Rule Filtering by Pattern for Efficient Hierarchical Translation

Gonzalo Iglesias\textsuperscript{1} Adrià de Gispert\textsuperscript{2}
Eduardo R. Banga\textsuperscript{1} William Byrne\textsuperscript{2}

\textsuperscript{1}Department of Signal Processing and Communications
University of Vigo, Spain

\textsuperscript{2}Department of Engineering.
University of Cambridge, U.K.

12th Conference of the EACL.
Athens, April 2009.
Outline

1 Refinements in the Cube Pruning Decoder
   - Hiero Cube Pruning
   - Smart Memoization
   - Spreading Neighborhood Exploration

2 Rule Filtering by Pattern
   - Rule Patterns
   - Hiero Shallow
   - Filtering Translations
   - Rule Patterns Revisited
   - Large Language Models and Evaluation
Outline

1. Refinements in the Cube Pruning Decoder
   - Hiero Cube Pruning
   - Smart Memoization
   - Spreading Neighborhood Exploration

2. Rule Filtering by Pattern
   - Rule Patterns
   - Hiero Shallow
   - Filtering Translations
   - Rule Patterns Revisited
   - Large Language Models and Evaluation
Hierarchical Cube Pruning

Very Brief Description!

- CYK: source side, hypotheses recombination, no pruning
- k-best algorithm: uses cube pruning with LM costs to extract efficiently k-best lists
- NIST 2008 Arabic-to-English task, k-best=10000
Outline

1. Refinements in the Cube Pruning Decoder
   - Hiero Cube Pruning
   - Smart Memoization
   - Spreading Neighborhood Exploration

2. Rule Filtering by Pattern
   - Rule Patterns
   - Hiero Shallow
   - Filtering Translations
   - Rule Patterns Revisited
   - Large Language Models and Evaluation

Refinements in the Cube Pruning Decoder

Rule Filtering by Pattern for Efficient Hierarchical Translation
Key aspect of Chiang’s k-best algorithm: memoization!

Each cell reached at least once by the k-best algorithm will store a k-best list

Only after finishing translation you can free memory (Gigs)
Idea: Couldn’t we delete k-best lists on the fly?
Problem: We do not know how many times will each cell be accessed
Solution: Traverse back-pointers twice:
  1st pass: count how many times each cell will be accessed (very fast)
  2nd pass, build translation hyps: Decrease counter for each cell. If counter=0, delete k-best list!

typically reduces memory usage in 30%
1 Refinements in the Cube Pruning Decoder
   • Hiero Cube Pruning
   • Smart Memoization
   • Spreading Neighborhood Exploration

2 Rule Filtering by Pattern
   • Rule Patterns
   • Hiero Shallow
   • Filtering Translations
   • Rule Patterns Revisited
   • Large Language Models and Evaluation
**Cube Pruning:** extracts efficiently first k-best hyps

- Original CP items X already added! Queue of candidates shrinks $\rightarrow$ Search Errors!
- Spreading neighborhood exploration adds candidates S to the queue
How can we assess the impact of SNE?

We use as a reference TTM, a phrase-based SMT system implemented with Weighted Finite-State Transducers.

TTM Reordering Models: MJO, or an MJ1 (maximum phrase jump of 0 and 1, respectively).

TTM works largely without pruning (even with big models).

HCP can easily emulate TTM MJ0 and MJ1 models.
Table: Hierarchical grammars for MJ1

<table>
<thead>
<tr>
<th>HIERO MJ1</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S \rightarrow \langle S \ X, S \ X \rangle$</td>
</tr>
<tr>
<td>$S \rightarrow \langle X, X \rangle$</td>
</tr>
<tr>
<td>$X \rightarrow \langle V_2 \ V_1, V_1 \ V_2 \rangle$</td>
</tr>
<tr>
<td>$X \rightarrow \langle V, V \rangle$</td>
</tr>
<tr>
<td>$V \rightarrow \langle s, t \rangle$</td>
</tr>
<tr>
<td>$s, t \in T^+$</td>
</tr>
</tbody>
</table>
### Hiero Search Errors III
A study in Phrase-Based Translation

