

Rule Filtering by Pattern for Efficient Hierarchical Translation

Gonzalo Iglesias¹ Adrià de Gispert²
Eduardo R. Banga¹ William Byrne²

¹Department of Signal Processing and Communications
University of Vigo, Spain

²Department of Engineering.
University of Cambridge, U.K.

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

Outline

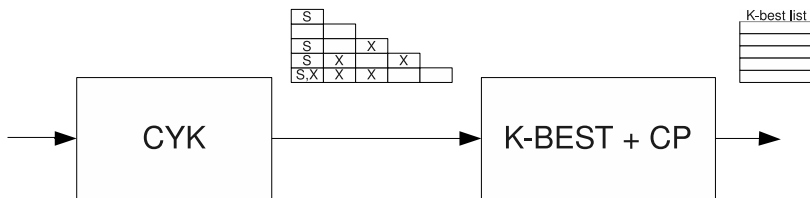
- 1 Refinements in the Cube Pruning Decoder
 - Hiero Cube Pruning
 - Smart Memoization
 - Spreading Neighborhood Exploration
- 2 Rule Filtering by Pattern
 - Rule Patterns
 - Hiero Shallow
 - Filtering Translations
 - Rule Patterns Revisited
 - Large Language Models and Evaluation

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!



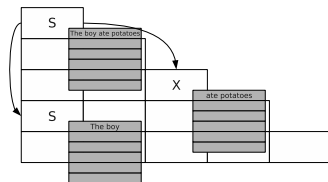
- CYK: source side, hypotheses recombination, no pruning
- k-best algorithm: uses cube pruning with LM costs to extract efficiently k-best lists
- NIST 2008 Arabic-to-English task, k-best=10000

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

Smart Memoization

- 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)



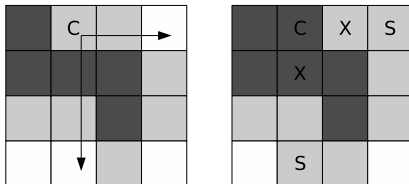
Smart Memoization

- 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%

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

Spreading Neighborhood Exploration I



- Cube Pruning: extracts efficiently first k-best hyps
- Original CP items X already added! Queue of candidates shrinks → Search Errors!
- Spreading neighborhood exploration adds candidates S to the queue

Hiero Search Errors I

A study in Phrase-Based Translation

- 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

Hiero Search Errors II

A study in Phrase-Based Translation

HIERO MJ1
$S \rightarrow \langle S X, S X \rangle$
$S \rightarrow \langle X, X \rangle$
$X \rightarrow \langle V_2 V_1, V_1 V_2 \rangle$
$X \rightarrow \langle V, V \rangle$
$V \rightarrow \langle s, t \rangle$
$s, t \in \mathbf{T}^+$

Table: Hierarchical grammars for MJ1

Hiero Search Errors III

A study in Phrase-Based Translation

	Monotone		MJ1+MET	
	BLEU	SE	BLEU	SE
TTM	44.7	-	49.1	-
HCP	44.5	342	48.4	822
HCP+SNE=20	44.7	77	48.9	360

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?

Rule Patterns I

- 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' ($w \in \mathbf{T}^+$).
- 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 $\langle wX_1, wX_1w \rangle$:
 $\langle w+ qAl X_1, the X_1 said \rangle$
Pattern $\langle wX_1w, wX_1 \rangle$:
 $\langle fy X_1 kAnwn Al \rangle wI, on december X_1 \rangle$
Pattern $\langle wX_1wX_2, wX_1wX_2w \rangle$:
 $\langle HI X_1 lAzmp X_2, a X_1 solution to the X_2 crisis \rangle$
- 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:
 $N_{nt} \cdot N_e = 1.2, 1.3, 2.3, 2.4, 2.5.$

Rule Patterns III

Class $N_{nt} \cdot N_e$	Rule Pattern $\langle \text{source}, \text{target} \rangle$	Types
1.2	$\langle wX_1, wX_1 \rangle$	1185028
	$\langle wX_1, wX_1w \rangle$	153130
	$\langle wX_1, X_1w \rangle$	97889
1.3	$\langle wX_1w, wX_1w \rangle$	32903522
	$\langle wX_1w, wX_1 \rangle$	989540
2.3	$\langle X_1wX_2, X_1wX_2 \rangle$	1554656
	$\langle X_2wX_1, X_1wX_2 \rangle$	39163
2.4	$\langle X_1wX_2w, X_1wX_2w \rangle$	26053969
	$\langle wX_1wX_2, wX_1wX_2w \rangle$	2534510
	$\langle wX_2wX_1, wX_1wX_2 \rangle$	349176
2.5	$\langle wX_1wX_2w, wX_1X_2w \rangle$	3149516
	$\langle wX_1wX_2w, X_1wX_2w \rangle$	2330797
	$\langle wX_2wX_1w, wX_1wX_2w \rangle$	275810

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 $\langle w X_1, X_1 w \rangle$ (0.01M), provides substantial gains.
 - Situation is analogous two non-terminals ($N_{nt}=2$).

