Online Large-Margin Training for SMT

Taro Watanabe, Jun Suzuki, Hajime Tsukada and Hideki Isozaki
NTT Communication Science Labs.
Overview
Overview

- MER Training Approach (Och, 2003):
  - Do not scale to large # of parameters.
Overview

• MER Training Approach (Och, 2003):
  • Do not scale to large # of parameters.

• Online Discriminative Training Approaches (Tillmann and Zhang, 2006; Liang et al., 2006):
  • Large # of parameters estimated on large data, but moderate improvements.
Overview

• MER Training Approach (Och, 2003):
  • Do not scale to large # of parameters.

• Online Discriminative Training Approaches (Tillmann and Zhang, 2006; Liang et al., 2006):
  • Large # of parameters estimated on large data, but moderate improvements.

• This work:
  • Online Large-Margin Training
  • Millions of parameters
  • Less than 1K sentences for training
Statistical Machine Translation

\[ \hat{e} = \arg\max_{e} w^\top \cdot h(f, e) \]
Hierarchical Phrase-based SMT

- Phrase embedded phrases via non-terminals (Chiang, 2005)
- An efficient top-down search (Watanabe et al., 2006)
Hierarchical Phrase-based SMT

- Phrase embedded phrases via non-terminals (Chiang, 2005)
- An efficient top-down search (Watanabe et al., 2006)
Features
Features

• A standard Hiero-like features (Chiang, 2005):
  • n-gram language model, (hierarchical) phrase translation probabilities etc.
  • Phrase motivated penalties
Features

- A standard Hiero-like features (Chiang, 2005):
  - n-gram language model, (hierarchical) phrase translation probabilities etc.
  - Phrase motivated penalties

- Sparse features:
  - Unigram/bigram word pair features
  - Target bigram features
  - Insertion features
  - Hierarchical features
Sparse Features

- Word pair features (unigram and bigram)
- Bigram features are ordered by the target side.
Sparse Features

- Word pair features (unigram and bigram)
- Bigram features are ordered by the target side.
Sparse Features

- Word pair features (unigram and bigram)
- Bigram features are ordered by the target side.
Sparse Features

- Target bigram features.

\[ e_{i-1} e_i e_{i+1} e_{i+2} e_{i+3} e_{i+4} \]

\[ f_{j-1} f_j f_{j+1} f_{j+2} f_{j+3} \]
Sparse Features

- Target bigram features.
Sparse Features

- Insertion features.
- Each inserted word is associated with all the source words.
Sparse Features

• Insertion features.

• Each inserted word is associated with all the source words.
Sparse Features

- Hierarchical features.
- Dependency structure on the source side.
Sparse Features

• Hierarchical features.

• Dependency structure on the source side.
Sparse Features

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

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

- Use of normalized tokens (POS/word class/prefix/etc.)
- Consider all possible combinations
• Use of normalized tokens (POS/word class/prefix/etc.)
• Consider all possible combinations
Online Training Algorithm

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

$m$-best oracles: $O = \{\}_t^T$

$i = 0$

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 $\frac{\sum_{i=1}^{NT} w^i}{NT}$
Online Training Algorithm

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

\( m \)-best oracles: \( \mathcal{O} = \{\}^T_{t=1} \)

\( i = 0 \)

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

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 \( \frac{\sum_{i=1}^{NT} w^i}{NT} \)

Online Training Algorithm

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

\( m \)-best oracles: \( O = \{\}_t \)

\( i = 0 \)

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

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

\( m \)-best oracles: \( \mathcal{O} = \{\} \)

\( i = 0 \)

1: \textbf{for} \( n = 1, \ldots, N \) \textbf{do}

2: \hspace{1em} \textbf{for} \( t = 1, \ldots, T \) \textbf{do}

3: \hspace{2em} \( C^t \leftarrow \text{best}_k(f^t; w^i) \)

4: \hspace{2em} \( O^t \leftarrow \text{oracle}_m(O^t \cup C^t; e^t) \)

5: \hspace{2em} \( w^{i+1} = \text{update } w^i \text{ using } C^t \text{ w.r.t. } O^t \)

6: \hspace{2em} \( i = i + 1 \)

7: \hspace{1em} \textbf{end for}

8: \hspace{1em} \textbf{end for}

9: \textbf{return} \( \frac{\sum_{i=1}^{NT} w^i}{NT} \)
Online Training Algorithm