<table>
<thead>
<tr>
<th></th>
<th>Monotone</th>
<th>MJ1+MET</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>BLEU</td>
<td>SE</td>
</tr>
<tr>
<td>TTM</td>
<td>44.7</td>
<td>-</td>
</tr>
<tr>
<td>HCP</td>
<td>44.5</td>
<td>342</td>
</tr>
<tr>
<td>HCP+SNE=20</td>
<td>44.7</td>
<td>77</td>
</tr>
</tbody>
</table>

**Table:** Phrase-based TTM and Hiero performance comparison on Arabic-to-English *mt02-05-tune*. SE is the number of Hiero hypotheses with search errors.
Outline

1. Refinements in the Cube Pruning Decoder
   - Hiero Cube Pruning
   - Smart Memoization
   - Spreading Neighborhood Exploration

2. Rule Filtering by Pattern
   - Rule Patterns
   - Hiero Shallow
   - Filtering Translations
   - Rule Patterns Revisited
   - Large Language Models and Evaluation
Initial Rule Sets Are Really Big

- Initial Rule Extraction: 175 M rules!
- Other approaches:
  - Lopez (2008) enforces rules with minspan of two words (115M)
  - Zollman et al. (2008) enforce mincount: (e.g. 57M mincount=3)
  - Shen et al. (2008) filter target-side rules that are not well-formed dependency trees
  - Chiang (2007) reports experiments with 5.5M
- Are all these rules needed for translation?
Hierarchical rules $X \rightarrow \langle \gamma, \alpha \rangle$: sequences of terminals and non-terminals (elements)

- Source Pattern and Target Pattern: replace every sequence of terminals by a single symbol ‘w’ ($w \in T^+$).
- Each hierarchical rule corresponds to a unique source and target pattern which together define the rule pattern.

- 65 hierarchical rule patterns
Rule Patterns II

Example:
Pattern $\langle wX_1, wX_1w \rangle$:
$\langle w+ qAl X_1, the X_1said \rangle$

Pattern $\langle wX_1w, wX_1 \rangle$:
$\langle fy X_1kAnwn Al>wl, on december X_1 \rangle$

Pattern $\langle wX_1wX_2, wX_1wX_2w \rangle$:
$\langle HI X_1lAzmp X_2, a X_1solution to the X_2crisis \rangle$

Rules can be classed by their number of non-terminals, $N_{nt}$, and their number of elements, $N_e$ (source side).

There are 5 possible classes:
$N_{nt}N_e = 1.2, 1.3, 2.3, 2.4, 2.5.$
## Rule Patterns III

<table>
<thead>
<tr>
<th>Class</th>
<th>Rule Pattern</th>
<th>Types</th>
</tr>
</thead>
<tbody>
<tr>
<td>$N_{nt} N_e$</td>
<td>$\langle \text{source} , \text{target} \rangle$</td>
<td></td>
</tr>
<tr>
<td>1.2</td>
<td>$\langle wX_1 , wX_1 \rangle$, ( \langle wX_1 , wX_1 w \rangle ), ( \langle wX_1 , X_1 w \rangle )</td>
<td>1185028, 153130, 97889</td>
</tr>
<tr>
<td>1.3</td>
<td>$\langle wX_1 w , wX_1 w \rangle$, ( \langle wX_1 w , wX_1 \rangle )</td>
<td>32903522, 989540</td>
</tr>
<tr>
<td>2.3</td>
<td>$\langle X_1 wX_2 , X_1 wX_2 \rangle$, ( \langle X_2 wX_1 , X_1 wX_2 \rangle )</td>
<td>1554656, 39163</td>
</tr>
<tr>
<td>2.4</td>
<td>$\langle X_1 wX_2 w , X_1 wX_2 w \rangle$, ( \langle wX_1 wX_2 , wX_1 wX_2 w \rangle ), ( \langle wX_2 wX_1 , wX_1 wX_2 \rangle )</td>
<td>26053969, 2534510, 349176</td>
</tr>
<tr>
<td>2.5</td>
<td>$\langle wX_1 wX_2 w , wX_1 X_2 w \rangle$, ( \langle wX_1 wX_2 w , X_1 wX_2 w \rangle ), ( \langle wX_2 wX_1 w , X_1 wX_2 w \rangle )</td>
<td>3149516, 2330797, 275810</td>
</tr>
</tbody>
</table>
Towards a more Workable Rule Set I

