Elastic-substitution decoding for Hierarchical SMT: efficiency, richer search and double labels
(Presentation MT Summit, Nagoya, Japan)

Gideon Maillette de Buy Wenniger¹, Khalil Sima’an², Andy Way¹

gemdbw AT gmail.com  k.simaan AT uva.nl
http://computing.dcu.ie/~gmdbwenniger/

¹ADAPT Center, School of Computing
Dublin City University, Ireland

²Statistical Language Processing and Learning Lab
Institute for Logic Language and Computation
University of Amsterdam, the Netherlands
Overview

- Hierarchical statistical machine translation (SMT) has been effective
- Labeled Hierarchical SMT even more so
- Chiang (2010) introduced soft-matching to overcome problems with strict-enforced label-matching constraints

In this work:

- Efficient soft-matching: howto
- More evidence superiority of soft-matching over strict-matching
- Better search during decoding
  - Ensure label variety → ensure exploration of matching substitutions
- Combining multiple labels
Part 1: Hierarchical Statistical Machine Translation (Hiero)
What is Hiero?
What is Hiero?

ONE MAN WILL CHALLENGE AN EMPIRE

QUENTIN TARANTINO PRESENTS

JET LI

TONE LEUNG
MAGGIE ZHANG
CHEN Zuoyong
DONNIE YEN

DIRECTED BY ACADEMY-AWARD NOMINEE ZHANG YIMO FROM THE PRODUCERS OF "CROUCHING TIGER, HIDDEN DRAGON"
What is Hiero?

Eehu ... no. Let's try again.
(Hierarchical) Statistical Machine Translation - Basics

Approximating best translation by best derivation

$$\arg \max_t P(t \mid s) = \arg \max_{d \in G} \sum_{d \in G} P(t, d \mid s)$$  \hspace{1cm} (1)

$$\approx \arg \max_{d \in G} P(t, d \mid s)$$  \hspace{1cm} (2)

Log-linear Model:

$$\arg \max_{d \in G} P(t, d \mid s) \approx \arg \max_{d \in G} \sum_{i=1}^{\mid \phi(d) \mid} \lambda_i \times \phi_i$$  \hspace{1cm} (3)
From SMT to Hierarchical SMT

- Generalize phrase pairs by introducing phrases with variables
  - Rule-table becomes synchronous context-free grammar
- Decoding: search for best synchronous derivation of the input
- Approximate intersection with the language model: cube pruning
Types of Hiero Rules

\[ X \rightarrow \langle \alpha, \gamma \rangle \]  
Phrase Pair

\[ X \rightarrow \langle \alpha X_1 \beta, \delta X_1 \zeta \rangle \]  
One gap rule

\[ X \rightarrow \langle \alpha X_1 \beta X_2 \gamma, \delta X_1 \zeta X_2 \eta \rangle \]  
Two gaps monotone rule

\[ X \rightarrow \langle \alpha X_1 \beta X_2 \gamma, \delta X_2 \zeta X_1 \eta \rangle \]  
Two gaps inverted rule
Part 2: Labeling Hiero
Overview used labels

1. Syntax-augmented machine translation labels (SAMT)
2. Pos-tags from phrase boundaries
3. Bilingual Phrase Reordering labels
Syntax-Augmented Machine Translation (SAMT)

1. label constituent phrases only
   - Risk coverage loss in strictly syntactic systems
     “Re-structuring, Re-labeling, and Re-aligning for Syntax-Based Machine Translation” (Wang et.al, 2010)

2. add syntax **without coverage loss** w.r.t Hiero
   relaxed syntactic labels akin to Combinatorial Categorial Grammar

   \[ C : NP \rightarrow \text{the great wall} \]
   \[ C1+C2 : NP+VB \rightarrow \text{she}+\text{went} \]
   \[ C1/C2 : NP/NN \rightarrow \text{the great (/wall)} \]
   \[ C1\backslash C2 : DT\backslash NP \rightarrow (\text{the}\backslash) \text{great wall} \]
   default : FAIL
Example

Which label for:
- we / wir (NP:PRP)
- do not know / ...
- is happening . / ...
- do ... happening . / ...
- we do not / ...

