Inducing a Discriminative Parser to Optimize Machine Translation Reordering

Graham Neubig$^{1,2,3}$, Taro Watanabe$^2$, Shinsuke Mori$^1$

now at

1. Kyoto University
2. NICT
3. now at NAIST
Preordering

- Long-distance reordering is a weak point of SMT
- Preordering first reorders, then translates

\[ F = \textit{kare wa gohan o tabeta} \]
\[ F' = \textit{kare wa tabeta gohan o} \]
\[ E = \textit{he ate rice} \]

- A good preorderer will effectively find \( F' \) given \( F \)
Inducing a Discriminative Parser to Optimize Machine Translation Reordering

**Syntactic Preordering**

- Define rules over a syntactic parse of the source

```
F = kare wa gohan o tabeta
```

What if we don't have a parser in the source language?

```
D =
   waP
   PRN wa
   S
   oP
   V

F = kare wa gohan o tabeta

E = he ate rice
```

```
D' =
   waP
   PRN wa
   S
   V
   oP

F' = kare wa tabeta gohan o

E = he ate rice
```
Bracketing Transduction Grammars

[Wu 97]

- Binary CFGs with only straight (S) and inverted (I) non-terminals, and pre-terminals (T)

- Language independent
- BTG tree uniquely defines a reordering
3-Step BTG Grammar Training for Reordering [DeNero+ 11]

1) Bilingual Grammar Induction

F= kare wa gohan o tabeta
A= kare wa gohan o tabeta
E= he ate rice

Unsupervised Induction (Several Hand-Tuned Features)

S

T T T T T T

2) Parser Training

Supervised Training (Max Tree Accuracy)

Parsing Model

3) Reorderer Training

Supervised Training (Max Label Accuracy)

Reordering Model
3-Step BTG Grammar Induction for Reordering [DeNero+ 11]

1) Parsing

2) Reordering

F = kare wa gohan o tabeta
Our Work: Inducing a Parser to Optimize Reordering

- What if we can reduce three steps to one, and directly maximize ordering accuracy?

Training

\[ F= \text{kare wa gohan o tabeta} \]
\[ A= \text{he ate rice} \]

Supervised Learning (Max Reordering Accuracy)

Parsing/Reordering Model

Testing

\[ F= \text{kare wa gohan o tabeta} \]

Parsing/Reordering Model

\[ S \quad \text{I} \quad S \]

\[ \text{kare wa tabeta gohan o} \]
Optimization Framework
Optimization Framework

- **Input:** Source sentence $F$
  
  \[ F = \text{kare wa gohan o tabeta} \]

- **Output:** Reordered source sentence $F'$
  
  \[ F' = \text{kare wa tabeta gohan o} \]

- **Latent:** Bracketing transduction grammar derivation $D$

\[ D = \]

```
   S
  / \  \
 /    \ /
S  I  S
 |    |
T    T
```

\[ S = \text{S}
\]

\[ T = \text{T} \]
Scores and Losses

- Define a score over source sentences and derivations

\[ S(F, D; w) = \sum_i w_i \phi_i(F, D) \]

- Optimization finds a weight vector that minimizes loss

\[ \arg\min_w \sum_{F,F'} L(F'^*, \arg\max_{F' \leftarrow F, D} S(F, D; w)) \]
Out of these, we want easy-to-reproduce trees

[DeNero+ 11] finds trees with bilingual parsing model

Our model discovers trees during training
Training: Latent Online Learning

- Find
  - model parse of maximal score and
  - oracle parse of maximal score among parses of minimal loss

\[
D = \arg\max_{\hat{D}} S(F, \hat{D}) \quad \text{and} \quad \hat{D} = \arg\max_{\tilde{D} \in \arg\min_{D} L(F, D)} S(F, \tilde{D})
\]

- Adjust weights (example: perceptron)
  \[
  w \leftarrow w + \phi(F, \hat{D}) - \phi(F, D)
  \]
Considering Loss in Online Learning

- Consider loss (how bad is the mistake?)

\[
\begin{align*}
\text{kare wa tabeta gohan o} & \quad \text{reference (L=0)} \\
\text{kare wa gohan tabeta o} & \quad \text{L=1} \\
\text{o gohan tabeta wa kare} & \quad \text{L=8}
\end{align*}
\]

- Make it easy to choose trees with high loss in training
  
  → To avoid high-loss trees, must give a large penalty

\[
D = \arg\max_{\tilde{D}} S(F, \tilde{D}) + L(F, \tilde{D})
\]
Parser
Parsing Setup: Standard Discriminative Parser

