Joint Decoding with Multiple Translation Models

Yang Liu, Haitao Mi, Yang Feng, and Qun Liu
Institute of Computing Technology,
Chinese Academy of Sciences
{yliu,htmi,fengyang,liuqun}@ict.ac.cn
System Combination

 decoding? postprocessing

System 1 → E1
System 2 → E2
System n → En

Combiner → e
This Work

- We propose a technique called joint decoding to combine different models directly in the decoding phase.
- Our preliminary work shows promising results.
Translation Hypergraph

give a talk
0-2

give talks
0-2

give
0-1
talk
1-2

S

fabiao yanjiang
0 1 2
Consensus Translations

- **Phrase-based**
- **Hierarchical phrase-based**
- **Tree-to-string**
Sharing Nodes

- give a speech
  - 0-2

- give a talk
  - 0-2

- give talks
  - 0-2

- give
  - 0-1

- talk
  - 1-2

- talks
  - 1-2

S
Scoring a Translation

\[ p(e | f) = \sum_{d \in \Delta(e, f)} p(e, d | f) \]

- target sentence
- source sentence
- one derivation
- the set of derivations that translate \( f \) into \( e \)

A derivation can come from any model!
MDD and MTD

\[
p(e \mid f) = \sum_{d \in \Delta(e, f)} \frac{\exp \left( \sum_{i} \lambda_i h_i (d, e, f) \right)}{Z}
\]

Blunsom et al. (2008)

\[
\hat{e} = \arg \max_{e} \left\{ \sum_{d \in \Delta(e, f)} \exp \left( \sum_{i} \lambda_i h_i (d, e, f) \right) \right\} \quad \text{max-translation decoding}
\]

\[
\approx \arg \max_{e, d} \left\{ \sum_{i} \lambda_i h_i (d, e, f) \right\} \quad \text{max-derivation decoding}
\]
### An Example of MTD

<table>
<thead>
<tr>
<th>model</th>
<th>feature</th>
<th>name</th>
<th>weight</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>t2s</td>
<td>p(e</td>
<td>f)</td>
<td>1.0</td>
<td>0.1</td>
</tr>
<tr>
<td></td>
<td>p(f</td>
<td>e)</td>
<td>1.0</td>
<td>0.2</td>
</tr>
<tr>
<td></td>
<td>l(e</td>
<td>f)</td>
<td>1.0</td>
<td>0.1</td>
</tr>
<tr>
<td></td>
<td>l(f</td>
<td>e)</td>
<td>1.0</td>
<td>0.3</td>
</tr>
<tr>
<td></td>
<td>rc</td>
<td>1.0</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>const</td>
<td>p(e</td>
<td>f)</td>
<td>1.0</td>
<td>0.2</td>
</tr>
<tr>
<td></td>
<td>p(f</td>
<td>e)</td>
<td>1.0</td>
<td>0.1</td>
</tr>
<tr>
<td></td>
<td>l(e</td>
<td>f)</td>
<td>1.0</td>
<td>0.3</td>
</tr>
<tr>
<td></td>
<td>l(f</td>
<td>e)</td>
<td>1.0</td>
<td>0.2</td>
</tr>
<tr>
<td></td>
<td>rc</td>
<td>1.0</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td></td>
<td>lm</td>
<td>1.0</td>
<td>0.7</td>
<td></td>
</tr>
<tr>
<td></td>
<td>wc</td>
<td>1.0</td>
<td>3</td>
<td></td>
</tr>
</tbody>
</table>

\[ \text{exp}(3.7) \times (\text{exp}(3.7) + \text{exp}(4.8)) \times \text{exp}(3.7) \]
Joint Decoding