G. Wenniger, K. Sima’an, Andy Way
Efficient Elastic-substitution decoding
21 September, 2017
Example

```
{1} {2,4} {3} {4} {5} {6,7} {8}

wir wissen nicht, was passiert.

We do not know what is happening.
```

**Which label for:**
- NP:PRP → *we / wir*
- VP/SBAR → *do not know*
- S:VP+. → *is happening .*
- VP+. → *do ... happening .*
- FAIL → *we do not*
Phrase-boundary labels

- Pos-tag the target-side of the corpus
- Concatenate the pos-tags for the phrase-boundary words to form labels
- Similar in spirit to SAMT
- In contrast to SAMT, yields a real (i.e. not “FAIL”) label for any span
Bilingual Phrase Reordering label categories

- Phrase-Centric
- Parent-Relative
Phrase-centric reordering labels

- Complexity relation between base phrase and direct children, when decomposing the phrase, determines label
- Five cases distinguished, ordered by increasing complexity

- Monotonic
  - Sample: this is an important matter
- Inversion
  - Sample: we all agree on this
- Permutation
  - Sample: i want to stress two points
- Complex
  - Sample: we owe this to our citizens
- Atomic
  - Sample: it would be possible
Known labels from ITG and Phrase pair Theory
Monotonic

- **Monotonic**: If the alignment can be split into two monotonically ordered parts.
Inverted

- **Inverted**: If the alignment can be split into two inverted parts.
**Atomic**

- **Atomic**: If the alignment does not allow the existence of smaller (child) phrase pairs.

![Diagram of atomic alignment examples](image_url)
New labels based on HATs
Permutation

- *Permutation*: If the alignment can be factored as a permutation of more than 3 parts.

Monotonic

Inversion

Permutation

1. I want
2. to
3. stress
4. two
5. points

Monotonic

Inversion

Permutation

1. auf
2. zwei
3. punkte
4. möchte
5. ich
6. hinweisen

Monotonic

Inversion

Permutation

1. G. Wenniger, K. Sima’an, Andy Way
2. Efficient Elastic-substitution decoding
3. 21 September, 2017
Complex

- **Complex**: No alignment factorization as a permutation of parts, but smaller phrase pair is contained (i.e., it is composite).
Phrase-Centric labeled derivation

\[ S \]

\[ S \]
Phrase-Centric labeled derivation

\[ s \xrightarrow{\text{uw}} \text{COMPLEX} \]

\[ s \xrightarrow{\text{uw}} \text{COMPLEX} \]
Phrase-Centric labeled derivation

tailor accordingly

darauf ausrichten

INVERTED MONO COMPLEX
Phrase-Centric labeled derivation
Phrase-Centric labeled derivation

\[ \text{COMPLEX} \rightarrow \text{INVERTED} \rightarrow \text{ATOMIC} \rightarrow \text{wir} \rightarrow \text{müssen} \rightarrow \text{darauf} \rightarrow \text{should} \rightarrow \text{tailor} \rightarrow \text{accordingly} \]
Phrase-Centric labeled derivation

Accordingly, our tailor should accordingly ausrichten.

COMPLEX

INVERTED

should

we

müssen

darauf

ATOMIC

unsere

ATOMIC

ATOMIC

ATOMIC

wir

unsere

INVERTED

MÖNO

OUR

s

S

G. Wenniger, K. Sima’an, Andy Way
Efficient Elastic-substitution decoding
21 September, 2017
Phrase-Centric labeled derivation

G. Wenniger, K. Sima’an, Andy Way
Efficient Elastic-substitution decoding
21 September, 2017
Parent-Relative reordering labels

- Describe type of reordering relative to embedding “parent” phrase

- First-order view on reordering

- 9 distinct labels indicating different forms of monontone, inverted and discontinuous reordering relations
Part 3: Soft-matching for labeled Hiero
Motivation soft-matching

