Fast and Adaptive Online Training of Feature-Rich Translation Models

Spence Green  Sida Wang

Daniel Cer  Christopher D. Manning

Stanford University

ACL 2013
<table>
<thead>
<tr>
<th>Feature-Rich Research</th>
<th>Industry/Evaluations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Liang et al. 2006</td>
<td></td>
</tr>
<tr>
<td>Tillmann and Zhang 2006</td>
<td></td>
</tr>
<tr>
<td>Arun and Koehn 2007</td>
<td></td>
</tr>
<tr>
<td>Ittycheriah and Roukos 2007</td>
<td></td>
</tr>
<tr>
<td>Watanabe et al. 2007</td>
<td></td>
</tr>
<tr>
<td>Chiang et al. 2008; Chiang et al. 2009</td>
<td>$n$-best/lattice MERT</td>
</tr>
<tr>
<td>Haddow et al. 2011</td>
<td>MIRA (ISI)</td>
</tr>
<tr>
<td>Hopkins and May 2011</td>
<td></td>
</tr>
<tr>
<td>Xiang and Ittycheriah 2011</td>
<td></td>
</tr>
<tr>
<td>Cherry and Foster 2012</td>
<td></td>
</tr>
<tr>
<td>Chiang 2012</td>
<td></td>
</tr>
<tr>
<td>Gimpel 2012</td>
<td></td>
</tr>
<tr>
<td>Simianer et al. 2012</td>
<td></td>
</tr>
<tr>
<td>Watanabe 2012</td>
<td></td>
</tr>
</tbody>
</table>
### Feature-Rich Research

- Liang et al. 2006
- Tillmann and Zhang 2006
- Arun and Koehn 2007
- Ittycheriah and Roukos 2007
- Watanabe et al. 2007
- Chiang et al. 2008; Chiang et al. 2009

### Industry/Evaluations

- Hopkins and May 2011
- Xiang and Ittycheriah 2011
- Cherry and Foster 2012
- Chiang 2012
- Gimpel 2012
- Simianer et al. 2012
- Watanabe 2012

- **n-best/lattice MERT**
- **MIRA (ISI)**
Feature-rich Shared Task Submissions

<table>
<thead>
<tr>
<th>Year</th>
<th>Task</th>
<th>Feature-rich</th>
</tr>
</thead>
<tbody>
<tr>
<td>2012</td>
<td>WMT</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>IWSLT</td>
<td>1</td>
</tr>
<tr>
<td>2013</td>
<td>WMT</td>
<td>2 ?</td>
</tr>
<tr>
<td></td>
<td>IWSLT</td>
<td>TBD</td>
</tr>
</tbody>
</table>
Speculation: Entrenchment Of MERT

Feature-rich on small tuning sets?

Implementation complexity

Open source availability
Speculation: Entrenchment Of MERT

Feature-rich on small tuning sets?

Implementation complexity

Open source availability

Top-selling phone of 2003
Motivation: Why Feature-Rich MT?

Make MT more like other machine learning settings

Features for specific errors

Domain adaptation
Motivation: Why Online MT Tuning?

Search: decode more often

Better solutions
See: [Liang and Klein 2009]

Computer-aided translation: incremental updating
Benefits Of Our Method

Fast and scalable

Adapts to dense/sparse feature mix

Not complicated
Online Algorithm Overview

Updating with an adaptive learning rate

Automatic feature selection via $L_1$ regularization

Loss function: Pairwise ranking
Notation

\( t \)

- time/update step
<table>
<thead>
<tr>
<th>Notation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t$</td>
<td>time/update step</td>
</tr>
<tr>
<td>$\omega_t$</td>
<td>weight vector in $\mathbb{R}^n$</td>
</tr>
</tbody>
</table>
Notation

\( t \) time/update step

\( w_t \) weight vector in \( \mathbb{R}^n \)

\( \eta \) learning rate
<table>
<thead>
<tr>
<th>Notation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t$</td>
<td>time/update step</td>
</tr>
<tr>
<td>$w_t$</td>
<td>weight vector in $\mathbb{R}^n$</td>
</tr>
<tr>
<td>$\eta$</td>
<td>learning rate</td>
</tr>
<tr>
<td>$l_t(w)$</td>
<td>loss of $t$’th example</td>
</tr>
</tbody>
</table>
### Notation

