- Annotation format from CGN project (Corpus of Spoken Dutch)
Dependency Relations

\[
\langle \text{zie} \quad \text{su} \quad \text{Cathy} \rangle
\]
\[
\langle \text{zie} \quad \text{obj}1 \quad \text{hen} \rangle
\]
\[
\langle \text{zie} \quad \text{vc} \quad \text{zwaai} \rangle
\]
\[
\langle \text{zwaai} \quad \text{mod} \quad \text{wild} \rangle
\]
\[
\langle \text{zwaai} \quad \text{su} \quad \text{hen} \rangle
\]
Evaluation Metrics

- $D_s$: number of dependencies found by system
- $D_g$: number of dependencies according to treebank
- $D_e$: number of wrong/missing dependencies

$$CA = 1 - \frac{D_e}{\max(D_g, D_s)}$$
## Current Status

<table>
<thead>
<tr>
<th>corpus</th>
<th>zinnen</th>
<th>lengte</th>
<th>CA%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Eindhoven corpus (cdbl)</td>
<td>7136</td>
<td>20</td>
<td>88.2</td>
</tr>
<tr>
<td>Trouw 2001</td>
<td>1400</td>
<td>17</td>
<td>90.8</td>
</tr>
<tr>
<td>CLEF questions (tuned for questions)</td>
<td>1345</td>
<td>11</td>
<td>96.2</td>
</tr>
</tbody>
</table>
Robust Parsing

- Lots of heuristics for unknown words and word sequences
- Heuristic to use partial parse results
Unknown word heuristics

- Pita, Peter Jan van Warmerdam
- karma, ancien régime, body mass index
- HELP, usa
- Italie, zó
- boterwetgeving, boter-wetgeving
More unknown word heuristics

- op- en terugbellen, land- en tuinbouw
- regering-Kok, Donald Duck-verhaal, science fiction-schrijver
- nummer 1-hit, artikel 12-status, oer-rock & roll
- wachtenden, verwijze
- animositeit, abusievelijk
- ...


If no full parse is available

- Top category: maximal projection (NP, VP, S, PP, AP . . . )
- Often: not a full parse
  - fragmentary input, ungrammatical input, . . .
  - omissions in the grammar, dictionary, . . .
Partial parse results

- Parser finds all instances of top category *anywhere* in input
- Find best sequence of non-overlapping parses
Partial parse results

- Parser finds all instances of top category *anywhere* in input
- Find best sequence of non-overlapping parses

Soms zes plastic bekers tegelijk, in een kartonnen dragertje

*Sometimes six plastic cups at the same time, in a cardboard retainer*

[ Soms zes plastic bekers ] [ tegelijk ] [ , ] [ in een kartonnen dragertje ]
Is this useful?

- It depends . . .
- Often: yes

- ✴ partial parse is correct (fragmentary input)
  - ✴ OVIS: important to recognize temporals and locatives
  - ✴ information extraction does not need full parses
  - . . .
Examples

- Fantastisch dus. *Fantastic, therefore.*
- Iedereen toch tevreden. *Everybody happy nonetheless.*
- Tijd is schaars. iedereen heeft haast. *Time is scarce. everybody is in a hurry.*
- 14 Hoe lang duurde de oorlog tussen Irak en Iran? *14 How long took the war between Iraq and Iran?*
- SKOPJE Een buitenwijk van de Macedonische hoofdstad Skopje wordt onder de voet gelopen door miljoenen miljoenpoten. *SKOPJE Part of the Macedonian capital Skopje is being run over by millions of . . .*
Examples

- SKOPJE Een buitenwijk van [de Macedonische hoofdstad Skopje] wordt onder de voet gelopen door miljoenen miljoenpoten.

- Raymann is laat Tante Esselien ontvangt [Boris Dittrich, fractievoorzitter van D66].

- [Voetballer Alexi Lalas] wordt genoemd (te veel aan testosteron), alsmede [tennisster Mary Joe Fernandez].

- Deelnemers onder anderen [burgemeester Meijer van Zwolle], projectontwikkelaar Peter Ruig rok ...