tabiao yanjiang
0 1 2
Joint Decoding

fabiao yanjiang
0 1 2

give 0-1

S
Joint Decoding

fabiao yanjiang

0 1 2

give 0-1

talk 1-2

S
Joint Decoding

fabiao  yanjiang
0 1 2

give 0-1

talk 1-2

talks 1-2

S
Joint Decoding

fabiao yanjiang
0 1 2

give 0-1

talk 1-2

talks 1-2

S
Joint Decoding

fabiao yanjiang 0 1 2

give a speech 0-2

give a talk 0-2

give talks 0-2

give 0-1

talk 1-2

S
Joint Decoding

fabiao  yanjiang
0       1       2

give
0-1
give a talk
0-2
talk
1-2
S

a pruned packed hypergraph
Sharing Hyperedges

give 0-1

give a talk 0-2

talk 1-2

S
Joint Decoding with Different Rules

```
VP
VV NN
  fabiao yanjiang
```

S
Joint Decoding with Different Rules

X -> <fabiao, give>
Joint Decoding with Different Rules

X -> \langle \text{fabiao, give} \rangle

X -> \langle \text{yanjiang, talk} \rangle
Joint Decoding with Different Rules

(VP (VV:x_1) (NN:x_2)) \rightarrow x_1 \text{ a } x_2

X \rightarrow \langle \text{fabiao, give} \rangle

X \rightarrow \langle \text{yanjiang, talk} \rangle
## An Example of MDD

<table>
<thead>
<tr>
<th>model</th>
<th>feature</th>
<th>name</th>
<th>weight</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>p(e</td>
<td>f)</td>
<td></td>
<td>1.0</td>
</tr>
<tr>
<td></td>
<td>p(f</td>
<td>e)</td>
<td></td>
<td>1.0</td>
</tr>
<tr>
<td>hier</td>
<td>l(e</td>
<td>f)</td>
<td></td>
<td>1.0</td>
</tr>
<tr>
<td></td>
<td>l(f</td>
<td>e)</td>
<td></td>
<td>1.0</td>
</tr>
<tr>
<td></td>
<td>rc</td>
<td></td>
<td>1.0</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>p(e</td>
<td>f)</td>
<td></td>
<td>1.0</td>
</tr>
<tr>
<td></td>
<td>p(f</td>
<td>e)</td>
<td></td>
<td>1.0</td>
</tr>
<tr>
<td>t2s</td>
<td>l(e</td>
<td>f)</td>
<td></td>
<td>1.0</td>
</tr>
<tr>
<td></td>
<td>l(f</td>
<td>e)</td>
<td></td>
<td>1.0</td>
</tr>
<tr>
<td></td>
<td>rc</td>
<td></td>
<td>1.0</td>
<td>1</td>
</tr>
<tr>
<td>const</td>
<td>lm</td>
<td></td>
<td>1.0</td>
<td>0.7</td>
</tr>
<tr>
<td></td>
<td>wc</td>
<td></td>
<td>1.0</td>
<td>3</td>
</tr>
</tbody>
</table>
# Sharing Matrix

<table>
<thead>
<tr>
<th></th>
<th>Phrase</th>
<th>Hiero</th>
<th>T2S</th>
<th>S2T</th>
<th>T2T</th>
</tr>
</thead>
<tbody>
<tr>
<td>Phrase</td>
<td>node, edge</td>
<td>node, edge</td>
<td>node, edge</td>
<td>node</td>
<td>node</td>
</tr>
<tr>
<td>Hiero</td>
<td>node, edge</td>
<td>node, edge</td>
<td>node, edge</td>
<td>node</td>
<td>node</td>
</tr>
<tr>
<td>T2S</td>
<td>node, edge</td>
<td>node, edge</td>
<td>node, edge</td>
<td>node</td>
<td>node</td>
</tr>
<tr>
<td>S2T</td>
<td>node</td>
<td>node</td>
<td>node, edge</td>
<td>node, edge</td>
<td>node, edge</td>
</tr>
<tr>
<td>T2T</td>
<td>node</td>
<td>node</td>
<td>node</td>
<td>node, edge</td>
<td>node, edge</td>
</tr>
</tbody>
</table>
How to Tune Feature Weights for MTD?

<table>
<thead>
<tr>
<th>model</th>
<th>feature</th>
<th>name</th>
<th>weight</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>p(e</td>
<td>f)</td>
<td>1.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>p(f</td>
<td>e)</td>
<td>1.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>l(e</td>
<td>f)</td>
<td>1.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>l(f</td>
<td>e)</td>
<td>1.0</td>
</tr>
<tr>
<td>hier</td>
<td>rc</td>
<td>1.0</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td></td>
<td>p(e</td>
<td>f)</td>
<td>1.0</td>
<td>0.2</td>
</tr>
<tr>
<td></td>
<td>p(f</td>
<td>e)</td>
<td>1.0</td>
<td>0.1</td>
</tr>
<tr>
<td></td>
<td>l(e</td>
<td>f)</td>
<td>1.0</td>
<td>0.3</td>
</tr>
<tr>
<td></td>
<td>l(f</td>
<td>e)</td>
<td>1.0</td>
<td>0.2</td>
</tr>
<tr>
<td></td>
<td>rc</td>
<td>1.0</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>t2s</td>
<td>lm</td>
<td>1.0</td>
<td>0.7</td>
<td></td>
</tr>
<tr>
<td></td>
<td>wc</td>
<td>1.0</td>
<td>3</td>
<td></td>
</tr>
</tbody>
</table>