- Greedy approach to building a rule set:
- Rules belonging to a pattern are added to the rule set guided by the improvements relative to Hiero Monotone
- Certain patterns seem not to contribute to any improvement.
  - No improvement when adding $\langle X_1 w, X_1 w \rangle$ (1.2M)
  - Adding $\langle wX_1, X_1 w \rangle$ (0.01M), provides substantial gains.
  - Situation is analogous two non-terminals ($N_{nt}=2$).
Towards a more Workable Rule Set II

<table>
<thead>
<tr>
<th>Types</th>
<th>Excluded Rules</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>( \langle X_1w,X_1w \rangle, \langle wX_1,wX_1 \rangle )</td>
<td>2332604</td>
</tr>
<tr>
<td>b</td>
<td>( \langle X_1wX_2,* \rangle )</td>
<td>2121594</td>
</tr>
<tr>
<td>c</td>
<td>( \langle X_1wX_2w,X_1wX_2w \rangle, \langle wX_1wX_2,wX_1wX_2 \rangle )</td>
<td>52955792</td>
</tr>
<tr>
<td>d</td>
<td>( \langle wX_1wX_2w,* \rangle )</td>
<td>69437146</td>
</tr>
<tr>
<td>e</td>
<td>( N_{nt}N_e = 1.3 ) ( w ) ( \text{mincount}=5 )</td>
<td>32394578</td>
</tr>
<tr>
<td>f</td>
<td>( N_{nt}N_e = 2.3 ) ( w ) ( \text{mincount}=5 )</td>
<td>166969</td>
</tr>
<tr>
<td>g</td>
<td>( N_{nt}N_e = 2.4 ) ( w ) ( \text{mincount}=10 )</td>
<td>11465410</td>
</tr>
<tr>
<td>h</td>
<td>( N_{nt}N_e = 2.5 ) ( w ) ( \text{mincount}=5 )</td>
<td>688804</td>
</tr>
</tbody>
</table>

Table: Rules excluded from the initial rule set. 171M filtered out, 3.5 hierarchical rules, 4.2 including phrase-based rules.
Outline

1. Refinements in the Cube Pruning Decoder
   - Hiero Cube Pruning
   - Smart Memoization
   - Spreading Neighborhood Exploration

2. Rule Filtering by Pattern
   - Rule Patterns
   - Hiero Shallow
   - Filtering Translations
   - Rule Patterns Revisited
   - Large Language Models and Evaluation
### Hiero Full versus Hiero Shallow I

<table>
<thead>
<tr>
<th>HIERO</th>
<th>HIERO SHALLOW</th>
</tr>
</thead>
<tbody>
<tr>
<td>$X \rightarrow \langle \gamma, \alpha \rangle$</td>
<td>$X \rightarrow \langle \gamma_s, \alpha_s \rangle$</td>
</tr>
<tr>
<td>$\gamma, \alpha \in ({X} \cup T)^+$</td>
<td>$X \rightarrow \langle V, V \rangle$</td>
</tr>
<tr>
<td></td>
<td>$V \rightarrow \langle s, t \rangle$</td>
</tr>
<tr>
<td></td>
<td>$s, t \in T^+$; $\gamma_s, \alpha_s \in ({V} \cup T)^+$</td>
</tr>
</tbody>
</table>

**Table:** Hierarchical grammars, Shallow versus Full
Hiero Full versus Hiero Shallow II

<table>
<thead>
<tr>
<th>System</th>
<th>-tune</th>
<th>-test</th>
</tr>
</thead>
<tbody>
<tr>
<td>HIERO</td>
<td>14.0</td>
<td>52.1</td>
</tr>
<tr>
<td>HIERO - shallow</td>
<td>2.0</td>
<td>52.1</td>
</tr>
</tbody>
</table>