- With strict-matching, many labeled variants of the same rule, and hence Hiero derivation exist
  - Further increase spurious ambiguity

- Strict-matching blocks prospective valid translations

- Quality of labels not always assured

- Some labels may be partly interchangeable, especially for Heuristic labeling schemes
Two variants of “label relaxation”

1. Preference Grammars

Preference Grammars (Venugopal et.al, 2009)

- “Pack” the distinct labeled versions of a Hiero rule type
- Associate with every Hiero rule type a distribution over different labelings
- Use dynamic programming to incrementally produce a kind of in-product between label distributions of substituting and substituted-to rules
- Loosely: approximates summation derivations over alternatively labeled-versions same Hiero rule type
Soft matching

- Any label may match any other label
- Use per Hiero rule typ only single most frequent version amongst all variants
- Add features that mark specific label substitutions
  - Allows tuning to learn preference over substitutions
How to make it fast?

Requirements:
- Quickly retrieve all rules, while ignoring the labels
- Still distinguish between glue nonterminals ("GOAL") and all other nonterminals

The solution . . .

Trie it!
What are tries?

- Tries are also known as Prefix-trees.

- Tries efficiently store \( \langle \text{key-list}, \text{value} \rangle \) pairs, when there are many common prefixes amongst the key-lists.

- This is exactly the case of Hiero grammars.

- While a single flat hashtables could also be used, that would typically require more memory.
Unlabeled Rule Trie

ROOT

[X,2]

[X,1]

marche

[X,1] joue

[X,1] lentement

[X,1] vite

[X,1] X \rightarrow \langle \text{[X,1] walks slowly} \rangle

[X,1] X \rightarrow \langle \text{[X,1] strolls} \rangle

[X,1] X \rightarrow \langle \text{[X,1] walks calmly} \rangle
Strict-Matching Rule Trie

```
ROOT
  |--- [N,1] marche
  |     |--- lentement
  |     |     |--- [ADJ,2]
  |     |     |     |--- joue
  |     |--- [NP,1] marche
  |     |     |--- lentement
  |     |     |     |--- vite
  |     |--- [VP,2] marche
  |     |     |--- [GOAL,1] le

S → ⟨ [NP,1] marche lentement, [NP,1] walks slowly ⟩
S → ⟨ [NP,1] marche lentement, [NP,1] strolls ⟩
S → ⟨ [N,1] marche lentement, [N,1] walks calmly ⟩
```
Observations Trie for soft-matching

- Labels are removed in the internal nodes
- But the GOAL label is maintained
- Labels are kept at the final rule-lists in the leaf nodes

**Crucial part:** retain labels only in the final rule lists, not in the index structure itself!
Recap and outlook

Introduced so far:
- Hiero
- Hiero labeling schemes
- Motivation soft-matching
- Implementation soft-matching

Now:
- Experiments strict-matching against soft-matching for the discussed labels
Experimental Setup

- Chinese-English translation experiments

Data properties

<table>
<thead>
<tr>
<th>dataset type</th>
<th>size</th>
<th>data origin</th>
</tr>
</thead>
<tbody>
<tr>
<td>train</td>
<td>7.34M</td>
<td>MultiUn + Hong Kong Parallel Text</td>
</tr>
<tr>
<td>dev</td>
<td>2K</td>
<td>Multiple Translation Chinese</td>
</tr>
<tr>
<td>test</td>
<td>2K</td>
<td>Multiple Translation Chinese</td>
</tr>
</tbody>
</table>

- Max sentence length 40

Language model

- 4-gram language model
- Kneser-Ney discounting
Results for labeled systems with strict or soft label matching