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

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

$i = 0$

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: $O^t \leftarrow \text{oracle}_m(\mathcal{O}^t \cup C^t; e^t)$
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 $\sum_{i=1}^{NT} \frac{w^i}{NT}$
Online Training Algorithm

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

$m$-best oracles: $\mathcal{O} = \{\}^T_{t=1}$

$i = 0$

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: $\mathcal{O}^t \leftarrow \text{oracle}_m(\mathcal{O}^t \cup C^t; e^t)$
5: $w^{i+1} = \text{update } w^i \text{ using } C^t \text{ w.r.t. } \mathcal{O}^t$
6: $i = i + 1$
7: end for
8: end for
9: return $\frac{\sum_{i=1}^{NT} w^i}{NT}$

Liang et al. (2006) discarded possibly better oracles.
Online Training Algorithm

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 \( \frac{\sum_{i=1}^{NT} w^i}{NT} \)
Online Training Algorithm

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

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

$i = 0$

1: $\textbf{for } n = 1, \ldots, N \textbf{ do}$
2: $\textbf{for } t = 1, \ldots, T \textbf{ do}$
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: $\textbf{end for}$
8: $\textbf{end for}$
9: $\text{return } \frac{\sum_{i=1}^{NT} w^i}{NT}$
Online Large-Margin Training

$$\hat{w}^{i+1} = \arg\min_{w^{i+1}} \frac{1}{2} \|w^{i+1} - w^i\|^2$$

subject to

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

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

$$\forall \hat{e} \in O^t, \forall e' \in C^t$$
Online Large-Margin Training

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

subject to

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

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

\[
\forall \hat{e} \in O^t, \forall e' \in C^t
\]
Online Large-Margin Training

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

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.
Parameter Updates

\[ w^{i+1} = w^i + \sum_{\hat{e}, e'} \alpha(\hat{e}, e') \left( h(f^t, \hat{e}) - h(f^t, e') \right) \]
Parameter Updates

\[ w^{i+1} = w^i + \sum_{\hat{e}, e'} \alpha(\hat{e}, e') \left( h(f^t, \hat{e}) - h(f^t, e') \right) \]

• This work (I-oracle + I-best)

\[ \alpha = \max \left( 0, \min \left( C, \frac{L(\hat{e}, e'; e^t) - (s^i(f^t, \hat{e}) - s^i(f^t, e'))}{\| h(f^t, \hat{e}) - h(f^t, e') \|^2} \right) \right) \]
Parameter Updates

\[ w^{i+1} = w^i + \sum_{\hat{e}, e'} \alpha(\hat{e}, e') \left( h(f^t, \hat{e}) - h(f^t, e') \right) \]

- This work (I-oracle + I-best)
  \[ \alpha = \max \left( 0, \min \left( C, \frac{L(\hat{e}, e'; e^t) - (s^i(f^t, \hat{e}) - s^i(f^t, e'))}{\| h(f^t, \hat{e}) - h(f^t, e') \|^2} \right) \right) \]

- Perceptron Training (Liang et al., 2006)
  \[ \alpha = 1 \]
Parameter Updates

\[ w^{i+1} = w^i + \sum_{\hat{e}, e'} \alpha(\hat{e}, e') \left( h(f^t, \hat{e}) - h(f^t, e') \right) \]

- This work (I-oracle + I-best)
  \[ \alpha = \max \left( 0, \min \left( C, \frac{L(\hat{e}, e'; e^t) - \left( s^i(f^t, \hat{e}) - s^i(f^t, e') \right)}{\| h(f^t, \hat{e}) - h(f^t, e') \|^2} \right) \right) \]

- Perceptron Training (Liang et al., 2006)
  \[ \alpha = 1 \]

- SGD Training (Tillmann and Zhand, 2006)
  \[ \alpha = \eta L(\hat{e}, e'; e^t) \cdot \max \left( 0, 1 - \left( s^i(f^t, \hat{e}) - s^i(f^t, e') \right) \right) \]
Approximated BLEU
Approximated BLEU

- Document-BLEU or sentence-BLEU?