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t$</td>
<td>time/update step</td>
</tr>
<tr>
<td>$\omega_t$</td>
<td>weight vector in $\mathbb{R}^n$</td>
</tr>
<tr>
<td>$\eta$</td>
<td>learning rate</td>
</tr>
<tr>
<td>$l_t(\omega)$</td>
<td>loss of $t$’th example</td>
</tr>
<tr>
<td>$z_{t-1} \in \partial l_t(\omega_{t-1})$</td>
<td>subgradient set (subdifferential)</td>
</tr>
</tbody>
</table>
Notation

\( t \)  \hspace{1cm} \text{time/update step}

\( w_t \)  \hspace{1cm} \text{weight vector in } \mathbb{R}^n

\( \eta \)  \hspace{1cm} \text{learning rate}

\( l_t(w) \)  \hspace{1cm} \text{loss of } t\text{'th example}

\( z_{t-1} \in \partial l_t(w_{t-1}) \)  \hspace{1cm} \text{subgradient set} (\text{subdifferential})

\( z_{t-1} = \nabla l_t(w_{t-1}) \)  \hspace{1cm} \text{for differentiable loss functions}
Notation

\( t \) \hspace{0.5cm} \text{time/update step}

\( w_t \) \hspace{0.5cm} \text{weight vector in } \mathbb{R}^n

\( \eta \) \hspace{0.5cm} \text{learning rate}

\( \ell_t(w) \) \hspace{0.5cm} \text{loss of } t\text{th example}

\( z_{t-1} \in \partial \ell_t(w_{t-1}) \) \hspace{0.5cm} \text{subgradient set (subdifferential)}

\( z_{t-1} = \nabla \ell_t(w_{t-1}) \) \hspace{0.5cm} \text{for differentiable loss functions}

\( r(w) \) \hspace{0.5cm} \text{regularization function}
Warm-up: Stochastic Gradient Descent

Per-instance update:

\[ w_t = w_{t-1} - \eta z_{t-1} \]

**Issue #1:** learning rate schedule

\[ \eta / t ? \]
Warm-up: Stochastic Gradient Descent

Per-instance update:

\[ w_t = w_{t-1} - \eta z_{t-1} \]

Issue #1: learning rate schedule

\[ \eta / t ? \]

\[ \eta / \sqrt{t} ? \]
Warm-up: Stochastic Gradient Descent

Per-instance update:

\[ w_t = w_{t-1} - \eta z_{t-1} \]

**Issue #1:** learning rate schedule

\[ \eta / t? \]

\[ \eta / \sqrt{t}? \]

\[ \eta / (1 + \gamma t)? \quad \text{Yuck.} \]
Warm-up: Stochastic Gradient Descent

SGD update:

\[ w_t = w_{t-1} - \eta Z_{t-1} \]

Issue #2: same step size for every coordinate

Intuitively, we might want:

Frequent feature: small steps e.g. \( \eta / t \)

Rare feature: large steps e.g. \( \eta / \sqrt{t} \)
SGD: Learning Rate Adaptation

SGD update:

\[ w_t = w_{t-1} - \eta Z_{t-1} \]

Scale learning rate with \( A^{-1} \in \mathbb{R}^{n \times n} \):

\[ w_t = w_{t-1} - \eta A^{-1} z_{t-1} \]

Choices:

\[ A^{-1} = I \quad \text{(SGD)} \]
SGD: Learning Rate Adaptation

SGD update:

\[ w_t = w_{t-1} - \eta Z_{t-1} \]

Scale learning rate with \( A^{-1} \in \mathbb{R}^{n \times n} \):

\[ w_t = w_{t-1} - \eta A^{-1} z_{t-1} \]

Choices:

- \( A^{-1} = I \) (SGD)
- \( A^{-1} = H^{-1} \) (Batch: Newton step)
AdaGrad

Update:

\[ w_t = w_{t-1} - \eta A^{-1} z_{t-1} \]

Set \( A^{-1} = G_t^{-1/2} \):

\[ G_t = G_{t-1} + z_{t-1} \cdot z_{t-1}^\top \]
AdaGrad: Approximations and Intuition

For high-dimensional $w_t$, use diagonal $G_t$

$$w_t = w_{t-1} - \eta G_t^{-1/2} z_{t-1}$$

Intuition:

1/√$t$ schedule on constant gradient

Small steps for frequent features

Big steps for rare features

[Duchi et al. 2011]
AdaGrad vs. SGD: 2D Illustration
Feature Selection

Traditional approach: frequency cutoffs

Unattractive for large tuning sets (e.g. bitext)

