Larger Feature Set Approach for MT: IWSLT 2007

Taro Watanabe, Jun Suzuki, Katsuhito Sudoh, Hajime Tsukada and Hideki Isozaki

NTT Communication Science Labs.
{taro,jun,sudoh,tsukada,Isozaki}@cslab.kecl.ntt.co.jp
NTT SMT System

Hierarchical Phrase-based SMT

Decoder maximizes:
\[ \hat{e} = \arg\max_{e} w^\top \cdot h(f, e) \]

Reranker votes:
\[ \hat{e} = \arg\max_{e} \{ w_i^\top \cdot h(f, e) \}_{i=1}^n \]

Both systems employ large # of sparse features
Hierarchical SMT

- Hierarchically embedded phrases (Chiang, 2005)
- An efficient top-down search (Watanabe et al., 2006)
Feature Set

• 5-gram language model
• Phrase probabilities
• Lexical weights
• Insertion/deletion penalties
• # of words/phrases

+ Sparse Features
Sparse Features

- Preserve word alignment inside hierarchical phrases
- Word-wise features (word-pair, target-bigram etc.)
Factoring

- Use of normalized tokens (POS/word class/prefix/etc.)
- Consider all possible combinations
- POS: expanded into all possible solutions
Sparse Features

• Sparse features:
  • \{1,2\}-gram of word-pairs
  • target word bigram
  • Insertion/deletion features
  • Hierarchical dependency features

• Word Factoring:
  • Surface word
  • Word class
  • POS/NE
  • WordNet’s synset
  • 4-letter prefix/suffix
Online Training

Training data: $\mathcal{T} = \{(f^t, e^t)\}_{t=1}^T$

$m$-best oracles: $O = \{\}_{t=1}^T$

$i = 0$

1: for $n = 1, \ldots, N$ do
2: for $t = 1, \ldots, T$ do
3: $C^t \leftarrow \text{best}_k(f^t; w^i)$
4: $O^t \leftarrow \text{oracle}_m(O^t \cup C^t; e^t)$
5: $w^{i+1} = \text{update } w^i \text{ using } C^t \text{ w.r.t. } O^t$
6: $i = i + 1$
7: end for
8: end for
9: return $\frac{\sum_{i=1}^{NT} w^i}{NT}$
Large Margin Constraints

\[ \hat{w}^{i+1} = \arg\min_{w^{i+1}} \frac{1}{2} \|w^{i+1} - w^i\|^2 + C \sum_{\hat{e}, e'} \xi(\hat{e}, e') \]

subject to

\[ s^{i+1}(f^t, \hat{e}) - s^{i+1}(f^t, e') + \xi(\hat{e}, e') \geq L(\hat{e}, e'; e^t) \]

\[ \xi(\hat{e}, e') \geq 0 \]

\[ \forall \hat{e} \in O^t, \forall e' \in C^t \]

• Constrained by m-oracle + k-best.
• “C” to control the amount of updates.
Reranker
Reranking

**Perceptron Training**

Training data: \( T = \{(f^t, C^t, e^t)\}_{i=1}^T \)

1: for \( n = 1, \ldots, N \) do
2: \( w^n = w^{n-1} \)
3: for \( t = 1, \ldots, T \) do
4: \( R = \text{rerank}(C^t; w^n) \)
5: for \( i = 1, \ldots, |R| \) do
6: for \( j = i + 1, \ldots, |R| \) do
7: if \( L(R_j, R_i; e^t) > 0 \) then
8: \( w^n = \text{update } w^n \text{ using } R_i \text{ and } R_j \)
9: end if
10: end for
11: end for
12: end for
13: end for
14: return \( \{w^n\}_{n=1}^N \)

**Decoding (Voting)**

\( k \)-best translation list: \((f, C)\)

Weight vectors: \( \{w^n\}_{n=1}^N \)

Votes: \( V = 0 \)

1: for \( n = 1, \ldots, N \) do
2: \( \hat{i} = \arg\max_i \{w^n\}^\top \cdot h(f, C_i) \)
3: \( V_{\hat{i}} = V_{\hat{i}} + 1 \)
4: end for
5: return \( C_{\hat{i}} \) where \( \hat{i} = \arg\max_i V_i \)

**Parameter Update**

\[
w^n = w^n + L(R_j, R_i; e^t) \cdot \left( h(f^t, R_j) - h(f^t, R_i) \right)
\]
Objectives

- Document-BLEU or sentence-BLEU?

$$\text{BLEU}(E; \hat{E}) = \exp \left( \frac{1}{N} \sum_{n=1}^{N} \log p_n(E, \hat{E}) \right) \cdot \text{BP}(E, \hat{E})$$