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

- 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.
Evaluation
Evaluation

• A standard NIST Arabic/English Translation
  • Hierarchical phrases from 3.8M sentences
  • 5-gram from English Gigaword
• Trained on MT 2003, tested on MT 2004/2005
Evaluation

- A standard NIST Arabic/English Translation
  - Hierarchical phrases from 3.8M sentences
  - 5-gram from English Gigaword
  - Trained on MT 2003, tested on MT 2004/2005
- Experiments on (10-oracle, 10-best, 50 iterations):
  - Token-types
  - Structural features
  - m-oracle + k-best constraints
Evaluation

- A standard NIST Arabic/English Translation
  - Hierarchical phrases from 3.8M sentences
  - 5-gram from English Gigaword
  - Trained on MT 2003, tested on MT 2004/2005
- Experiments on (10-oracle, 10-best, 50 iterations):
  - Token-types
  - Structural features
  - m-oracle + k-best constraints
- 0.5M to 14M active features
Results (BLEU)
Results (BLEU)

- surface form
- w/ prefix/suffix
- w/ word class
- w/ digits
- All tokens

2003(dev) 2004 2005
Results (BLEU)

- surface form
- w/ prefix/suffix
- w/ word class
- w/ digits
- All tokens

2003(dev) 2004 2005
Results (BLEU)

- surface form
- w/ prefix/suffix
- w/ word class
- w/ digits
- All tokens

2003 (dev) 2004 2005
Results (BLEU)

- surface form
- w/ prefix/suffix
- w/ word class
- w/ digits
- All tokens

Year: 2003 (dev), 2004, 2005
Results (BLEU)

(surface form)

w/ prefix/suffix

w/ word class

w/ digits

All tokens

2003(dev) 2004 2005
Results (BLEU)

- surface form
- w/ prefix/suffix
- w/ word class
- w/ digits
- All tokens
Results (BLEU)

- surface form
- w/ prefix/suffix
- w/ word class
- w/ digits
- All tokens

Word pairs:
- + target bigram
- + insertion
- + hierarchical

Years: 2003 (dev) 2004 2005
Results (BLEU)

- surface form
- w/ prefix/suffix
- w/ word class
- w/ digits
- All tokens

- word pairs
- + target bigram
- + insertion
- + hierarchical

Year: 2003 (dev), 2004, 2005
Results (BLEU)

- surface form
- w/ prefix/suffix
- w/ word class
- w/ digits
- All tokens
- word pairs
- + target bigram
- + insertion
- + hierarchical
Results (BLEU)

- surface form
- w/ prefix/suffix
- w/ word class
- w/ digits
- All tokens

word pairs
- + target bigram
- + insertion
- + hierarchical

2003(dev) 2004 2005
2003(dev) 2004 2005
Results (BLEU)

- surface form
- w/ prefix/suffix
- w/ word class
- w/ digits
- All tokens

- word pairs
- + target bigram
- + insertion
- + hierarchical

Years:
- 2003 (dev)
- 2004
- 2005

Comparison across years and features:
- Surface form shows a significant increase from 2003 to 2005.
- Features like word pairs, target bigram, insertion, and hierarchical show variations across years.
Results (BLEU)

-.surface form
- w/ prefix/suffix
- w/ word class
- w/ digits
- All tokens
- word pairs
- + target bigram
- + insertion
- + hierarchical
- baseline (MERT)
- 1-oracle 1-best
- 1-oracle 10-best
- 10-oracle 1-best
- 10-oracle 10-best
- sentence-BLEU


2003(dev) 2004 2005

45 50 55 60 65
Results (BLEU)

- surface form
- w/ prefix/suffix
- w/ word class
- w/ digits
- All tokens
- word pairs
- + target bigram
- + insertion
- + hierarchical
- baseline (MERT)
- 1-oracle 1-best
- 1-oracle 10-best
- 10-oracle 1-best
- 10-oracle 10-best
- sentence-BLEU

Years: 2003(dev), 2004, 2005
Two-fold cross validation

<table>
<thead>
<tr>
<th></th>
<th>closed test</th>
<th>open test</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>NIST</td>
<td>BLEU</td>
</tr>
<tr>
<td>baseline</td>
<td>10.71</td>
<td>44.79</td>
</tr>
<tr>
<td>online</td>
<td>11.58</td>
<td>53.42</td>
</tr>
</tbody>
</table>
Summary

- Online Large-Margin Training (This work)
  - Memorized local update strategy
  - Approximated BLEU
- SGD Training (Tillmann and Zhang, 2006)
  - Precomputed oracles/no real valued features.
- Perceptron Training (Liang et al., 2006)
  - Local update strategy
Conclusion

• Exploited only a small data set for millions of features:
  • Easy to explore alternative features, such as POS/NE etc.

• Future work:
  • Larger data + more features.