Semantic Mapping for Lexical Sparseness Reduction in Parsing

Simon Šuster, Gertjan van Noord
University of Groningen

New Frontiers in Parsing and Generation Workshop
12. 4. 2013
Context and motivation

- we know semantics can help syntactic parsing
  - specifically: semantic classes for mostly data-driven systems
- classes provide generalization for reducing lexical sparseness
- obtain a baseline using human-built semantic inventories for Dutch
  - issues of such an approach

Example

- “open with scissors” not in training $\Rightarrow$
- but “knife” and “scissors” share the class (cutting tools) $\Rightarrow$
- correct analysis possible
Comparison to related work


1 applying generalization \textit{indiscriminately}
⇒ isolate relevant dependency types
Comparison to related work


1. applying generalization **indiscriminately**
   ⇒ isolate relevant dependency types

2. enhancing **base** parsers with semantic classes
   ⇒ enhance an already well-performing bilexical component of a system driven by a hand-crafted grammar
Comparison to related work


1. applying generalization **indiscriminately**
   ⇒ isolate relevant dependency types

2. enhancing **base** parsers with semantic classes
   ⇒ enhance an already well-performing bilexical component of a system driven by a hand-crafted grammar

3. usually **extremes** of granularity are taken as representation level,
   ⇒ “appropriate” level of generality
Alpino

- parser for Dutch
- manually crafted HPSG grammar, augmented to represent dependency structure
- MaxEnt disambiguation component

Lexical association component

- part of disambiguation component
- bilexical preferences measured by normalized PMI and learned from a 500M word corpus
- improvement on the base parser (Van Noord, 2010)

Example

(verb,SU) dependency type:

\[
\left( \text{“drink”}, \verb, \text{su}, \text{noun}, \text{“baby”} \right) \quad 0.28, \quad 4.89
\]

\[w_1, \text{pos}_1, \text{relation}, \text{pos}_2, w_2\]
Selection of dependency types

- identify types whose bilexical sparseness hurts parser the most
  \[\Rightarrow\] correlation between coverage and parsing accuracy: Cramer’s Φ, odds ratio:

<table>
<thead>
<tr>
<th>Type</th>
<th>Odds</th>
<th>φ coef.</th>
</tr>
</thead>
<tbody>
<tr>
<td>(adj,MOD)</td>
<td>2.653</td>
<td>0.2</td>
</tr>
<tr>
<td>(noun,CNJ)</td>
<td>2.042</td>
<td>0.12</td>
</tr>
<tr>
<td>(noun,MOD)</td>
<td>1.962</td>
<td>0.11</td>
</tr>
</tbody>
</table>

- correct parse of (noun,CNJ) is then 2 times more likely with available bilexical preference
- use Cornetto, a Dutch wordnet

**Fine**: immediate synset (SYN)
- take the 1st most-prominent sense
- little generalization

**Coarse**: semantic type (ST)
- assigned to 50% of lexical units (LUs)
- \(\sim 20\) POS-dependent labels: “action”, “human”, “concrete” …
Intermediate (INT)

- find a level that is both general and precise
- calculate generality of a synset
- if too concrete, map to a more general synset
- treat Cornetto as a tree-like directed graph
  - hypernymic relations are arcs
  - synsets are nodes

- **Information Content** is:
  (Sánchez et al. 2011)

\[
IC(s) = - \log \frac{|leaves_s|}{|subsumers_s|} + 1
\frac{1}{total\_leaves + 1}
\]
Intermediate (INT)

- find a level that is both general and precise
- calculate generality of a synset
- if too concrete, map to a more general synset
- treat Cornetto as a tree-like directed graph
  - hypernymic relations are arcs
  - synsets are nodes
- **Information Content** is:
  (Sánchez et al. 2011)

\[
IC(s) = -\log \frac{|leaves_s|}{|subsumers_s|} + 1
\]

\[
IC > \delta
\]
Semantic representation: 3 levels

**Intermediate (INT)**

- find a level that is both general and precise
- calculate generality of a synset
- if too concrete, map to a more general synset
- treat Cornetto as a tree-like directed graph
  - hypernymic relations are arcs
  - synsets are nodes
- **Information Content** is:
  (Sánchez et al. 2011)

\[
IC(s) = -\log\frac{|leaves_s|}{|subsumers_s|} + 1
\]

\[
IC > \delta
\]
Use of classes

For training
1. obtain relevant dependencies in Lexical Association model
2. make a copy with classes replacing words
3. calculate MI scores

For testing
- use bilexical preference when possible, back-off to generalized classes otherwise

Test set
- Alpino Treebank: 7,136 sentences of newspaper texts
- parts of Lassy Small: 3,917 sentences
Example of enhancement

“Utrechtse Camera bioscoop” (Camera cinema in Utrecht)

⇒ no bilexical preference for <Utrechtse, mod, bioscoop>
⇒ parser backs-off to a generalization of “Utrechtse”
⇒ new instance: “place_{adj} Camera bioscoop”
⇒ preference now exists for <place_{adj}, mod, bioscoop>
⇒ parse correct

• Cornetto coverage in test: 60% (backed-off tokens only)
Results II

- **SYN**: number of improvements **levels** the number of deteriorations . . .
  - (noun/CNJ) is the best performing type
- **ST**: poor performance due to overgeneralizing
- **INT** ($\delta_{IC} = 6$): seems only slightly better than ST

<table>
<thead>
<tr>
<th>All 3 dependency types</th>
<th>(noun/CNJ) only</th>
</tr>
</thead>
<tbody>
<tr>
<td>% found</td>
<td># Improved</td>
</tr>
<tr>
<td>SYN</td>
<td>7.8</td>
</tr>
<tr>
<td>ST</td>
<td>62.1</td>
</tr>
</tbody>
</table>

| SYN    | 7         | 2           |
| ST     | 20        | 26          |
| INT    | 16        | 19          |
Remarks

- synset level: very modest improvement on nominal conjunction
- intermediate level: no improvement
- unlike Agirre et al. 2011 and MacKinlay et al. 2012, semantic types are the worst performer
- for more impact, more generalization is needed, but it introduces noise
- parser’s degree of lexicalization might affect the “working” space
  - bilexical component gets “the low-hanging fruit”

- next: distributional semantic methods
  - increased coverage
  - alternative granularity
  - sense disambiguation in context
  - composition