Large-scale Discriminative n-gram Language Models for Statistical Machine Translation

Zhifei Li and Sanjeev Khudanpur
Johns Hopkins University
Language Model: Data Mismatch

- A regular language model is trained on well-formed monolingual corpora (e.g., Gigaword)
  - it does not require bilingual data
- During training, the language model does not see the MT outputs
  - But, the LM will be used to rank MT outputs
  - MT outputs differ substantially from Gigaword
- Can we make the LM task-specific without losing its big advantage in using enormous monolingual data?
Task: reranking MT outputs

I am the best translation.

<table>
<thead>
<tr>
<th>Hypothesized translation</th>
<th>TM</th>
<th>LM</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>i am a most best translation .</td>
<td>9.1</td>
<td>10</td>
<td>19.1</td>
</tr>
<tr>
<td>i are the best translation .</td>
<td>9.0</td>
<td>10</td>
<td>19.0</td>
</tr>
<tr>
<td>i am the best translation .</td>
<td>10</td>
<td>8</td>
<td>18</td>
</tr>
<tr>
<td>i are the good translate .</td>
<td>9</td>
<td>8</td>
<td>17</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Hypothesized translation</th>
<th>TM</th>
<th>LM</th>
<th>Corrective</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>i am the best translation .</td>
<td>10</td>
<td>8</td>
<td>1.0</td>
<td>19</td>
</tr>
<tr>
<td>i am a most best translation .</td>
<td>9.1</td>
<td>10</td>
<td>-0.5</td>
<td>18.6</td>
</tr>
<tr>
<td>i are the best translation .</td>
<td>9.0</td>
<td>10</td>
<td>-0.5</td>
<td>18.5</td>
</tr>
<tr>
<td>i are the good translate .</td>
<td>9</td>
<td>8</td>
<td>-1</td>
<td>16</td>
</tr>
</tbody>
</table>
Discriminative LM reranking

- A discriminative language model should
  - discover useful n-gram features
  - find optimal weights for these features
- The discriminative LM is trained on
  - hypotheses produced by a baseline system
  - desired translation
Discriminative Modeling

- Global linear model
  \[ s(f, e) = \Phi(f, e) \cdot \vec{\alpha} = \sum_j \Phi_j(f, e) \alpha_j \]

- Training
  \[ \vec{\alpha}^* = \arg \max_{\vec{\alpha}} F(Data, \vec{\alpha}) \]

- Decision rule
  \[ e^* = \arg \max_{e \in \text{TRANS}(f)} s(f, e) \]
Discriminative Reranking

- Score after reranking
  \[ s(f, e) = \Phi(f, e) \cdot \tilde{\alpha} \]
  \[ = \alpha_0 \Phi_0(f, e) + \sum_{j \in [1, J]} \alpha_j \Phi_j(f, e) \]

- Features
  - baseline feature
    \[ \Phi_0(f, e) = \text{Baseline score for translation } e \]
  - reranking n-gram features, e.g.,
    \[ \Phi_1(f, e) = \text{Count of the bigram "the of" in } e \]
Leave-one-out Baseline Training

part-1

part-2

part-n

Baseline

$N$-bests
Leave-one-out Baseline Training

- **part-1**
- **part-2**
- **part-n**

→

**Baseline**

→

**N-bests**
Leave-one-out Baseline Training

part-1

part-2

part-n

Baseline

N-bests
Data Selection

- Data is very noisy in our MT application
  - Human annotation is noisy
  - Automatic sentence alignment is noisy
- We aim to select high-quality training data for discriminative training
  - An training example will be selected only if it satisfies certain conditions
\[ G(\text{ref}, \text{oracle}) > T_1 \quad \rightarrow \quad \text{matched translation} \]

\[ G(\text{ref}, \text{oracle}) - G(\text{ref}, \text{1best}) > T_2 \quad \rightarrow \quad \text{profitable} \]

\[ G(\text{oracle}, \text{1best}) > T_3 \quad \rightarrow \quad \text{correctable} \]
## Experiments: facts

