Using compounds list for German decompounding in a back-off scenario

Pedro Bispo Santos
Introduction and motivation

- Compounds in several languages:
  - English, Greek, Turkish, German, Scandinavian languages, and so on
  - Significant number of compounds in German: 5% of 9 million tokens (Schiller, 2005)

- Applications in several areas: IR, MT

- When should a word be decompounded?
  - Verkehrzeichen? ✔
  - Löwenzahn? ✗
  - Aktionsplan? Aktion+Plan or Akt+Ion+Plan
Decompounding Algorithms

- Monz and Rijke (2002) decompounding algorithm
- Recursively splitting from the left to the right
- Lexicon-based
- JWord API
Decompounding Algorithms

- JWord API doesn’t detect non-compounds
- Ranking function implemented
- Web1T Corpus as background corpus
- Köhn and Knight (2003) corpus frequency-based metric:
Decompounding Algorithms

- Holz and Biemann (2008) decompounding algorithm
- Compact Patricia Tries
- Prefixes CPT, Suffixes CPT
- Automatische Sprachverarbeitung (ASV) Toolbox
Compound Lists - GermaNet

- GermaNet compounds list
- 54,759 split nominal compounds
- Modifier + Head decomposition
- Recursive decomposition
- **Hypothesis:** Compound lists improve existing approaches
Using compounds list

HUNDERTMETERLAUF

Compounds List

HUNDERT + METER + LAUF
Algorithms as back-off solution

NASENNEBENHÖHLENENTZÜNDUNG

Compounds List

Back-off solution

Decompounding algorithm

NASE(N)+NEBENHÖHLEN+ENTZÜNDUNG
Evaluation – Decompounding APIs

- JWord Splitter API
  - [https://github.com/danielnaber/jwordsplitter](https://github.com/danielnaber/jwordsplitter)
  - Algorithm similar to Monz and Rijke (2002)
  - German dictionary igerman98

- ASV Toolbox API
  - [http://wortschatz.uni-leipzig.de/~cbiemann/software/toolbox/](http://wortschatz.uni-leipzig.de/~cbiemann/software/toolbox/)
Evaluation

- Corpus created by Marek (2006)
- 158,653 compounds
- Semi-automatically generated
  - Reviewed by human annotators
- Precision and recall (Köhn and Knight, 2003):
  - cc – compound correctly split
  - wfc – compound wrongly split
  - wnc – compound which was not split

\[
P = \frac{cc}{cc + wfc}
\]

\[
R = \frac{cc}{cc + wfc + wnc}
\]
Results

- Compound list without back-off algorithm: Low Recall

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Recall</th>
<th>Precision</th>
</tr>
</thead>
<tbody>
<tr>
<td>GermaNet list</td>
<td>.083</td>
<td>.917</td>
</tr>
</tbody>
</table>
Results

- Compound list without back-off algorithm: Low Recall
- Decompounding algorithms without compound list: reasonable performance

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Recall</th>
<th>Precision</th>
</tr>
</thead>
<tbody>
<tr>
<td>GermaNet list</td>
<td>0.083</td>
<td>0.917</td>
</tr>
<tr>
<td>ASV Toolbox</td>
<td>0.755</td>
<td>0.799</td>
</tr>
<tr>
<td>Jword</td>
<td>0.766</td>
<td>0.799</td>
</tr>
</tbody>
</table>
Results

- Compound list without back-off algorithm: Low Recall
- Decompounding algorithms without compound list: reasonable performance
- Usage of algorithms as backoff solution: best outcome
- Significant improvement

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Recall</th>
<th>Precision</th>
</tr>
</thead>
<tbody>
<tr>
<td>GermaNet list</td>
<td>.083</td>
<td>.917</td>
</tr>
<tr>
<td>ASV Toolbox</td>
<td>.755</td>
<td>.799</td>
</tr>
<tr>
<td><strong>ASV Toolbox with GermaNet list</strong></td>
<td><strong>.766</strong>*</td>
<td><strong>.803</strong>*</td>
</tr>
<tr>
<td>JWord</td>
<td>.766</td>
<td>.799</td>
</tr>
<tr>
<td><strong>JWord with GermaNet list</strong></td>
<td><strong>.780</strong>*</td>
<td><strong>.808</strong>*</td>
</tr>
</tbody>
</table>
Results – error analysis

- **Badezimmer** is only split using a compound list
  - The reason for JWord is its frequency: *Badezimmer* is a very common word
  - ASV toolbox: Badezimmer is a prefix/suffix in the CPT
- **Kartenaufbau** is split as *karten+aufbau* by the algorithms
  - It is debatable whether or not *aufbau* should be split
  - The same occurs for *ausdrucken*
  - Prepositions as constituents are hard to deal by these algorithms
- The gold standard should be revised
Conclusions and Future Works

- Three strategies were evaluated:
  - Using a compound list
  - Using algorithms for automatically decompounding
  - Using a compound list with decompounding algorithms as a back-off solution
    - This approach had the best outcome

- Available as a module in DKPro Core:
  - [https://code.google.com/p/dkpro-core-asl/](https://code.google.com/p/dkpro-core-asl/)
Conclusions and Future Works

- Regarding results:
  - Using only compound list has low recall
  - Algorithms tested failed with frequent compounds
  - Prepositions as constituents are hard to deal with
Conclusions and Future Works

Future work:

- Also use words which are not compounds in the evaluation dataset
- Measure the accuracy regarding false positives
- Revise existing compounds in our gold standard
Thank you!