More principled: $L_1$ regularization

$$r(w) = \sum_i |w_i|$$
Feature Selection: FOBOS

Two-step update:

\[ w_{t-\frac{1}{2}} = w_{t-1} - \eta Z_{t-1} \quad (1) \]

\[
\begin{align*}
    w_t &= \arg \min_{w} \left( \frac{1}{2} \left\| w - w_{t-\frac{1}{2}} \right\|^2 + \frac{1}{2} \left\| w_{t-\frac{1}{2}} \right\|^2 \right) + \lambda \cdot r(w) \\
    &= \text{proximal term} + \text{regularization} \quad (2)
\end{align*}
\]

[Duchi and Singer 2009]

Extension: AdaGrad update in step (1)
Feature Selection: FOBOS

For $L_1$, FOBOS becomes soft thresholding:

$$w_t = \text{sign}(w_{t-\frac{1}{2}}) \left[ \left| w_{t-\frac{1}{2}} \right| - \lambda \right]_+$$

Squared-$L_2$ also has a simple form
Feature Selection: Lazy Regularization

Lazy updating: only update active coordinates

Big speedup in MT setting

Easy with FOBOS:

\[ t'_j : \text{last update of dimension } j \]

Use \( \lambda(t - t'_j) \)
AdaGrad+FOBOS: Full Algorithm

1. Additive update: $G_t$
AdaGrad+FOBOS: Full Algorithm

1. Additive update: $G_t$

2. Additive update: $w_{t-\frac{1}{2}}$
AdaGrad+FOBOS: Full Algorithm

1. Additive update: $G_t$

2. Additive update: $w_{t-\frac{1}{2}}$

3. Closed-form regularization: $w_t$
AdaGrad+FOBOS: Full Algorithm

1. Additive update: $G_t$
2. Additive update: $w_{t-\frac{1}{2}}$
3. Closed-form regularization: $w_t$

Not complicated

Very fast
Recap: Pairwise Ranking

For derivation \( d \), feature map \( \phi(d) \), references \( e^{1:k} \)

Metric: \( B(d, e^{1:k}) \)  

(e.g. BLEU+1)

Model score: \( M(d) = w \cdot \phi(d) \)

Pairwise consistency:

\[
M(d_+) > M(d_-) \iff B(d_+, e^{1:k}) > B(d_-, e^{1:k})
\]

[Hopkins and May 2011]
Loss Function: Pairwise Ranking

\[ M(d_+) > M(d_-) \iff w \cdot (\phi(d_+) - \phi(d_-)) > 0 \]

Loss formulation:

Difference vector: \( \nu = \phi(d_+) - \phi(d_-) \)

Find \( w \) so that \( w \cdot \nu > 0 \)

**Binary classification problem** between \( \nu \) and \( -\nu \)

**Logistic loss**: convex, differentiable

[Hopkins and May 2011]
Parallelization

Online algorithms are inherently sequential

Out-of-order updating:

\[ w_7 = w_6 - \eta z_4 \]
\[ w_8 = w_7 - \eta z_6 \]
\[ w_9 = w_8 - \eta z_5 \]
Parallelization

Online algorithms are inherently sequential

Out-of-order updating:

\[ w_7 = w_6 - \eta z_4 \]
\[ w_8 = w_7 - \eta z_6 \]
\[ w_9 = w_8 - \eta z_5 \]

Low-latency regret bound: \( O(\sqrt{T}) \) \[\text{[Langford et al. 2009]}\]
Translation Quality Experiments

Arabic-English (Ar–En) and Chinese-English (Zh–En)

Newswire and mixed-genre experiments

BOLT bitexts: data up to 2012

<table>
<thead>
<tr>
<th></th>
<th>Bilingual</th>
<th>Monolingual</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Sentences</td>
<td>Tokens</td>
</tr>
<tr>
<td>Ar–En</td>
<td>6.6M</td>
<td>375M</td>
</tr>
<tr>
<td>Zh–En</td>
<td>9.3M</td>
<td>538M</td>
</tr>
</tbody>
</table>
MT System

Phrase-based MT: Phrasal
[Cer et al. 2010]

Dense baseline: MERT
Cer et al. 2008 line search
Accumulates $n$-best lists
Random starting points, etc.
Feature-Rich Baseline: PRO

Pairwise Ranking Optimization (PRO)

Batch log loss minimization

Phrasal implementation:

L-BFGS with $L_2$ regularization

[Hopkins and May 2011]

Sanity check: Moses PRO and kb-MIRA (batch) implementations
Dense Features

8 Hierarchical lex. reordering
Dense Features

8 Hierarchical lex. reordering
5 Moses phrase table features
1 Rule bitext count
1 Unique rule indicator
Dense Features

8  Hierarchical lex. reordering
5  Moses phrase table features
1  Rule bitext count
1  Unique rule indicator
1  Word penalty
1  Linear distortion
1  LM
1  Unknown word

