The resolution of PP attachment ambiguities in German

Martin Volk
Stockholm University
volk@ling.su.se

PP attachment ambiguity

```
<table>
<thead>
<tr>
<th>Sentence</th>
<th>Sentence</th>
</tr>
</thead>
<tbody>
<tr>
<td>verb</td>
<td>verb</td>
</tr>
<tr>
<td>NP</td>
<td>NP</td>
</tr>
<tr>
<td>PP</td>
<td>PP</td>
</tr>
<tr>
<td>Check</td>
<td>Check</td>
</tr>
<tr>
<td>Deine Emails in der Badehose</td>
<td>Deine Emails in der Badehose</td>
</tr>
<tr>
<td>det</td>
<td>noun</td>
</tr>
<tr>
<td>noun</td>
<td>PP</td>
</tr>
<tr>
<td>Deine Emails in der Badehose</td>
<td>Deine Emails in der Badehose</td>
</tr>
</tbody>
</table>
```

Check Deine Emails in der Badehose
Check Deine Emails in der Badehose
PP attachment ambiguities

- Def.: In German a PP is in an **ambiguous position** if it follows immediately after a noun in the *Mittelfeld.*
  - *Du hast Deine Emails in der Badehose gecheckt.*
- We concentrate on **verb vs. noun attachment** ambiguities. I.e. we disregard adjective attachments.
- We ignore the difference between adjunct vs. complement function.

Overview of the talk

- Unsupervised approach to disambiguation
- Supervised approaches to disambiguation
- Using the Web
Previous approaches to PP attachment

- **Structural constraints:**
  - Minimal attachment: Use as few nonterminals as possible.
  - Right Association: Attach to the most recent phrase.

- **Linguistic constraints:**
  - Use subcategorisation information (*to ask for*).
  - Use semantic type: temporal PP is attached to the verb.

Statistical approaches

- **Supervised**
  - Learn attachment preferences from treebank
    - For English: up to 84% accuracy
  - Learn from treebank and use WordNet
    - For English: up to 88% accuracy

- **Unsupervised**
  - Learn attachment preferences from shallow parsed corpus
    - For English: 80-84% accuracy
Our approach

- Unsupervised statistical approach (combined with some linguistics)
- For German
- Learn attachment preferences from a shallow parsed corpus
- Use simple cooccurrence measure
  \[ \text{cooc}(\text{noun},\text{prep}) = \frac{\text{freq}(\text{noun},\text{prep})}{\text{freq}(\text{noun})} \]

Example of cooccurrence measure

For: *Check deine Emails in der Badehose*

- \( \text{freq}(\text{Emails}, \text{in}) = 50 \)
- \( \text{freq}(\text{Emails}) = 10'000 \)
  \[ \rightarrow \text{cooc}(\text{Emails}, \text{in}) = 0.005 \]
- \( \text{freq}(\text{check}, \text{in}) = 15 \)
- \( \text{freq}(\text{check}) = 1'000 \)
  \[ \rightarrow \text{cooc}(\text{check}, \text{in}) = 0.015 \]
Training Corpus

Annotate a 6 million words computer journal corpus (raw text) through
1. Named entity recognition and classification
2. PoS-Tagging
3. Lemmatisation
4. NP/PP chunking
5. Clause boundary detection

→ Learn cooc(noun,prep) and cooc(verb,prep)

Named Entity Recognition

is complicated in German since all nouns are capitalized.
Named entity classification into
- person names
- geographical names (mostly cities and countries)
- company names
Experiment setup

1. Learn cooccurrence values from the training corpus
2. Evaluate against the CZ test set
3. Evaluate against NEGRA test set
4. Exchange the training corpus
   1. Neue Zürcher Zeitung as training corpus
   2. Web as training corpus

The learner

- if Noun_Prep sequence
  then: $\text{freq}(\text{Noun})++$, $\text{freq}(\text{Noun}, \text{Prep})++$
  elsif Noun_X with X <> Prep
  then: $\text{freq}(\text{Noun})++$
- if Verb...Prep$_1$...Prep$_2$ in clause
  then: $\text{freq}(\text{Verb})+=2$,
  $\text{freq}(\text{Verb}, \text{Prep}_1)++$, $\text{freq}(\text{Verb}, \text{Prep}_2)++$
  elsif Verb without Prep in clause
  then: $\text{freq}(\text{Verb})++$
What is learned?