- [CNV-voorzitter Doekle Terpstra] spreekt van het 'meest
Learn from your mistakes
Next: Error Mining

Learn from your mistakes

Extend your expectations
Goal: improve coverage of the parser

- sets of hand-crafted examples $\implies$ problems must be anticipated
- treebanks $\implies$ much too small
- Instead: use unannotated material
Goal: improve coverage of the parser

- Run the parser on many sentences
- Analyse sentences with *missing* parses
- Find words and word sequences that occur (much) more often in these sentences
Corpora

* Twente Newspaper Corpus TwNC
* Various newspapers 1994-2004 (Trouw, NRC, AD, Volskrant, Parool)
* Sentences up to 20 (25, 30) words (with time-out)
* About 500 million words
Metric (1)

- full parse: a parse spanning the whole sentence
- $C(w)$: frequency of word $w$
- $C(w | \text{OK})$: frequency of word $w$ in sentences with a full parse
- compute for all words $w$:

\[
R(w) = \frac{C(w | \text{OK})}{C(w)}
\]
Coverage

- For this material: 91–96%
- An R-value significantly below .91 is interesting

0.000 7 l'd
0.000 9 aangroei
0.000 9 aanzoek
0.000 7 adoreert
0.000 8 afkeur
0.000 21 afroep
0.000 7 après
0.000 7 berge
0.000 7 einmal
Metric (2)

- Often, words are problematic only in certain contexts
- \( C(w_i \ldots w_j) \): frequency
- \( C(w_i \ldots w_j | OK) \): frequency in full parse
- \( R(w_i \ldots w_j) = \frac{C(w_i \ldots w_j | OK)}{C(w_i \ldots w_j)} \)

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.902</td>
<td>716</td>
<td>via</td>
</tr>
<tr>
<td></td>
<td>0.167</td>
<td>15</td>
<td>via via</td>
</tr>
<tr>
<td></td>
<td>0.916</td>
<td>165</td>
<td>waard</td>
</tr>
<tr>
<td></td>
<td>0.000</td>
<td>10</td>
<td>de waard</td>
</tr>
</tbody>
</table>
Metric (3)

- **Threshold**: only consider sequences where
  \[ C(w_i \ldots w_j) - C(w_i \ldots w_j|\text{OK}) > 5 \]

- **Consider longer sequences only if worse than corresponding shorter ones**:
  
  \[ R(w_h w_i \ldots w_j w_k) < R(w_h w_i \ldots w_j) \text{ and } \]
  
  \[ R(w_h w_i \ldots w_j w_k) < R(w_i \ldots w_j w_k) \]

- **if the score is the same, the longer Ngram is typically more informative**
Sort according to R value (1)

0.000 82 ! enz. chess
0.000 8 ! gevolgd chess
0.000 7 , zo 12-17u announcement
0.000 15 , zo 13-17u
0.000 316 - fl. new books
0.000 12 ; 127 blz.
0.000 10 ; 142 blz.
0.000 14 ; 143 blz.
0.000 19 16x27 checkers
0.000 7 2Klaver pas bridge
0.000 8 4 t/m 12 jaar announcement (theater, ..)
0.000 17 I have foreign language
Sort according to R value (2)

0.000 7 de huisraad
0.000 7 Maar eerlijk is eerlijk
0.000 9 en noem maar
0.000 18 is daar een voorbeeld
0.000 7 par excellence
0.000 7 In vroeger tijden
0.000 7 dan ten hele
0.000 7 hele gedwaald
0.000 7 het libido
0.000 9 kinds af
0.000 8 tenzij .  

unless .
List problematic examples

@ Vroeger was het nee, tenzij.
@ Nou ja, het is een Nee, tenzij . . .
@ Vlandse wetgever staat een ’nee, tenzij.
@ Orgaandonatie tenzij . . . . ik de nagel
@ Officeel is het : ja, tenzij.
@ Anderen : nee, tenzij.
@ Gebied tussen ja, mits en nee, tenzij.
@ Geen jacht tenzij.
@ U zult niet doden , tenzij.
Metric (4)

- \( R(w) = \frac{C(w|\text{OK})}{C(w)} \)

- But \( \frac{1}{4} \) less interesting, reliable than \( \frac{10}{40} \)

- Construct binomial confidence interval of \( \frac{C(w|\text{OK})}{C(w)} \)

- Use maximum value of this interval for sorting

- Slower

- Better results
Spots variety of errors

- tokenization
- mistakes in lexicon
- incomplete lexical descriptions
- frozen expressions (with idiosyncratic syntax)
- incomplete grammatical descriptions
- . . .
Tokenization