- Our method: compute the difference from an oracle BLEU (Watanabe et al., 2006)

$$\text{BLEU}(\{\hat{e}^1, ..., \hat{e}^{t-1}, e', \hat{e}^{t+1}, ..., \hat{e}^T\}; E)$$

- Loss by an approximated BLEU $\approx$ document-wise loss.
Task Setting
Preprocessing

- Removed bitexts matching regexp: [0-9]
- English: MaxEnt/Brill POS tagger
- Arabic: Isolate Arabic scripts/punctuations
- Italian: Treetagger
- Japanese/Chinese: HMM-based POS/NE tagger
- Casing preserved for English
- Punctuation removed for source side
## Bitexts

<table>
<thead>
<tr>
<th></th>
<th>ar-en</th>
<th>it-en</th>
<th>ja-en</th>
<th>zh-en</th>
</tr>
</thead>
<tbody>
<tr>
<td>sentences</td>
<td>833K</td>
<td>854K</td>
<td>1.0M</td>
<td>3.3M</td>
</tr>
<tr>
<td>words</td>
<td>25M</td>
<td>24M</td>
<td>8.6M</td>
<td>57M</td>
</tr>
<tr>
<td>vocabulary</td>
<td>132K</td>
<td>67K</td>
<td>254K</td>
<td>961K</td>
</tr>
<tr>
<td>source</td>
<td>LDC</td>
<td>EuroParl</td>
<td>NiCT</td>
<td>LDC</td>
</tr>
</tbody>
</table>

- Data comes from various sources (LDC or public domain)
- We used devset 4,5,5b for tuning, since they had ASR data.
Task Adaptation

Source side 3-gram perplexity

<table>
<thead>
<tr>
<th></th>
<th>ar-en</th>
<th>it-en</th>
<th>ja-en</th>
<th>zh-en</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>dev 4,5,5b</strong></td>
<td>561.96</td>
<td>277.24</td>
<td>51.29</td>
<td>188.49</td>
</tr>
<tr>
<td><strong>test</strong></td>
<td>214.99</td>
<td>271.39</td>
<td>13.45</td>
<td>73.18</td>
</tr>
</tbody>
</table>

- Sample bitexts for phrase-table extraction (Ittycheriah and Roukos, 2007)
- For each source sentence in test(dev) set:
  - Extract bitexts from the universe of training data.
  - Similarity measured by ngram precision.
ASR Translation

• 1-best ASR translation
• 20-best ASR translation
  • Translate all the 20-bests and select the best one by our reranker.
• Various word/sentence-wise confidence measures integrated as features.
Parameter Estimation

- Decoder:
  - Estimated on devset 4, 5, 5b.
  - 200-300 iterations

- Reranker:
  - 1,000-best list
  - Estimated on devset 4, 5, 5b and IWSLT’s 20,000 sentences.
Results (BLEU)

- ASR-1-best + 1-best
- ASR-20-best + rerank (devset)
- ASR-1-best + rerank (devset+IWSLT)
- clean 1-best
- clean rerank (devset)
- clean rerank (devset+IWSLT)
Post Evaluation

• Use IWSLT data only.....
• Held-out set to terminate iterations
• Arabic/Japanese/Chinese are close to IWSLT data.
  • Estimated on devset 1 and 2, held-out devset 3.
• Italian data is totally different:
  • Extract phrases from devset 5b, too
  • Estimation on devset 4 and 5, held-out devset 5b
Results (BLEU)

Primary

Post-Evaluation

<table>
<thead>
<tr>
<th>Language</th>
<th>Primary Rank</th>
<th>Post-Evaluation Rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>ar-en (ASR)</td>
<td>10th</td>
<td>1st</td>
</tr>
<tr>
<td>ar-en (clean)</td>
<td>9th</td>
<td>2nd</td>
</tr>
<tr>
<td>it-en (ASR)</td>
<td>4th</td>
<td>3rd</td>
</tr>
<tr>
<td>it-en (clean)</td>
<td>4th</td>
<td>3rd</td>
</tr>
<tr>
<td>ja-en (ASR)</td>
<td>5th</td>
<td>2nd</td>
</tr>
<tr>
<td>ja-en (clean)</td>
<td>1st</td>
<td>5th</td>
</tr>
<tr>
<td>zh-en (clean)</td>
<td>7th</td>
<td>14th</td>
</tr>
</tbody>
</table>
Conclusion

- NTT SMT System:
  - Large # of features are integrated both in decoder/reranker
  - Careful devset selection
  - Careful tuning
  - Larger data helps for reranking

- Future Work:
  - More rich features, more experiments.