Table: Translation performance and time (in seconds per word) for full vs. shallow Hiero. Arabic-to-English task, kbest=10000, SNE=20
Outline

1. Refinements in the Cube Pruning Decoder
   - Hiero Cube Pruning
   - Smart Memoization
   - Spreading Neighborhood Exploration

2. Rule Filtering by Pattern
   - Rule Patterns
   - Hiero Shallow
   - Filtering Translations
   - Rule Patterns Revisited
   - Large Language Models and Evaluation
Filtering by Number of Translations I

- \( \forall \gamma \not\in T^+ \) filter \( X \rightarrow \langle \gamma, \alpha \rangle \) with the following criteria:
  - Keep the \( \text{NT} \) most frequent \( \alpha \), i.e. each \( \gamma \) is allowed to have at most \( \text{NT} \) rules.
  - Keep the \( \text{NRT} \) most frequent \( \alpha \) with monotonic non-terminals and the \( \text{NRT} \) most frequent \( \alpha \) with reordered non-terminals.
  - Keep the most frequent \( \alpha \) until their aggregated number of counts reaches a certain percentage \( \text{CP} \) of the total counts of \( X \rightarrow \langle \gamma, * \rangle \).
### Filtering by Number of Translations II

<table>
<thead>
<tr>
<th>Filter</th>
<th>Time</th>
<th>Rules</th>
<th>BLEU</th>
<th>BLEU</th>
</tr>
</thead>
<tbody>
<tr>
<td>baseline</td>
<td>2.0</td>
<td>4.20</td>
<td>52.1</td>
<td>51.4</td>
</tr>
<tr>
<td>NT=10</td>
<td>0.8</td>
<td>3.25</td>
<td>52.0</td>
<td>51.3</td>
</tr>
<tr>
<td>NT=15</td>
<td>0.8</td>
<td>3.43</td>
<td>52.0</td>
<td>51.3</td>
</tr>
<tr>
<td>NT=20</td>
<td>0.8</td>
<td>3.56</td>
<td>52.1</td>
<td>51.4</td>
</tr>
<tr>
<td>NRT=10</td>
<td>0.9</td>
<td>3.29</td>
<td>52.0</td>
<td>51.3</td>
</tr>
<tr>
<td>NRT=15</td>
<td>1.0</td>
<td>3.48</td>
<td>52.0</td>
<td>51.4</td>
</tr>
<tr>
<td>NRT=20</td>
<td>1.0</td>
<td>3.59</td>
<td>52.1</td>
<td>51.4</td>
</tr>
<tr>
<td>CP=50</td>
<td>0.7</td>
<td>2.56</td>
<td>51.4</td>
<td>50.9</td>
</tr>
<tr>
<td>CP=90</td>
<td>1.0</td>
<td>3.60</td>
<td>52.0</td>
<td>51.3</td>
</tr>
</tbody>
</table>

**Table:** Impact of general rule filters on translation (IBM BLEU), time (in seconds per word) and number of rules (in millions).
Outline

1. Refinements in the Cube Pruning Decoder
   - Hiero Cube Pruning
   - Smart Memoization
   - Spreading Neighborhood Exploration

2. Rule Filtering by Pattern
   - Rule Patterns
   - Hiero Shallow
   - Filtering Translations
   - Rule Patterns Revisited
   - Large Language Models and Evaluation
Revisiting Pattern Filtering Strategies I

- As many decisions were based on the initial greedy approach, we revisit our strategy.
- Different (class) mincount filterings.
- Rule pattern filterings: Reintroduce different monotone patterns.
Revisiting Pattern Filtering Strategies II

<table>
<thead>
<tr>
<th>mto2-05-</th>
<th>-tune</th>
<th>-test</th>
</tr>
</thead>
<tbody>
<tr>
<td>$N_{nt} \cdot N_e$</td>
<td>Filter</td>
<td>Time</td>
</tr>
<tr>
<td>baseline, NRT=20</td>
<td></td>
<td>1.0</td>
</tr>
<tr>
<td>2.3</td>
<td>+monotone</td>
<td>1.1</td>
</tr>
<tr>
<td>2.4</td>
<td>+monotone</td>
<td>2.0</td>
</tr>
<tr>
<td>2.5</td>
<td>+monotone</td>
<td>1.8</td>
</tr>
</tbody>
</table>