19
Sparse Feature Templates

**Discriminative Phrase Table (PT)**

Rule indicator: \( \implies \) برنامه الفضاء \( \Rightarrow \) space program

**Discriminative Alignments (AL)**

Source word deletion: 

Word alignments:

\( \implies \) الفضاء \( \Rightarrow \) space

**Discriminative Lex. Reordering (LO)**

Phrase orientation: 

\( \text{swap}(\text{الفضاء} \Rightarrow \text{space}) \)
Evaluation: NIST OpenMT

Small tuning set: MT06

“Large” tuning set: MT0568 (≈4200 segments)

BLEU-4 uncased, Four references

Paper: mixed genre (bitext) experiments
<table>
<thead>
<tr>
<th></th>
<th>Ar–En</th>
<th>Zh–En</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Tune</td>
<td>Test Avg.</td>
</tr>
<tr>
<td>MERT</td>
<td>45.08</td>
<td>50.51</td>
</tr>
<tr>
<td>This paper</td>
<td>43.16</td>
<td>50.11</td>
</tr>
</tbody>
</table>
## Results: Add More Features

<table>
<thead>
<tr>
<th></th>
<th>Ar–En</th>
<th>Zh–En</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Tune</td>
<td>Test Avg.</td>
</tr>
<tr>
<td>MERT—Dense</td>
<td>45.08</td>
<td>50.51</td>
</tr>
<tr>
<td>This paper +PT</td>
<td>50.61</td>
<td>50.52</td>
</tr>
</tbody>
</table>
# Results: Add More Features

<table>
<thead>
<tr>
<th></th>
<th>Ar–En</th>
<th>Zh–En</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Tune</td>
<td>Test Avg.</td>
</tr>
<tr>
<td>MERT—Dense</td>
<td>45.08</td>
<td>50.51</td>
</tr>
<tr>
<td>This paper +PT</td>
<td>50.61</td>
<td>50.52</td>
</tr>
<tr>
<td>This paper +All</td>
<td>60.85</td>
<td>50.97</td>
</tr>
</tbody>
</table>

(MT06 tuning set)
### Results: Add More Data

<table>
<thead>
<tr>
<th></th>
<th>Ar–En</th>
<th>Zh–En</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Test Avg.</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MERT—mt06</td>
<td>50.51</td>
<td>34.49</td>
</tr>
<tr>
<td>MERT—mt0568</td>
<td>50.74</td>
<td>34.55</td>
</tr>
</tbody>
</table>

This paper + All—mt0568 does worse than MERT—mt0568 + All worse than MERT—mt0568.
## Results: Add More Data

<table>
<thead>
<tr>
<th></th>
<th>Ar–En</th>
<th>Zh–En</th>
<th>Test Avg.</th>
<th>Test Avg.</th>
</tr>
</thead>
<tbody>
<tr>
<td>MERT—mt06</td>
<td>50.51</td>
<td></td>
<td>34.49</td>
<td></td>
</tr>
<tr>
<td>MERT—mt0568</td>
<td>50.74</td>
<td></td>
<td>34.55</td>
<td></td>
</tr>
<tr>
<td>This paper</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>+All—mt06</td>
<td>50.97</td>
<td></td>
<td>35.31</td>
<td></td>
</tr>
</tbody>
</table>
## Results: Add More Data

<table>
<thead>
<tr>
<th></th>
<th>Ar–En</th>
<th></th>
<th>Zh–En</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Test Avg.</strong></td>
<td></td>
<td><strong>Test Avg.</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MERT—mt06</td>
<td>50.51</td>
<td></td>
<td>34.49</td>
<td></td>
</tr>
<tr>
<td>MERT—mt0568</td>
<td>50.74</td>
<td></td>
<td>34.55</td>
<td></td>
</tr>
<tr>
<td><strong>This paper</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>+All—mt06</td>
<td>50.97</td>
<td></td>
<td>35.31</td>
<td></td>
</tr>
<tr>
<td>+All—mt0568</td>
<td><strong>52.34</strong></td>
<td>+1.60</td>
<td><strong>36.61</strong></td>
<td>+2.06</td>
</tr>
</tbody>
</table>