<table>
<thead>
<tr>
<th>System Name</th>
<th>BLEU ↑</th>
<th>METEOR ↑</th>
<th>BEER ↑</th>
<th>TER ↓</th>
<th>KRS ↑</th>
<th>Length</th>
<th>CPU time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hiero (H)</td>
<td>31.63</td>
<td>30.56</td>
<td>13.15</td>
<td>59.28</td>
<td>58.03</td>
<td>97.15</td>
<td>3.34</td>
</tr>
<tr>
<td>Hiero-0&lt;sup&gt;th&lt;/sup&gt;-Str</td>
<td>31.90 ▲H</td>
<td>30.79 ▲H</td>
<td>13.45</td>
<td>60.11 ▼H</td>
<td>59.68 ▲H</td>
<td>98.65 ■H</td>
<td>2.87</td>
</tr>
<tr>
<td>Hiero-1&lt;sup&gt;st&lt;/sup&gt;-Str</td>
<td>31.77 ▲H</td>
<td>30.62</td>
<td>13.20</td>
<td>60.13 ▼H</td>
<td>59.89 ▲H</td>
<td>98.47 ■H</td>
<td>4.63</td>
</tr>
<tr>
<td>Hiero-0&lt;sup&gt;th&lt;/sup&gt;-1&lt;sup&gt;st&lt;/sup&gt;-Str</td>
<td>32.35▲H ▲S</td>
<td>30.98▲H ▲S</td>
<td>13.75▲H ▲S</td>
<td>60.26▼H</td>
<td>60.01▲H</td>
<td>99.11■H ▲S</td>
<td>8.45</td>
</tr>
<tr>
<td>SAMT-Str</td>
<td>31.87 △H</td>
<td>30.61</td>
<td>13.38</td>
<td>59.97 ▼H</td>
<td>59.94 ▲H</td>
<td>98.46 ■H</td>
<td>25.59</td>
</tr>
<tr>
<td>SAMT</td>
<td>32.40▲H ▲S</td>
<td>31.20▲H ▲S</td>
<td>14.01▲H ▲S</td>
<td>60.19▼H ▼S</td>
<td>60.38▲H △S</td>
<td>99.37■H ■H</td>
<td>8.09</td>
</tr>
<tr>
<td>BoundaryTag-Str</td>
<td>32.26▲H</td>
<td>30.94▲H</td>
<td>13.91▲H</td>
<td>60.20▼H</td>
<td>58.78▲H</td>
<td>98.98▲H</td>
<td>29.29</td>
</tr>
<tr>
<td>BoundaryTag</td>
<td>32.77▲H ▲S</td>
<td>31.27▲H ▲S</td>
<td>14.17▲H ▲S</td>
<td>60.15▼H</td>
<td>60.83▲H ▲S</td>
<td>99.72■H ■H</td>
<td>8.60</td>
</tr>
</tbody>
</table>
Part 4: Beyond basic Soft-matching: extended search, shuffling and double labels
Extending the search

1. Increase the pop-limit (amount of brute-force search)

2. *Explore all rule labelings*: during cube pruning initialization we allow all matching distinctly labeled versions of the same rule source side to be evaluated

3. Combine both (1) and (2)

Next: important detail - shuffling
Shuffling: the origin

“Shuffling, son. I’m talking shuffling. I want you to launder and shuffle. That’s why I called on you. If it was simple brainwash laundry, there wouldn’t have been any need t’call you.”

“I don’t get it,” I said, recrossing my legs. “How do you know about shuffling? That’s classified information. No outsider’s supposed to know about it.”

“Well, I do. I’ve got a pretty open pipeline to the top of the system.”

“Okay, then run this thorough your pipeline. Shuffling procedures are completely frozen at this time. Don’t ask me why. Obviously some kind of trouble. Whatever the case, shuffling is now prohibited.”
Shuffling in this work

*Shuffling* is the **randomization** of the order of the rules **before** they are evaluated and sorted by score in the cube-pruning queue.
The effect of shuffling, an example

- Translating the source phrase “Elle marche lentement”
  - Evaluation order and final order without shuffling.