- Features independent with respect to each node
- Parsing, reordering possible in $O(n^3)$ with CKY
- Multi-word pre-terminals allowed
Language Independent Features

- No linguistic analysis necessary

![Diagram](image)

- **Lexical**: Left, right, inside, outside, boundary words
- **Class**: Same as lexical but induced classes
- **Phrase Table**: Whether span exists in phrase table
- **Balance**: Left branching or right branching?
Language Dependent Features

- **POS Features**: Same as lexical, but over POSs

- **CFG Features**: Whether nodes match supervised parser's spans
Reordering Losses
Reordering Losses [Talbot+ 11]:
Chunk Fragmentation

- How many chunks are necessary to reproduce reference?

**System Reordering:**

<
<s>
\begin{array}{c}
\text{kare} \\
\text{wa} \\
\text{gohan} \\
\text{o} \\
\text{tabeta}
\end{array}
</s>

**Reference Reordering:**

<
<s>
\begin{array}{c}
\text{kare} \\
\text{wa} \\
\text{tabeta} \\
\text{gohan} \\
\text{o}
\end{array}
</s>

**Loss:**

\[ L_{\text{chunk}} (F, \tilde{D}) = \text{Number of Chunks} - 1 \]

**Accuracy:**

\[ A_{\text{chunk}} (F, \tilde{D}) = 1 - \frac{(\text{Number of Chunks} - 1)}{(J+1)} \]
Reordering Losses [Talbot+ 11]: Kendall's Tau

- How many pairs of reversed words?

<table>
<thead>
<tr>
<th>System Reordering:</th>
<th>kare wa gohan o tabeta</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reference Reordering:</td>
<td>kare wa tabeta gohan o</td>
</tr>
</tbody>
</table>

**Loss:**

\[ L_{\tau}(F, \tilde{D}) = \text{Reversed Words} \]

**Accuracy:**

\[ A_{\tau}(F, \tilde{D}) = 1 - \frac{\text{Reversed Word/Potential Reversed Words}}{} \]
Calculating Loss by Node

- Large-margin training, must calculate loss efficiently
  \[ D = \text{argmax} \; S(F, \tilde{D}) + L(F, \tilde{D}) \]
- Can factor loss by node as well (detail in paper)
Experiments
Experimental Setup

- English-Japanese and Japanese-English translation
- Data from the Kyoto Free Translation Task

<table>
<thead>
<tr>
<th></th>
<th>sent.</th>
<th>word (ja)</th>
<th>word (en)</th>
</tr>
</thead>
<tbody>
<tr>
<td>RM-train</td>
<td>602</td>
<td>14.5k</td>
<td>14.3k</td>
</tr>
<tr>
<td>RM-test</td>
<td>555</td>
<td>11.2k</td>
<td>10.4k</td>
</tr>
<tr>
<td>LM/TM</td>
<td>329k</td>
<td>6.08M</td>
<td>5.91M</td>
</tr>
<tr>
<td>tune</td>
<td>1166</td>
<td>26.8k</td>
<td>24.3k</td>
</tr>
<tr>
<td>test</td>
<td>1160</td>
<td>28.5k</td>
<td>26.7k</td>
</tr>
</tbody>
</table>
Experimental Setup

• Reordering Model Training:
  • 500 iterations
  • Using Pegasos with regularization constant $10^{-3}$
  • Default: chunk fragmentation loss, standard features

• Translation: Moses with lexicalized reordering

• Compare: Original order, 3-step training, the proposed method
Result:
Proposed Model Improves Reordering

- Results for chunk fragmentation/Kendall's Tau
Result:
Proposed Model Improves Translation

- Results for BLEU and RIBES:
Result:
Adding Linguistic Info (Generally) Helps
Result:
Training Loss Affects Reordering

- Optimized criterion is higher on test set as well
Result: Training Loss Affects Translation

- Optimizing chunk fragmentation generally gives best results
Result: Automatic Alignments, Better than Nothing, Worse than Manual
Yoshimitsu Ashikaga was the 3rd Sei Taishogun of the Muromachi Shogunate and reigned from 1368 to 1394.
The 3rd Sei Taishogun was of the Muromachi Shogunate and reigned from 1368 to 1394.
Conclusion

- Presented a method to induce a discriminative parser to optimize machine translation reordering
- Favorable results for English ↔ Japanese
- Future Work:
  - Development of better features
  - Incorporation into tree-to-string translation
  - Probabilistic inference

Will be!

^ Available Open Source: http://www.phontron.com/lader
Thank you!