Asynchronous Binarization for Synchronous Grammars

John DeNero, Adam Pauls, and Dan Klein
{denero, adpauls, klein} @ cs.berkeley.edu

Overview
In multi-pass decoding with synchronous grammars, rule binarization can be decoupled, rather than synchronized. Each unconstrained monolingual binarization can then be optimized for the relevant stage in decoding.

Parsing Stage: Source-Side Binarization
First, we project the synchronous grammar to the source language

<table>
<thead>
<tr>
<th>Root</th>
<th>Source Yield</th>
<th>Target Yield</th>
<th>Source Projection</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S$</td>
<td>PRP, NN, VBD, PP</td>
<td>PRP, NN, VBD</td>
<td>$S$ → PRP, NN, VBD, PP</td>
</tr>
<tr>
<td>$S$</td>
<td>PRP, NN, VBD, PP</td>
<td>PRP, NN, VBD</td>
<td>$S$ → PRP, NN, VBD, PP</td>
</tr>
<tr>
<td>PP</td>
<td>en casa</td>
<td>at home</td>
<td>PP → en casa</td>
</tr>
<tr>
<td>PP</td>
<td>en casa</td>
<td>indoors</td>
<td>PP → en casa</td>
</tr>
</tbody>
</table>

Next, we binarize the source-side projection (see our NAACL paper)

$S$ → PRP, NN, VBD, PP

Then, we build a source-binarized parse forest via CKY-style parsing

Highlights of our decoder:
- Binarization is chosen to minimize the total number of grammar symbols
- Coarse-to-fine parsing uses subsets of the monolingual grammar projection
- Forests are pruned by thresholding node max-marginals (before LM)

Reranking Stage: Target-Side Binarization
Source-side binarization is collapsed out to create an n-ary forest

The parse forest is then re-binarized for target-side gap adjacency

Target Yield Binarizations
$S$ → PRP, VBD, NN, PP
$S$ → PRP, VBD, PP, NN

Derivations are reranked efficiently with an n-gram language model

Search error reduction relative to n-ary reranking gave 0.3 BLEU