Towards a more Workable Rule Set II

	Excluded Rules	Types
a	$\langle X_1 w, X_1 w \rangle, \langle w X_1, w X_1 \rangle$	2332604
b	$\langle X_1 w X_2, * \rangle$	2121594
c	$\langle X_1 w X_2 w, X_1 w X_2 w \rangle, \langle w X_1 w X_2, w X_1 w X_2 \rangle$	52955792
d	$\langle w X_1 w X_2 w, * \rangle$	69437146
e	$N_{nt}.N_e = 1.3$ w mincount=5	32394578
f	$N_{nt}.N_e = 2.3$ w mincount=5	166969
g	$N_{nt}.N_e = 2.4$ w mincount=10	11465410
h	$N_{nt}.N_e = 2.5$ w mincount=5	688804

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

HIERO	HIERO SHALLOW
$X \rightarrow \langle \gamma, \alpha \rangle$ $\gamma, \alpha \in (\{X\} \cup \mathbf{T})^+$	$X \rightarrow \langle \gamma_s, \alpha_s \rangle$ $X \rightarrow \langle V, V \rangle$ $V \rightarrow \langle s, t \rangle$ $s, t \in \mathbf{T}^+; \gamma_s, \alpha_s \in (\{V\} \cup \mathbf{T})^+$

Table: Hierarchical grammars, Shallow versus Full

Hiero Full versus Hiero Shallow II

<i>mt02-05-</i>	<i>-tune</i>		<i>-test</i>
System	Time	BLEU	BLEU
HIERO	14.0	52.1	51.5
HIERO - shallow	2.0	52.1	51.4

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

- $\forall \gamma \notin \mathbf{T}^+$ filter $X \rightarrow \langle \gamma, \alpha \rangle$ with the following criteria:
 - Keep the **NT** most frequent α , i.e. each γ is allowed to have at most **NT** rules.
 - Keep the **NRT** most frequent α with monotonic non-terminals and the **NRT** most frequent α with reordered non-terminals.
 - Keep the most frequent α until their aggregated number of counts reaches a certain percentage **CP** of the total counts of $X \rightarrow \langle \gamma, * \rangle$.

Filtering by Number of Translations II

<i>mt02-05-</i>	<i>-tune</i>			<i>-test</i>
Filter	Time	Rules	BLEU	BLEU
baseline	2.0	4.20	52.1	51.4
NT=10	0.8	3.25	52.0	51.3
NT=15	0.8	3.43	52.0	51.3
NT=20	0.8	3.56	52.1	51.4
NRT=10	0.9	3.29	52.0	51.3
NRT=15	1.0	3.48	52.0	51.4
NRT=20	1.0	3.59	52.1	51.4
CP=50	0.7	2.56	51.4	50.9
CP=90	1.0	3.60	52.0	51.3

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

<i>mt02-05-</i>		<i>-tune</i>			<i>-test</i>
$N_{nt} \cdot N_e$	Filter	Time	Rules	BLEU	BLEU
baseline NRT=20		1.0	3.59	52.1	51.4
2.3	+monotone	1.1	4.08	51.5	51.1
2.4	+monotone	2.0	11.52	51.6	51.0
2.5	+monotone	1.8	6.66	51.7	51.2

- Reintroducing monotonic rules degrades performance, substantial increase of n of rules.

Revisiting Pattern Filtering Strategies III

<i>mt02-05-</i>		<i>-tune</i>			<i>-test</i>
$N_{nt} \cdot N_e$	Filter	Time	Rules	BLEU	BLEU
	baseline NRT=20	1.0	3.59	52.1	51.4
1.3	mincount=3	1.0	5.61	52.1	51.3
2.3	mincount=1	1.2	3.70	52.1	51.4
2.4	mincount=5	1.8	4.62	52.0	51.3
2.4	mincount=15	1.0	3.37	52.0	51.4
2.5	mincount=1	1.1	4.27	52.2	51.5
1.2	mincount=5	1.0	3.51	51.8	51.3
1.2	mincount=10	1.0	3.50	51.7	51.2

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

Large Language Models and Evaluation I

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

	<i>mt06-nist-nw</i>	<i>mt06-nist-ng</i>	<i>mt08</i>
HCP+MET	48.4 / 43.6	35.3 / 53.2	42.5 / 48.6
+rescoring	49.4 / 42.9	36.6 / 53.5	43.4 / 48.1

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

Thank you!

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