Evaluation order without shuffling:

1. S -> [NP,1] marche lentement, [NP,1] walks slowly (A)
2. S -> [NP,1] marche lentement, [NP,1] strolls (B)
3. S -> [N,1] marche lentement, [N,1] walks calmly (C)
4. S -> Elle marche [ADV,1], She walks [ADV,1] (D)

Final order after scoring:

1. S -> [NP,1] marche lentement, [NP,1] walks slowly $$$
   Elle marche lentement / She walks slowly $$$ -4.0 (A)
2. S -> [NP,1] marche lentement, [NP,1] strolls $$$
   Elle marche lentement / She strolls $$$ -5.0 (B)
   Elle marche lentement / She walks calmly $$$ -5.0 (C)
4. S -> Elle marche [ADV,1], She walks [ADV,1]
   Elle marche lentement / She walks leisurely $$$ -5.0 (D)
The effect of shuffling, an example

- Translating the source phrase “Elle marche lentement”
  - Evaluation order and final order with shuffling.

Evaluation order with shuffling (i.e. random):

1. \( S \rightarrow \text{Elle \ marche [ADV,1], She \ walks [ADV,1]} \) (D)
2. \( S \rightarrow [NP,1] \ marche \ lentement, \ [NP,1] \ walks \ slowly \) (A)
3. \( S \rightarrow [N,1] \ marche \ lentement, \ [N,1] \ walks \ calmly \) (C)
4. \( S \rightarrow [NP,1] \ marche \ lentement, \ [NP,1] \ strolls \) (B)

Final order after scoring:

1. \( S \rightarrow [NP,1] \ marche \ lentement, \ [NP,1] \ walks \ slowly \ ||| \)
   Elle marche lentement / She walks slowly ||| -4.0 (A)

2. \( S \rightarrow \text{Elle \ marche [ADV,1], She \ walks [ADV,1]} \)
   Elle marche lentement / She walks leisurely ||| -5.0 (D)

3. \( S \rightarrow [N,1] \ marche \ lentement, \ [N,1] \ walks \ calmly \ ||| \)
   Elle marche lentement / She walks calmly ||| -5.0 (C)

4. \( S \rightarrow [NP,1] \ marche \ lentement, \ [NP,1] \ strolls \ ||| \)
   Elle marche lentement / She strolls ||| -5.0 (B)
Shuffling: observations

- Shuffling only affects relative order rules that tie for same score

- Avoid “lazy tuning”: shuffling can encourage more robust tuning, by eliminating repeated order for hypotheses with same score
### Results extended search experiments