<table>
<thead>
<tr>
<th>word W</th>
<th>prep. P</th>
<th>freq(W,P)</th>
<th>freq(W)</th>
<th>cooc(W,P)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Höchstmass</td>
<td>an</td>
<td>13</td>
<td>13</td>
<td>1.000</td>
</tr>
<tr>
<td>Hinblick</td>
<td>auf</td>
<td>133</td>
<td>135</td>
<td>0.985</td>
</tr>
<tr>
<td>Verweis</td>
<td>auf</td>
<td>21</td>
<td>22</td>
<td>0.955</td>
</tr>
<tr>
<td>Umgang</td>
<td>mit</td>
<td>293</td>
<td>307</td>
<td>0.954</td>
</tr>
<tr>
<td>logiert</td>
<td>unter</td>
<td>55</td>
<td>56</td>
<td>0.982</td>
</tr>
<tr>
<td>paktiert</td>
<td>mit</td>
<td>13</td>
<td>14</td>
<td>0.928</td>
</tr>
<tr>
<td>verlautet</td>
<td>aus</td>
<td>16</td>
<td>19</td>
<td>0.842</td>
</tr>
</tbody>
</table>

The evaluation corpora
⇒ The test sets
The Computer Zeitung (CZ) treebank

- 3'000 manually annotated sentences that contain ambiguously located PPs
- German domain specific newspaper texts
  → 4562 PPs in ambiguous positions
    1761 with verb attachment (39%)
    2801 with noun attachment (61%)

Extraction of 5-tuples from treebank sentences

Sentence:

\textit{Check (deine Emails) (in der Badehose)}

1. Verb: \textit{check}
2. Reference noun N1:
3. Preposition:
4. PP-noun N2:
5. Function:
Extraction of 5-tuples from treebank sentences

Sentence:

*Check (deine Emails) (in der Badehose)*

1. Verb: check
2. Reference noun N1: Emails
3. Preposition: in
4. PP-noun N2: 
5. Function: 

Extraction of 5-tuples from treebank sentences

Sentence:

*Check (deine Emails) (in der Badehose)*

1. Verb: check
2. Reference noun N1: Emails
3. Preposition: in
4. PP-noun N2: Badehose
5. Function: verb attachment
Disambiguation Algorithm 1

if (cooc(N1, P) && cooc(V, P)) then

    if (cooc(N1, P) > cooc(V, P)) then
        noun attachment
    else
        verb attachment

Disambiguation Results 1

<table>
<thead>
<tr>
<th></th>
<th>correct</th>
<th>incorrect</th>
<th>accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>noun att.</td>
<td>925</td>
<td>60</td>
<td>93.91%</td>
</tr>
<tr>
<td>verb att.</td>
<td>743</td>
<td>608</td>
<td>55.00%</td>
</tr>
<tr>
<td>total</td>
<td>1668</td>
<td>668</td>
<td>71.40%</td>
</tr>
<tr>
<td>coverage</td>
<td>2336 / 4143 (57%)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
The noun factor

- Observation: Verb+Prep cooccurrence values are too strong
- Needed: a factor to strengthen Noun+Prep values
- Based on the overall attachment tendency:
  - cooc(all_Nouns, all_Preps) = 0.182
  - cooc(all_Verbs, all_Preps) = 0.774
  → noun factor = 0.774 / 0.182 = 4.25

Disambiguation Algorithm 2

if (cooc(N1,P) \&\& cooc(V,P)) then
  if ((cooc(N1,P) * noun_factor) > cooc(V,P)) then
    noun attachment
  else
    verb attachment
Disambiguation Results 2
with noun factor 4.25

<table>
<thead>
<tr>
<th></th>
<th>correct</th>
<th>incorrect</th>
<th>accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>noun att.</td>
<td>1377</td>
<td>280</td>
<td>83.10%</td>
</tr>
<tr>
<td>verb att.</td>
<td>524</td>
<td>157</td>
<td>76.94%</td>
</tr>
<tr>
<td>total</td>
<td>1901</td>
<td>437</td>
<td>81.31%</td>
</tr>
</tbody>
</table>

Coverage increase

- Using lemmas
  - verb lemmas
  - noun lemmas (incl. reduction of compounds)
    - Forschungsinstitut → Institut
      - coverage of 83% (a gain of 26%)
      - accuracy of 78.13% (a loss of 3%!!)

- Using proper name classes
  - coverage of 86%
  - accuracy of 78.31%
Coverage increase

- not explored
  - diminutive forms (Kästchen → Kasten)
  - different nominalizations (das Zusammenschalten, die Zusammenschaltung)
  - number words (Hundert, Million, Milliarde)
  - weak nominal prefixes (Vizepräsident → Präsident)

Accuracy increase

Distinguish sure and possible attachment!

