Project sketch 2:
Distributional Semantics of Derivation
Background: Derivational morphology

• In computational linguistics, derivational morphology has received relatively little attention
  – Stemming gets you (relatively) far
  – English has a lot of zero conversion (*fish, view*)
  – ...

• Exactly one (English) resource around: CatVar
  – (Some work in the context of other resources, e.g. WordNet)
Current work on derivation [submitted to ACL 2013]

• Clustering words into *derivational families*
  – German

• *Rule-based* approach: Focus on high precision
  – About 150 rewrite rules, extracted from grammar books
  – Prefixation + Affixation, includes stem changes
  – Applied to lemma list extracted from sdewac
    • Result: 18,000 non-singleton derivational families

• Evaluation: Precision 93%, Recall 61%
  – NB. word formation rules are morphological, not semantic
# Example families

<table>
<thead>
<tr>
<th>German Word</th>
<th>English Word</th>
</tr>
</thead>
<tbody>
<tr>
<td>Liberalisierung_Nf</td>
<td>machbar_A</td>
</tr>
<tr>
<td>liberalisierend_A</td>
<td>Machbare_Nn</td>
</tr>
<tr>
<td>Liberalität_Nf</td>
<td>machend_A</td>
</tr>
<tr>
<td>liberal_A</td>
<td>Macherin_Nf</td>
</tr>
<tr>
<td>antiliberal_A</td>
<td>Machen_Nn</td>
</tr>
<tr>
<td>Antiliberalismus_Nm</td>
<td>Machenschaft_Nf</td>
</tr>
<tr>
<td>Liberale_N</td>
<td>machen_V</td>
</tr>
<tr>
<td>Liberalismus_Nm</td>
<td>Mache_Nf</td>
</tr>
<tr>
<td>liberalistisch_A</td>
<td>Macher_Nm</td>
</tr>
<tr>
<td>liberalisieren_V</td>
<td>vermachen_V</td>
</tr>
</tbody>
</table>
Project goal

• Model *distributional semantics* of rules [German]
  – Simplification: Rule expresses semantic relation
  – Semantics of rule $r$ is *transformation* $\text{tr}_r$, so that for each word pair $(w_1,w_2)$ in $r$: $\text{tr}_r(\text{vec}(w_1)) = \text{vec}(w_2)$

• In the spirit of work on modeling other kinds of semantic relations (synonymy, analogy, ... -- e.g. Peter Turney’s work)

• Central question: What kind of object can $\text{tr}$ be?
  – Vector
  – Matrix
Vectors for transformations

• Simplest case: $w_2 = w_1 + \text{tr}_r$
  – $\text{vec(Verkäufer)} = \text{vec(verkaufen)} + \text{tr}_r$

• This is not always sufficient!
  – cf. later
Transformations and Vector Space Topology

• Vector spaces with words as dimensions
  – Vector similarity is *topical similarity*
  – Quasi-synonymy rules: Identity transformation
  – Other rules categories: polarity change, sentiment, ...
    • In this space, vectors are reasonable to model transformations

• Vector spaces with syntactic dimensions
  – Vector similarity is *functional similarity*
  – Quasi-synonymy rules: involve *per-argument shifts*
    • Verb $\rightarrow$ Event Noun: Subj $\rightarrow$ Gen, Obj $\rightarrow$ Gen, Adv $\rightarrow$ Adj
    • Verb $\rightarrow$ Agent Noun: Subj $\rightarrow$ $\emptyset$, Obj $\rightarrow$ Gen, $\emptyset$ $\rightarrow$ Adj
  – Vectors are insufficient!
Matrices for transformations

• kochen, Koch [Event nominalization]
  – Subj → Gen, Obj → Gen, Adv → Adj

\[
\begin{bmatrix}
  \text{subj} & \text{obj} & \text{gen} & \text{adj} & \text{adv} \\
  \text{subj} & 0 & 0 & 1 & 0 & 0 \\
  \text{obj} & 0 & 0 & 1 & 0 & 0 \\
  \text{gen} & 0 & 0 & 0 & 0 & 0 \\
  \text{adj} & 0 & 0 & 0 & 0 & 1 \\
  \text{adv} & 0 & 0 & 0 & 0 & 0 
\end{bmatrix}
\]

• cf. ideas from Baroni et al. (project sketch 1)
Rule Instance Clustering

• Not all instances of a rule share a relation
  – Exceptions: -er: agentivization, but kochen/Kocher
  – Generality: -ung: event vs. state vs. object nominalizations

• Strategy: group rule instances into clusters and/or detect outliers
  – When rule semantics is vector (lexical space), standard clustering methods applicable
  – When rule semantic is not (syntactic space), new methods are necessary
Potential Applications

• **Rule subclassification**: Distinguish “quasi-paraphrase” rules from non-paraphrastic ones
  – (liberal, Liberaler) vs. (liberal, liberalistisch)
  – Ideally, also classify non-paraphrastic rules

• **Smoothing**: If w1 is frequent and w2 is not, predict vector for w2 from vectors of w1
  – Useful for all kinds of distributional tasks

• **Information transfer**: Generalization of syntacto-semantic knowledge across parts of speech
  – selectional preferences, semantic role labeling models, ...
Outlook

• Cross-lingual information transfer:
  – e.g. learn *gender* indicators by observing context of translations of female derivations vs. male derivations
    • Schläfer vs. Schläferin -> sleeper

• (Probably not in scope of project)
Collaborations

- Established collaboration with Jan Snajder, University of Zagreb
- Metaphor interpretation (Sabine Schulte im Walde)
- Incremental sentence meaning project
- B area: Theoretical accounts of derivation / word formation