RESTORING EMPTY CATEGORIES FOR ARABIC
What is the empty category problem?

- Various non-word nodes in parse trees
  - Usually indicating non-local syntactic relations
  - Necessary for predicate-argument structure

(SQ (WHNP-1 What)
  do
  (NP-2 you)
  (VP want (S (NP *-2)
    (VP to
      see
      (NP *T*-1))))))?
In what distribution?

<table>
<thead>
<tr>
<th>Type</th>
<th>Antecedent</th>
<th>Arabic</th>
<th>English</th>
</tr>
</thead>
<tbody>
<tr>
<td>NP <em>T</em></td>
<td>WHNP</td>
<td>30%</td>
<td>17%</td>
</tr>
<tr>
<td>NP *</td>
<td>None</td>
<td>24%</td>
<td>19%</td>
</tr>
<tr>
<td>NP <em>T</em></td>
<td>NP</td>
<td>17%</td>
<td></td>
</tr>
<tr>
<td>WHNP <em>0</em></td>
<td>None</td>
<td>14%</td>
<td>3.5%</td>
</tr>
<tr>
<td>NP *</td>
<td>NP</td>
<td>12%</td>
<td>36%</td>
</tr>
<tr>
<td>ADVP <em>T</em></td>
<td>WHADVP</td>
<td>1.3%</td>
<td>5%</td>
</tr>
<tr>
<td>NP *</td>
<td>SBAR</td>
<td>0.5%</td>
<td></td>
</tr>
<tr>
<td>SBAR <em>ICH</em></td>
<td>None</td>
<td>0.4%</td>
<td></td>
</tr>
<tr>
<td>PP <em>ICH</em></td>
<td>None</td>
<td>0.3%</td>
<td></td>
</tr>
<tr>
<td>NP <em>T</em></td>
<td>None (???)</td>
<td>0.2%</td>
<td></td>
</tr>
</tbody>
</table>

51,068 in Arabic training data; 50,961 in English
Missing in Arabic: S *T* → S (8%, 4% SBAR), WHADVP 0 (1.1%)
Preliminary Comparison to English

- In some ways, easier
  - No ambiguity between nominal and adverbial for null complementizers
  - NP * without antecedents more common (2:1); in English it’s almost the other way around
  - No S traces

- New things
  - Extensive topicalization
  - More wh-traces, but fewer adverbial
Previous work on English

- **Parser-integrated approach**
  - Collins (1999); Dienes and Dubey (2003); Schmid (2006)

- **Post-processing**
  - Johnson (2002); Levy and Manning (2004); Campbell (2004); Gabbard, Kulick, and Marcus (2006); Filimonov and Harper (2007)

- Only non-English work is on Chinese by Guo, et. al. (2007)
Approach for Arabic

- We adopt basically the model of Gabbard, et al. (2006)
  - Good performance, flexible
- It applied a series of maximum entropy classifiers to relevant locations in the tree
- However, had a few cascading error problems due to multiple different types competing for the same locations.
- New model: do the inference all at once (CRF)
New Model: Slot Variables

- Assign a **slot variable** to each
  - Unfilled subject and object slot of every verb
  - Unfilled subject of –PRD
  - Unfilled object of PP
- Resumptive pronouns are treated as unfilled
- Each slot variable has the following values
  - Null
  - NP *
  - Each wh-word (variable) which could come from there
  - Each NP which could have topicalized from there
New Model: WH-variables

- Insert a variable for every wh-word
- Its values are all the slots the wh-word could have come from

Who sneezed?

Sneezed/SB
J
Sneezed/OB
J

Null NP *
Who

Null NP *
Who
New model: Path factors

- Between each wh-variable and each of its values, add a path factor.
- This factor will add a “mismatch” feature if one variable points to the other, but not vice-versa.
- If neither points to the other, it adds no features.
- If both match, it adds features based on the path between the trace and antecedent.
New model: Slot Factors

- Every slot variable has an associated slot factor
- This adds features such as:
  - How many argument NPs are present
  - Whether the verb has other arguments: VP, SBAR, etc.
  - Verb’s POS tag
  - Path to topicalized NP and features about the topic location
  - Resumptive pronouns
# Current Results

<table>
<thead>
<tr>
<th>Type</th>
<th>F-measure</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>96.7</td>
</tr>
<tr>
<td>WHNP 0</td>
<td>99.5</td>
</tr>
<tr>
<td>Adverbial Wh-traces</td>
<td>73.7</td>
</tr>
<tr>
<td>Nominal Wh-traces</td>
<td>85.5</td>
</tr>
<tr>
<td>Nominal topicalization</td>
<td>90.1</td>
</tr>
<tr>
<td><em>NP</em> (placement only)</td>
<td>72.1</td>
</tr>
</tbody>
</table>

- *NP* is very poor
  - Hasn’t had much attention yet
    - Lacks some of English’s easy cases
  - Nominal wh-traces about ten points worse than English
    - Looking into why
Future work

- Increase performance
- function-tagging into the same framework
- Do reranking over trees with empty categories restored
- Ideally you’d like it to be in the parser
  - But attempts to do this for lexicalized parsers have lowered parsing performance