<table>
<thead>
<tr>
<th>System Name</th>
<th>Pop-limit</th>
<th>Explore all rule labelings</th>
<th>BLEU ↑</th>
<th>METEOR ↑</th>
<th>BEER ↑</th>
<th>TER ↓</th>
<th>KRS ↑</th>
<th>Length</th>
<th>CPU time</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Hiero</strong></td>
<td>1000</td>
<td>--</td>
<td>31.63</td>
<td>30.56</td>
<td>13.15</td>
<td>59.28</td>
<td>58.03</td>
<td>97.15</td>
<td>3.34</td>
</tr>
<tr>
<td><strong>Hiero 0&lt;sup&gt;th&lt;/sup&gt;</strong></td>
<td>1000</td>
<td>NO</td>
<td>32.03↑H</td>
<td>30.70↑H</td>
<td>13.42↑H</td>
<td>59.58↓H</td>
<td>58.87↑H</td>
<td>97.87↑H</td>
<td>8.99</td>
</tr>
<tr>
<td><strong>Hiero 0&lt;sup&gt;th&lt;/sup&gt;</strong></td>
<td>2000</td>
<td>NO</td>
<td>32.24↑HIB</td>
<td>30.66↑H</td>
<td>13.41</td>
<td>59.01↑HIB</td>
<td>58.59△H</td>
<td>97.42↑HIB</td>
<td>16.13</td>
</tr>
<tr>
<td><strong>Hiero 0&lt;sup&gt;th&lt;/sup&gt;</strong></td>
<td>4000</td>
<td>NO</td>
<td>32.30↑HIB</td>
<td>30.85↑HIB</td>
<td>13.71↑HIB</td>
<td>59.56↑H</td>
<td>98.19↑HIB</td>
<td>28.27</td>
<td></td>
</tr>
<tr>
<td><strong>Hiero 0&lt;sup&gt;th&lt;/sup&gt;-Sh</strong></td>
<td>1000</td>
<td>YES</td>
<td>32.18↑H</td>
<td>30.77↑HIB</td>
<td>13.55↑HIB</td>
<td>59.23↑HIB</td>
<td>58.74↑H</td>
<td>97.69↑HIB</td>
<td>9.32</td>
</tr>
<tr>
<td><strong>Hiero 0&lt;sup&gt;th&lt;/sup&gt;-Sh</strong></td>
<td>2000</td>
<td>YES</td>
<td>32.25↑HIB</td>
<td>30.80↑HIB</td>
<td>13.60↑HIB</td>
<td>59.44↑H</td>
<td>97.99↑H</td>
<td>10.14</td>
<td></td>
</tr>
<tr>
<td><strong>Hiero 0&lt;sup&gt;th&lt;/sup&gt;-Sh</strong></td>
<td>4000</td>
<td>YES</td>
<td>32.47↑HIB</td>
<td>30.87↑HIB</td>
<td>13.69↑HIB</td>
<td>59.51↑H</td>
<td>98.27↑HIB</td>
<td>16.59</td>
<td></td>
</tr>
<tr>
<td><strong>Hiero 1&lt;sup&gt;st&lt;/sup&gt; (B&lt;sub&gt;1&lt;/sub&gt;)</strong></td>
<td>1000</td>
<td>NO</td>
<td>32.35↑H</td>
<td>30.98↑H</td>
<td>13.75↑H</td>
<td>60.26↓H</td>
<td>60.01↑H</td>
<td>99.11↑H</td>
<td>8.45</td>
</tr>
<tr>
<td><strong>Hiero 1&lt;sup&gt;st&lt;/sup&gt;</strong></td>
<td>2000</td>
<td>NO</td>
<td>32.40↑H</td>
<td>31.00↑H</td>
<td>13.74↑H</td>
<td>60.36↓H</td>
<td>59.93↑H</td>
<td>99.15↑H</td>
<td>15.40</td>
</tr>
<tr>
<td><strong>Hiero 1&lt;sup&gt;st&lt;/sup&gt;</strong></td>
<td>4000</td>
<td>NO</td>
<td>32.49↑H</td>
<td>30.95↑H</td>
<td>13.75↑H</td>
<td>60.29↓H</td>
<td>60.00↑H</td>
<td>99.15↑H</td>
<td>28.84</td>
</tr>
<tr>
<td><strong>Hiero 1&lt;sup&gt;st&lt;/sup&gt;-Sh</strong></td>
<td>1000</td>
<td>YES</td>
<td>32.50↑HIB</td>
<td>31.03↑HIB</td>
<td>13.80↑H</td>
<td>60.14↓H</td>
<td>59.77↑H</td>
<td>99.05↑H</td>
<td>9.55</td>
</tr>
<tr>
<td><strong>Hiero 1&lt;sup&gt;st&lt;/sup&gt;-Sh</strong></td>
<td>1000</td>
<td>YES</td>
<td>32.66↑HIB</td>
<td>31.05↑HIB</td>
<td>13.88↑H</td>
<td>60.01↓HIB</td>
<td>59.99↑H</td>
<td>99.14↑H</td>
<td>10.16</td>
</tr>
<tr>
<td><strong>Hiero 1&lt;sup&gt;st&lt;/sup&gt;-Sh</strong></td>
<td>2000</td>
<td>YES</td>
<td>32.62↑HIB</td>
<td>30.90↑HIB</td>
<td>13.70↑H</td>
<td>59.72↓HIB</td>
<td>59.41↑HIB</td>
<td>98.60↑HIB</td>
<td>21.79</td>
</tr>
<tr>
<td><strong>Hiero 1&lt;sup&gt;st&lt;/sup&gt;-Sh</strong></td>
<td>4000</td>
<td>YES</td>
<td>32.70↑HIB</td>
<td>30.99↑H</td>
<td>13.83↑H</td>
<td>60.22↓H</td>
<td>59.91↑HIB</td>
<td>99.20↑H</td>
<td>40.81</td>
</tr>
</tbody>
</table>