- Reintroducing monotonic rules degrades performance, substantial increase of n of rules.
# Revisiting Pattern Filtering Strategies III

<table>
<thead>
<tr>
<th>$N_{nt} \cdot N_e$</th>
<th>Filter</th>
<th>Time</th>
<th>Rules</th>
<th>BLEU</th>
<th>BLEU</th>
</tr>
</thead>
<tbody>
<tr>
<td>baseline $NRT=20$</td>
<td></td>
<td>1.0</td>
<td>3.59</td>
<td>52.1</td>
<td>51.4</td>
</tr>
<tr>
<td>1.3</td>
<td>mincount=3</td>
<td>1.0</td>
<td>5.61</td>
<td>52.1</td>
<td>51.3</td>
</tr>
<tr>
<td>2.3</td>
<td>mincount=1</td>
<td>1.2</td>
<td>3.70</td>
<td>52.1</td>
<td>51.4</td>
</tr>
<tr>
<td>2.4</td>
<td>mincount=5</td>
<td>1.8</td>
<td>4.62</td>
<td>52.0</td>
<td>51.3</td>
</tr>
<tr>
<td>2.4</td>
<td>mincount=15</td>
<td>1.0</td>
<td>3.37</td>
<td>52.0</td>
<td>51.4</td>
</tr>
<tr>
<td>2.5</td>
<td>mincount=1</td>
<td>1.1</td>
<td>4.27</td>
<td>52.2</td>
<td><strong>51.5</strong></td>
</tr>
<tr>
<td>1.2</td>
<td>mincount=5</td>
<td>1.0</td>
<td>3.51</td>
<td>51.8</td>
<td>51.3</td>
</tr>
<tr>
<td>1.2</td>
<td>mincount=10</td>
<td>1.0</td>
<td>3.50</td>
<td>51.7</td>
<td>51.2</td>
</tr>
</tbody>
</table>
Outline

1. Refinements in the Cube Pruning Decoder
   - Hiero Cube Pruning
   - Smart Memoization
   - Spreading Neighborhood Exploration

2. Rule Filtering by Pattern
   - Rule Patterns
   - Hiero Shallow
   - Filtering Translations
   - Rule Patterns Revisited
   - Large Language Models and Evaluation
Rescoring steps:

- *Large-LM rescoring* of 10000-best list with 5-gram language models,
- *Minimum Bayes Risk (MBR)*. Rescore 1000-best hyps

<table>
<thead>
<tr>
<th></th>
<th>mt06-nist-nw</th>
<th>mt06-nist-ng</th>
<th>mt08</th>
</tr>
</thead>
<tbody>
<tr>
<td>HCP+MET</td>
<td>48.4 / 43.6</td>
<td>35.3 / 53.2</td>
<td>42.5 / 48.6</td>
</tr>
<tr>
<td>+rescoring</td>
<td>49.4 / 42.9</td>
<td>36.6 / 53.5</td>
<td>43.4 / 48.1</td>
</tr>
</tbody>
</table>

Table: Arabic-to-English translation results (lower-cased IBM BLEU / TER)

- Mixed case NIST BLEU for *mt08* is 42.5
Summary

- Smart memoization and spreading neighborhood exploration reduce memory consumption and Hiero search errors.
- For Arabic-to-English, Shallow hierarchical decoding is as good as fully hierarchical decoding (and much faster!)
- Filtering Rules by Translations further increases speed with no cost in scores
- For hierarchical rules grouped in classes and patterns:
  - Certain patterns are of much greater value in translation than others
  - Separate minimum count filtering should be applied

Iglesias, de Gispert, R. Banga, Byrne
Thank you!

For further reading, check out NAACL2009 paper: "Hierarchical Phrase-Based Translation with Weighted Finite State Transducers"