- Annotate sure verb attachment (all PPs not following a noun)
  - An EU-externe Länder dürfen Daten nur exportiert werden, ...

- Annotate sure noun attachment (e.g. PPs in sentence-initial phrases)
  - Die Abkehr von den proprietären Produkten erzeugt mehr Wettbewerb ...
**Shallow parsed Corpus**

- learning
- better
- cooc-values

**Corpus**

(6 million)

**annotation**

**sure PP attachment**

**attachment preferences**

---

**Using pair and triple frequencies**

- Hypothesis: Accuracy increases if the PP’s noun is also used.
  - Peter *saw* the *thief* with his own *eyes*.
  - Peter *saw* the *thief* with the red *coat*.

\[ cooc(N_1, P, N_2) = \frac{freq(N_1, P, N_2)}{freq(N_1)} \]
Disambiguation Algorithm 3

\[
\text{if (cooc(N1,P,N2) && cooc(V,P,N2)) then}
\]
\[
\quad \text{if (cooc(N1,P,N2) * noun_factor) > cooc(V,P,N2)}
\]
\[
\quad \quad \text{then: noun attachment}
\]
\[
\quad \quad \text{else: verb attachment}
\]
\[
\text{elsif (cooc(N1,P) && cooc(V,P)) then}
\]
\[
\quad \text{if ((cooc(N1,P) * noun_factor) > cooc(V,P))}
\]
\[
\quad \quad \text{then: noun attachment}
\]
\[
\quad \quad \text{else: verb attachment}
\]

Results for unsupervised method

<table>
<thead>
<tr>
<th>decision level</th>
<th># of cases</th>
<th>accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>support verb unit</td>
<td>97</td>
<td>100.00%</td>
</tr>
<tr>
<td>triple comparison</td>
<td>953</td>
<td>84.36%</td>
</tr>
<tr>
<td>pair comparison</td>
<td>2813</td>
<td>79.95%</td>
</tr>
<tr>
<td>cooc(N1,P) &gt; thr.</td>
<td>74</td>
<td>85.13%</td>
</tr>
<tr>
<td>cooc(V,P) &gt; thr.</td>
<td>91</td>
<td>84.61%</td>
</tr>
<tr>
<td>total</td>
<td>4028</td>
<td>81.67%</td>
</tr>
<tr>
<td>(coverage: 90% )</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Supervised methods

Supervised methods for PP attachment disambiguation

- idea: learn attachment tendencies from manually disambiguated cases (= treebank)
- Example methods
  - Transformation-based learning
  - Back-off method
Training material

- the NEGRA test set (6064 test cases)
- the CZ test set (4562 test cases)
  ➔ as basis for small training sets!!

The NEGRA treebank

- 10'000 manually annotated sentences
- German newspaper texts
  ➔ 6064 PPs in ambiguous positions
    2664 with verb attachment (44%)
    3400 with noun attachment (56%)
The sparse data problem

Many quadruples will occur rarely!
Therefore: clustering is needed
- verbs → lemmas
- contracted prepositions → base forms
- proper names → class labels
- numbers → number tag
- nouns → lemmas (of last compound element)

Back-off Method

- by Collins and Brooks
- idea: learn attachment tendencies from manually disambiguated cases
- in case of missing information back-off to the next level
  - quadruples (Verb, Noun1, Prep, Noun2)
  - triples (which include the preposition)
  - pairs (which include the preposition)
  - prepositions alone
Back-off Method

if (freq(V, N1, P, N2) > 0) then
  if (freq(noun_att, V,N1,P,N2) / (freq(V,N1,P,N2) ) > 0.5
      then noun attachment
      else verb attachment
  elsif (( fr( V,N1,P) + fr(V,P,N2) + fr(N1,P,N2)) > 0) then
    ...
  elsif (( fr(V,P) + fr(P,N2) + fr(N1,P) ) > 0 ) then
    ...