Some observations:

- Exploration all rule labelings more effective than increasing pop-limit
- Both combined give best results
- Shuffling helps
Combining labels

- Fast soft-matching implementation enables combining labels
- We simply concatenate two labels
- For the label-substitution features, we add features for each part separately, to reduce sparsity and hence overfitting
Combining labels: results

<table>
<thead>
<tr>
<th>System Name</th>
<th>BLEU ↑</th>
<th>METEOR ↑</th>
<th>BEER ↑</th>
<th>TER ↓</th>
<th>KRS ↑</th>
<th>Length</th>
<th>CPU time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hiero</td>
<td>31.63</td>
<td>30.56</td>
<td>13.15</td>
<td>59.28</td>
<td>58.03</td>
<td>97.15</td>
<td>3.34</td>
</tr>
<tr>
<td>Hiero 0&lt;sup&gt;th&lt;/sup&gt; + Hiero 1&lt;sup&gt;st&lt;/sup&gt;</td>
<td>32.30 ▲H ▲H&lt;sub&gt;0&lt;/sub&gt;</td>
<td>30.95 ▲H ▲H&lt;sub&gt;0&lt;/sub&gt;</td>
<td>13.75 ▲H ▲H&lt;sub&gt;0&lt;/sub&gt;</td>
<td>60.16 ▼H ▼H&lt;sub&gt;0&lt;/sub&gt;</td>
<td>60.13 ▲H ▲H&lt;sub&gt;0&lt;/sub&gt;</td>
<td>99.07 ▲H ▲H&lt;sub&gt;0&lt;/sub&gt;</td>
<td>7.18</td>
</tr>
<tr>
<td>SAMT+ Hiero 1&lt;sup&gt;st&lt;/sup&gt;</td>
<td>32.57 ▲H ▽H&lt;sub&gt;1&lt;/sub&gt;</td>
<td>31.07 ▼H ▽S ▽H&lt;sub&gt;1&lt;/sub&gt;</td>
<td>13.84 ▲H ▽S</td>
<td>59.94 ▼H ▽H&lt;sub&gt;1&lt;/sub&gt;</td>
<td>60.18 ▲H</td>
<td>99.13 ▲H ▽S</td>
<td>11.63</td>
</tr>
<tr>
<td>Bnd.Tags+ Hiero 1&lt;sup&gt;st&lt;/sup&gt;</td>
<td>32.65 ▲H ▽H&lt;sub&gt;1&lt;/sub&gt;</td>
<td>31.36 ▲H ▽B ▽H&lt;sub&gt;1&lt;/sub&gt;</td>
<td>14.16 ▲H ▽H&lt;sub&gt;1&lt;/sub&gt;</td>
<td>60.21 ▼H</td>
<td>61.46 ▲H ▽B ▽H&lt;sub&gt;1&lt;/sub&gt;</td>
<td>99.86 ▲H ▽H&lt;sub&gt;1&lt;/sub&gt;</td>
<td>10.32</td>
</tr>
</tbody>
</table>

- **BoundaryTags+Hiero 1<sup>st</sup>** is the overall best system for METEOR and KRS; nearly best for BLEU and BEER
  - Adding Hiero 1<sup>st</sup> label in addition to BoundaryTags further improves the word-order
Summary

- Discussed Hiero + labels, and motivated and described soft-matching

- Explained how to make soft-matching fast, and why this is important

- Explored several extensions to boost the effectiveness of soft-matching:
  1. Extending the search space to ensure exploration matching substitutions
  2. Shuffling as a way to reduce issues with extending the search space, and to reduce overfitting
  3. Experiments with double labels
Conclusions

- Efficient soft-matching is important for practical usefulness of the method but non-trivial. We provided the details of a fast implementation.

- Soft matching is typically superior to strict matching.

- Extension of the search that diversifies the explored labels can boost performance.

- Double labels can work, and further boost performance, despite risks of overfitting etc.
Questions?