<table>
<thead>
<tr>
<th>Language pair</th>
<th>Chinese to English</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Translation system</strong></td>
<td>Hiero</td>
</tr>
<tr>
<td><strong>Language model data</strong></td>
<td>160 M words</td>
</tr>
<tr>
<td><strong>Translation model data</strong></td>
<td>30M words</td>
</tr>
<tr>
<td><strong>Number of partitions</strong></td>
<td>30</td>
</tr>
<tr>
<td><strong>DEV set for baseline MERT</strong></td>
<td>MT03</td>
</tr>
<tr>
<td><strong>DEV set for reranking</strong></td>
<td>MT04</td>
</tr>
<tr>
<td><strong>Test sets</strong></td>
<td>MT05, MT06</td>
</tr>
<tr>
<td><strong>N-best size</strong></td>
<td>300 unique</td>
</tr>
<tr>
<td><strong>Training algorithm</strong></td>
<td>averaged perceptron</td>
</tr>
</tbody>
</table>
Data Selection: varying $T_1$

Figure 5: BLEU Scores on MT’04 when varying the value of $T_1 \in [0.05, 0.25]$ with a step size 0.01.

$G(\text{ref, oracle}) > T_1$  

*matched translation*
Data Selection: varying $T_2$

Figure 6: BLEU Scores on MT’04 when varying the value of $T_2 \in [0.01, 0.10]$ with a step size 0.01.

$G(\text{ref, oracle}) - G(\text{ref, 1best}) > T_2$ profitable
Data Selection: varying $T_3$

Figure 7: BLEU Scores on MT’04 when varying the value of $T_3 \in [0.20, 0.75]$ with a step size 0.05.

$G(oracle, 1best) > T_3$ → correctable
## Experiments: reranking results

<table>
<thead>
<tr>
<th>Task</th>
<th>Baseline</th>
<th>Reranking</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Full</td>
</tr>
<tr>
<td>MT’04</td>
<td>0.357</td>
<td>0.365</td>
</tr>
<tr>
<td>MT’05</td>
<td>0.326</td>
<td>0.332</td>
</tr>
<tr>
<td>MT’06</td>
<td>0.283</td>
<td>0.292</td>
</tr>
</tbody>
</table>

### Selected data

<table>
<thead>
<tr>
<th>T1</th>
<th>T2</th>
<th>T3</th>
<th>Selected data</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.10</td>
<td>0.01</td>
<td>0.25</td>
<td>610K out of 1M</td>
</tr>
</tbody>
</table>

### n-gram | active
---|---
1-gram | 34k
2-gram | 1908k

Tuesday, August 18, 2009
Summary

- We have developed a discriminative n-gram LM to rerank MT outputs
- Discriminative LM reranking improves the translation quality over a state of the art system
- With data selection, we can train a better/comparable model using less data
Hypergraph-based Discriminative Rescoring

- generate a hypergraph (instead of an n-best) for each Chinese sentence
- identify oracle translations on the hypergraph
- train a model and use it in decoding on a hypergraph
- the hypergraph is pruned using the posterior pruning
## Hypergraph rescoring results

<table>
<thead>
<tr>
<th>System</th>
<th>MT04</th>
<th>MT05</th>
<th>MT06</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Chiang’07 Baseline</td>
<td>34.6</td>
<td>31.8</td>
<td>NA</td>
</tr>
<tr>
<td>Ours</td>
<td>35.7</td>
<td>32.6</td>
<td>28.3</td>
</tr>
<tr>
<td>N-best</td>
<td>36.5</td>
<td>33.3</td>
<td>29.4</td>
</tr>
<tr>
<td>Hypergraph</td>
<td>35.9</td>
<td>33.0</td>
<td>28.2</td>
</tr>
</tbody>
</table>
Joshua: an open-source parsing-based MT decoder

• Team members
  - **JHU**: Zhifei Li, Chris Callison-Burch, Sanjeev Khudanpur, Wren Thornton, Jonathan Weese, and Omar Zaidan
  - **UMD**: Chris Dyer
  - **U of Minnesota**: Lane Schwartz

• Functions
  - Chart-parsing, pruning, language model integration, kbest extraction, distributed and parallel decoding
  - Suffix-array based grammar extraction
  - Minimum error rate training
Thank you!
谢谢！