0.001 1884 @ . @ .
0.009 385 @ ! @ !
0.154 22 ’s advocaat ’s lawyer
0.482 8 H. ’s H. ’s
0.036 98 @ , roept @ , yells
0.011 469 @ , vraagt @ , asks
<table>
<thead>
<tr>
<th>Score</th>
<th>Position</th>
<th>Dutch Word</th>
<th>English Translation</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.369</td>
<td>8</td>
<td>de artikel</td>
<td>the article</td>
</tr>
<tr>
<td>0.551</td>
<td>11</td>
<td>de heimwee</td>
<td>the homesickness</td>
</tr>
<tr>
<td>0.437</td>
<td>18</td>
<td>de kaft</td>
<td>the cover</td>
</tr>
<tr>
<td>0.445</td>
<td>9</td>
<td>het folie</td>
<td>the foil</td>
</tr>
<tr>
<td>0.652</td>
<td>7</td>
<td>heeft opgetreden</td>
<td>has performed</td>
</tr>
<tr>
<td>Score</td>
<td>Word(s)</td>
<td>Translation</td>
<td></td>
</tr>
<tr>
<td>-------</td>
<td>------------</td>
<td>-------------</td>
<td></td>
</tr>
<tr>
<td>0.205</td>
<td>dupe van</td>
<td>victim of</td>
<td></td>
</tr>
<tr>
<td>0.247</td>
<td>het Turks</td>
<td>the Turkish</td>
<td></td>
</tr>
<tr>
<td>0.284</td>
<td>begunstigden</td>
<td>favoured (N/V)</td>
<td></td>
</tr>
<tr>
<td>0.360</td>
<td>aan te klikken</td>
<td>on to click</td>
<td></td>
</tr>
<tr>
<td>0.360</td>
<td>doodzonde dat</td>
<td>mortal sin that</td>
<td></td>
</tr>
<tr>
<td>0.454</td>
<td>zwarts</td>
<td>black’s</td>
<td></td>
</tr>
<tr>
<td>0.538</td>
<td>zich eraan dat</td>
<td>self there-on that</td>
<td></td>
</tr>
<tr>
<td>Score</td>
<td>Line</td>
<td>Dutch Expression</td>
<td>English Translation</td>
</tr>
<tr>
<td>-------</td>
<td>------</td>
<td>------------------</td>
<td>---------------------</td>
</tr>
<tr>
<td>0.149</td>
<td>34</td>
<td>Het zij</td>
<td><em>It be[l]</em></td>
</tr>
<tr>
<td>0.231</td>
<td>14</td>
<td>te weeg</td>
<td></td>
</tr>
<tr>
<td>0.247</td>
<td>13</td>
<td>dan schaadt het</td>
<td><em>then harms it</em></td>
</tr>
<tr>
<td>0.308</td>
<td>10</td>
<td>te berge</td>
<td><em>to mountain[l]</em></td>
</tr>
<tr>
<td>0.308</td>
<td>10</td>
<td>hele gedwaald</td>
<td><em>whole[l] dwelled</em></td>
</tr>
<tr>
<td>0.399</td>
<td>25</td>
<td>God zij</td>
<td><em>God be[l]</em></td>
</tr>
<tr>
<td>0.677</td>
<td>12</td>
<td>goedun huize</td>
<td><em>good house[l]</em></td>
</tr>
</tbody>
</table>
Incomplete Grammatical descriptions

- Wij Nederlanders: We Dutch
- Geeft niet: Matters not
- de alles: the everything
- Het laten: The letting
- de ik: the I
- @ Links de: @ Left the
Error Mining: Conclusion

- Error mining metric spots many errors
- Coverage can be increased dramatically
- But
  - only those errors that lead to parsing failure
  - coverage is not the same as accuracy
Automated Lexical Acquisition

joint work with Tim Van de Cruys
Use parser to find category of unknown word

- For each problematic word, a large number of sentences ($\pm 100$) is parsed
- Parser assigns “all” lexical categories to unknown word
- Different from unknown word heuristics that are used normally
- POS-tagger keeps filtering implausible tag combinations
- Note which tag was used for the unknown word in the best parse
- Use these tags as input to Maximum Entropy classification algorithm
Universal tag set

- All tags that belong to open part of speech class
- If a certain tag appears with >15 words in a large corpus, it is considered universal
  - infrequent tags & closed tags ignored
- Slightly manually adapted to get a systematic tagset
  - tagset of 340 universal tags
Adaptation of Disambiguation Component

- Problem: disambiguation model relies heavily on lexicon
  - preference to parse prepositional phrase as prepositional complement to the verb if such a subcategorization frame exists
- Solution: weight of each universal tag: a priori probability
Example

- Unknown word: *sneup*
  - *ik/pron heb/verb zin/noun in/prep sneup/noun*
    - ‘I would like some sneup’
  - *de/det sneup/noun ligt/verb in/prep de/det kast/noun*
    - ‘De sneup is in the cupboard’
  - *ik/pron wil/verb sneup/adverb*
    - ‘I want sneup’

- *sneup* → [noun,2], [adverb,1]

- *sneup* is probably *noun*
Maximum Entropy Classifier

- **Syntactic features:**
  - the tags found by the parser
  - each subattribute of the tags found by the parser

- **Morphological features:**
  - the last three letters of the word
  - ± past participle
  - ± word starts with particle (e.g. *doorlaten*, ‘let through’)
Training the classifier