PRO+All worse than MERT—mt0568
### Analysis: Zh–En MT06 Tuning

(16 threads) | Epochs | Min/epoch |
---|---|---|
MERT Dense | 22 | 180 |

This paper takes about 5 days.
## Analysis: Zh–En MT06 Tuning

<table>
<thead>
<tr>
<th>(16 threads)</th>
<th>Epochs</th>
<th>Min/epoch</th>
</tr>
</thead>
<tbody>
<tr>
<td>MERT</td>
<td>Dense</td>
<td>22</td>
</tr>
<tr>
<td>PRO</td>
<td>+PT</td>
<td>25</td>
</tr>
<tr>
<td>kb-MIRA*</td>
<td>+PT</td>
<td>26</td>
</tr>
<tr>
<td>This paper</td>
<td>+PT</td>
<td>10</td>
</tr>
</tbody>
</table>

MERT—mt0568 tuning takes about 5 days.
## Analysis: Zh–En MT06 Tuning

<table>
<thead>
<tr>
<th></th>
<th>(16 threads)</th>
<th>Epochs</th>
<th>Min/epoch</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>MERT</strong></td>
<td>Dense</td>
<td>22</td>
<td>180</td>
</tr>
<tr>
<td><strong>PRO</strong></td>
<td>+PT</td>
<td>25</td>
<td>35</td>
</tr>
<tr>
<td><strong>kb-MIRA</strong></td>
<td>+PT</td>
<td>26</td>
<td>25</td>
</tr>
<tr>
<td><strong>This paper</strong></td>
<td>+PT</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td><strong>PRO</strong></td>
<td>+All</td>
<td>13</td>
<td>100</td>
</tr>
<tr>
<td><strong>This paper</strong></td>
<td>+All</td>
<td>5</td>
<td>15</td>
</tr>
</tbody>
</table>

MERT—mt0568 tuning takes about 5 days
Analysis: Runtime

Online regret bounds depend on # updates

Large datasets: more updates per epoch

Fewer epochs to converge

Lazy updating helps:

\[ w_t \approx 100k \text{ features} \]

\[ z_{t-1} \approx 500 \text{ features} \]
Analysis: Reordering

Arabic matrix clauses often **verb-initial**

Manually selected 208 verb-initial segments (MT09)

32 differed for MERT–Dense vs. +All
## Analysis: Reordering

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>+All correct</td>
<td>18</td>
<td>56.3%</td>
</tr>
<tr>
<td>MERT–Dense correct</td>
<td>4</td>
<td>12.5%</td>
</tr>
<tr>
<td>Both wrong</td>
<td>10</td>
<td>31.3%</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>32</strong></td>
<td></td>
</tr>
</tbody>
</table>

**Ref:** The newspaper and television reported

- MERT: She said the newspaper and television
- +All: Television and newspaper said
Analysis: Domain Adaptation

برنامج $\Rightarrow$ program, programme

<table>
<thead>
<tr>
<th></th>
<th># bitext–5k</th>
<th># MT0568</th>
</tr>
</thead>
<tbody>
<tr>
<td>programme</td>
<td>185</td>
<td>0</td>
</tr>
<tr>
<td>program</td>
<td>19</td>
<td>449</td>
</tr>
</tbody>
</table>
Analysis: Domain Adaptation

برنامه $\Rightarrow$ *program, programme*

<table>
<thead>
<tr>
<th></th>
<th># bitext–5k</th>
<th># MT0568</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>programme</em></td>
<td>185</td>
<td>0</td>
</tr>
<tr>
<td><em>program</em></td>
<td>19</td>
<td>449</td>
</tr>
</tbody>
</table>

$\pm$PT rules: *programme* 353 79

$\pm$PT rules: *program* 9 31
Caveats and Next Steps

Single-reference setting

BLEU+1 is unreliable

Lexicalized features cause overfitting

Current work

Bitext tuning

Different loss function
Conclusion

Fast, adaptive, online tuning for MT

Easy to implement

Works as well as MERT for Dense

Sane feature engineering
Fast and Adaptive Online Training of Feature-Rich Translation Models

Spence Green        Sida Wang

Daniel Cer         Christopher D. Manning

Stanford University

Try the code in Phrasal:

nlp.stanford.edu/software/phrasal/
En–De Learning Curve

![Graph showing BLEU scores for different models over epochs.](image-url)
Sparse Features: Negative Results

<table>
<thead>
<tr>
<th>Feature Type</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Discriminative LM</td>
<td><strong>Jane</strong> called <strong>Sally</strong></td>
</tr>
<tr>
<td>Phrase boundary features</td>
<td><strong>Jane</strong></td>
</tr>
<tr>
<td>Alignment constellation</td>
<td><strong>1-0 0-1</strong></td>
</tr>
<tr>
<td>Target word insertion</td>
<td>**Jane called the <strong>Sally</strong></td>
</tr>
</tbody>
</table>