Results for the Back-off method
trained on the NEGRA test set and
evaluated against the CZ test set

<table>
<thead>
<tr>
<th>decision level</th>
<th># of cases</th>
<th>accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>quadruples</td>
<td>8</td>
<td>100.00%</td>
</tr>
<tr>
<td>triples</td>
<td>329</td>
<td>88.75%</td>
</tr>
<tr>
<td>pairs</td>
<td>3040</td>
<td>75.66%</td>
</tr>
<tr>
<td>preposition</td>
<td>1078</td>
<td>64.66%</td>
</tr>
<tr>
<td>default</td>
<td>14</td>
<td>64.29%</td>
</tr>
<tr>
<td>total</td>
<td>4469</td>
<td>73.98%</td>
</tr>
</tbody>
</table>

(coverage: 100%)
Results for the Back-off Method

<table>
<thead>
<tr>
<th>Training corpus</th>
<th>Test set</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>NEGRA</td>
<td>CZ</td>
<td>74.0%</td>
</tr>
<tr>
<td>CZ</td>
<td>NEGRA</td>
<td>68.3%</td>
</tr>
<tr>
<td>NEGRA + 4/5 CZ</td>
<td>1/5 CZ * 5</td>
<td>(\varnothing = 79.4%)</td>
</tr>
</tbody>
</table>

Intertwined Combination

1. Support verb unit
2. -
3. -
4. Triple comparison
5. Pair comparison
6. Threshold compar.
7. -
8. -
9. -

1. -
2. Quadruples
3. Triples
4. -
5. -
6. -
7. Pairs
8. Preposition
9. Default
Comparison of the results

<table>
<thead>
<tr>
<th></th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unsupervised</td>
<td>79.14</td>
</tr>
<tr>
<td>Supervised</td>
<td>73.98</td>
</tr>
<tr>
<td>Combined</td>
<td>80.98</td>
</tr>
</tbody>
</table>

Conclusions

• Unsupervised method is as good as supervised method over small training corpus.
• Combination of unsupervised and supervised leads to the best results.
• On the side:
  • Most important for increase of accuracy: sure attachment distinction and triple frequencies
  • Most important for increase of coverage: lemmatisation and proper name clustering
Using the WWW to resolve PP attachment ambiguities

Getting WWW frequencies

- Search engines report: number of pages found
  - Noun NEAR Preposition
  - Noun
  - Verb NEAR Preposition
  - Verb
  → bad results
Getting WWW frequencies

- Search engines report:
  - number of pages found
    - Noun1 NEAR Preposition NEAR Noun2
    - Noun1
    - Verb NEAR Preposition NEAR Noun2
    - Verb

Disambiguation-Algorithm

- if (cooc(N1,P,N2) && cooc(V,P,N2)) then
  - if (cooc(N1,P,N2) > cooc(V,P,N2)) then
    - noun attachment
  - else
    - verb attachment
- else (default)
  - noun attachment
Disambiguation Results 2
including triples, lemmas and threshold test

<table>
<thead>
<tr>
<th></th>
<th>correct</th>
<th>incorrect</th>
<th>accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>noun att.</td>
<td>1615</td>
<td>459</td>
<td>77.9%</td>
</tr>
<tr>
<td>verb att.</td>
<td>735</td>
<td>300</td>
<td>71.0%</td>
</tr>
<tr>
<td>total</td>
<td>2350</td>
<td>759</td>
<td>75.6%</td>
</tr>
<tr>
<td>coverage</td>
<td>3109 (71%)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Conclusion

- WWW frequencies can be used to resolve PP attachment ambiguities.
- Triple cooccurrence values are better than tuple cooccurrence values.
- Combining full forms and lemmas increases the attachment rate.
- WWW queries are imprecise.
Note

- Beware: V, N1, P, N2 makes the PP attachment task look easier than it is!!

- Often there are sequences such as:
  - V ... NP_PP_PP or
  - V ... NP_NP_PP

How about ...
resolving a classical example with web frequencies

*He sees the man with the telescope.*

\[
\begin{align*}
\text{freq}(\text{sees NEAR with}) &= 244'865 \\
\text{freq}(\text{sees}) &= 1'806'082 \\
\rightarrow \text{cooc}(\text{sees, with}) &= 0.124 \\
\text{freq}(\text{man NEAR with}) &= 2'550'804 \\
\text{freq}(\text{man}) &= 14'444'376 \\
\rightarrow \text{cooc}(\text{man, with}) &= 0.176
\end{align*}
\]
How about ...
resolving a classical example with web frequencies

He sees the man with the telescope.

freq(sees NEAR with NEAR telescope) = 150
freq(sees) = 1'806'082
→ cooc(sees, with, telescope) = 8.305 * 10^{-5}

freq(man NEAR with NEAR telescope) = 478
freq(man) = 14'444'376
→ cooc(man,with,telescope) = 3.309 * 10^{-5}