\[
\hat{e} = \arg \max_e \left\{ \sum_{d \in \Delta(e,f)} \exp \left( \sum_i \lambda_i h_i (d,e,f) \right) \right\}
\]

\[
(\exp(3.7) + \exp(4.8)) \times \exp(3.7)
\]
MERT for MDD

\[ f \]

\[
\begin{array}{l}
\text{e}_1 \\
\quad \text{d}_1 \quad 0.1 \ 0.2 \ 0.3 \ 0.1 \\
\text{e}_2 \\
\quad \text{d}_1 \quad 0.2 \ 0.1 \ 0.3 \ 0.1 \\
\text{e}_3 \\
\quad \text{d}_1 \quad 0.1 \ 0.3 \ 0.1 \ 0.2 \\
\end{array}
\]
MERT for MTD

<table>
<thead>
<tr>
<th></th>
<th>f</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>e1</td>
<td>model 1</td>
<td>model 2</td>
</tr>
<tr>
<td></td>
<td>d1</td>
<td>0.1 0.2 0.3 0.1</td>
<td>0.0 0.0 0.0 0.0</td>
</tr>
<tr>
<td></td>
<td>d2</td>
<td>0.0 0.0 0.0 0.0</td>
<td>0.2 0.3 0.4 0.1</td>
</tr>
<tr>
<td></td>
<td>e2</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>d1</td>
<td>0.0 0.0 0.0 0.0</td>
<td>0.3 0.1 0.2 0.1</td>
</tr>
<tr>
<td></td>
<td>d2</td>
<td>0.1 0.2 0.3 0.1</td>
<td>0.0 0.0 0.0 0.0</td>
</tr>
<tr>
<td></td>
<td>e3</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>d1</td>
<td>0.1 0.2 0.1 0.2</td>
<td>0.0 0.0 0.0 0.0</td>
</tr>
<tr>
<td></td>
<td>d2</td>
<td>0.0 0.0 0.0 0.0</td>
<td>0.1 0.3 0.2 0.1</td>
</tr>
</tbody>
</table>
Curves

\[ f(x) = \sum_{k=1}^{K} e^{a_k x + b_k} \]
Setup

- Models
  - Hierarchical phrase-based (Chiang, 2005)
  - Tree-to-string (Liu et al., 2006)
- Training set: FBIS (6.9M + 8.9M)
- Language model: 4-gram trained on GIGAWORD Xinhua portion
- Development set: NIST 2002 C2E
- Test set: NIST 2005 C2E
## Individual Decoding Vs. Joint Decoding

<table>
<thead>
<tr>
<th>Model</th>
<th>Sharing</th>
<th>Max-derivation</th>
<th>Max-translation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Time</td>
<td>BLEU</td>
</tr>
<tr>
<td>Hiero</td>
<td>-</td>
<td>40.53</td>
<td>30.11</td>
</tr>
<tr>
<td>T2S</td>
<td>-</td>
<td>6.13</td>
<td>27.23</td>
</tr>
<tr>
<td>both</td>
<td>node &amp; edge</td>
<td>48.45</td>
<td>31.63</td>
</tr>
</tbody>
</table>

**Max-translation**

- **Max-derivation**
  - Hiero: 40.53, 30.11
  - T2S: 6.13, 27.23
  - Both: 48.45, 31.63

- **Time**
  - Hiero: 44.87
  - T2S: 6.69
  - Both: 54.91
## Compared with System Combination

<table>
<thead>
<tr>
<th>Method</th>
<th>Model</th>
<th>BLEU</th>
</tr>
</thead>
<tbody>
<tr>
<td>individual</td>
<td>Hiero</td>
<td>30.11</td>
</tr>
<tr>
<td></td>
<td>T2S</td>
<td>27.23</td>
</tr>
<tr>
<td>system comb.</td>
<td>both</td>
<td>31.50</td>
</tr>
<tr>
<td>joint</td>
<td>both</td>
<td>31.63</td>
</tr>
</tbody>
</table>
## Individual Training Vs. Joint Training

<table>
<thead>
<tr>
<th>Training</th>
<th>Max-derivation</th>
<th>Max-translation</th>
</tr>
</thead>
<tbody>
<tr>
<td>individual</td>
<td>30.70</td>
<td>29.95</td>
</tr>
<tr>
<td>joint</td>
<td>31.63</td>
<td>30.79</td>
</tr>
</tbody>
</table>
Conclusion and Future Work

- We have presented a framework for combining different translation models in the decoding phrase.

Future work

- Including more models
- Forced decoding
- Hypergraph-based MERT (Kumar et al., 2009)
Thanks!