- 1000 arbitrary words, removed from Alpino dictionary
- parse these words, as if they are unknown
- collect features
- correct tags: as in the original Alpino dictionary
<table>
<thead>
<tr>
<th></th>
<th>precision (%)</th>
<th>recall (%)</th>
<th>f-measure (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>baseline</td>
<td>23.01</td>
<td>20.80</td>
<td>21.85</td>
</tr>
<tr>
<td>only morphology</td>
<td>53.93</td>
<td>59.73</td>
<td>56.68</td>
</tr>
<tr>
<td>only parsing</td>
<td>69.79</td>
<td>72.47</td>
<td>71.11</td>
</tr>
<tr>
<td>combined</td>
<td>77.55</td>
<td>72.15</td>
<td>74.75</td>
</tr>
</tbody>
</table>
Issues

- Some attributes cannot be learned in some combinations
  - for plural nouns, the gender attribute has no effect, yet it is specified on the tag
  - neuter nouns are assigned noun(de,pl) instead of noun(het,pl)

- Some attributes overlap / subsume e.o.

- Infrequent subcategorization frames are not found
Future work

- Develop and evaluate a **cascading classifier**, i.e. a separate classifier for each part of speech

- Develop an algorithm that generalizes over **word paradigms** in order to find certain subfeatures, e.g. subcategorization frames

- Keep the **top n correct parses**, instead of only using the best parse
Evaluation of Robustness

- The main objective is to have a robust algorithm that always produces some solution ...
Evaluation of Robustness

- The main objective is to have a robust algorithm that always produces some solution . . .

Robustness \neq Coverage
Instead

a smooth *degradation in the performance of a system when faced with unexpected input*

- Take into account the performance (accuracy, usefulness, quality) of the result
Proposal

Robustness: in terms of variation in accuracy
Estimate deviation

- $\overline{CA}$: per sentence mean CA
- $CA_i$: Concept Accuracy for sentence $i$
- $N$: number of test sentences

\[
\sqrt{\frac{\sum_{i}^{N} (\overline{CA} - CA_i)^2}{N}}
\]
## Examples

<table>
<thead>
<tr>
<th></th>
<th>mean CA</th>
<th>robustness</th>
</tr>
</thead>
<tbody>
<tr>
<td>10<em>0, 90</em>100</td>
<td>90.00</td>
<td>30.0</td>
</tr>
<tr>
<td>100*90</td>
<td>90.00</td>
<td>0.0</td>
</tr>
<tr>
<td>50<em>80, 50</em>100</td>
<td>90.00</td>
<td>10.0</td>
</tr>
<tr>
<td>50<em>100, 30</em>90, 10<em>80, 10</em>50</td>
<td>90.00</td>
<td>14.8</td>
</tr>
</tbody>
</table>
Example HMM tagger

<table>
<thead>
<tr>
<th>N-grams</th>
<th>Mean Accuracy</th>
<th>Robustness</th>
</tr>
</thead>
<tbody>
<tr>
<td>unigrams</td>
<td>90.64</td>
<td>9.9</td>
</tr>
<tr>
<td>bigrams</td>
<td>95.47</td>
<td>7.4</td>
</tr>
<tr>
<td>trigrams</td>
<td>95.99</td>
<td>7.0</td>
</tr>
</tbody>
</table>
### Results for Alpino

<table>
<thead>
<tr>
<th>Condition</th>
<th>Mean CA</th>
<th>Robustness</th>
</tr>
</thead>
<tbody>
<tr>
<td>no partial parses</td>
<td>89.85</td>
<td>13.6</td>
</tr>
<tr>
<td>no unknown word heuristics</td>
<td>84.94</td>
<td>26.5</td>
</tr>
<tr>
<td>none</td>
<td>69.81</td>
<td>27.2</td>
</tr>
<tr>
<td>none</td>
<td>59.29</td>
<td>45.5</td>
</tr>
</tbody>
</table>
# Results for Alpino

<table>
<thead>
<tr>
<th>Configuration</th>
<th>mean CA</th>
<th>robustness</th>
</tr>
</thead>
<tbody>
<tr>
<td>89.85 13.6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>no partial parses</td>
<td>84.94</td>
<td>26.5</td>
</tr>
<tr>
<td>no unknown word heuristics</td>
<td>69.81</td>
<td>27.2</td>
</tr>
<tr>
<td>none</td>
<td>59.29</td>
<td>45.5</td>
</tr>
<tr>
<td>CGN Treebank (spoken language)</td>
<td>80.33</td>
<td>24.5</td>
</tr>
</tbody>
</table>
Discussion

- only interested in variation *below* mean
- depends on availability of representative test corpora
- . . .