Fast and Adaptive Online Training of Feature-Rich Translation Models

Spence Green    Sida Wang

Daniel Cer    Christopher D. Manning

Stanford University

ACL 2013
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

n-best/lattice MERT

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

MIRA (ISI)
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
Haddow et al. 2011
Hopkins and May 2011
Xiang and Ittycheriah 2011
Cherry and Foster 2012
Chiang 2012
Gimpel 2012
Simianer et al. 2012
Watanabe 2012

Industry/Evaluations
$n$-best/lattice MERT
MIRA (ISI)
# Feature-rich Shared Task Submissions

<table>
<thead>
<tr>
<th>Year</th>
<th>Event</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
Notation

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

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

\( t \) \quad \text{time/update step}

\( \omega_t \) \quad \text{weight vector in } \mathbb{R}^n

\( \eta \) \quad \text{learning rate}
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}} \text{ example}
### 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>$w_t$</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>
<tr>
<td>$z_{t-1} \in \partial l_t(w_{t-1})$</td>
<td>subgradient set (subdifferential)</td>
</tr>
</tbody>
</table>
Notation

\[ t \quad \text{time/update step} \]

\[ \mathbf{w}_t \quad \text{weight vector in } \mathbb{R}^n \]

\[ \eta \quad \text{learning rate} \]

\[ l_t(\mathbf{w}) \quad \text{loss of } t\text{'th example} \]

\[ z_{t-1} \in \partial l_t(\mathbf{w}_{t-1}) \quad \text{subgradient set (subdifferential)} \]

\[ z_{t-1} = \nabla l_t(\mathbf{w}_{t-1}) \quad \text{for differentiable loss functions} \]
Notation

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

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

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

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

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

\( z_{t-1} = \nabla l_t(\mathbf{w}_{t-1}) \) \hspace{1cm} \text{for differentiable loss functions}

\( r(\mathbf{w}) \) \hspace{1cm} \text{regularization function}
Warm-up: Stochastic Gradient Descent

Per-instance update:

\[ w_t = w_{t-1} - \eta z_{t-1} \]
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
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 \( A^{-1} \in \mathbb{R}^{n \times n} \):

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

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

\[ A^{-1} = H^{-1} \quad \text{(Batch: Newton step)} \]
AdaGrad

Duchi et al. 2011

Update:

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

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

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

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

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

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

\[ w_t = \arg \min_{w} \left( \frac{1}{2} \left\| w - w_{t-\frac{1}{2}} \right\|^2 + \frac{\lambda}{2} \cdot r(w) \right) \] (2)

[Duchi and Singer 2009]
Feature Selection: FOBOS

Two-step update:

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

\[ w_t = \arg \min _w \left( \frac{1}{2} \left\| w - w_{t-\frac{1}{2}} \right\|^2 + \lambda \cdot r(w) \right) \]  \hspace{1cm} (2)

[Duchi and Singer 2009]

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

For $L_1$, FOBOS becomes **soft thresholding**:

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

3. Closed-form regularization: $\mathcal{W}_t$
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: $M(d) = w \cdot \phi(d)$
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:

$$M(d_+) > M(d_-) \iff B\left(d_+, e^{1:k}\right) > B\left(d_-, e^{1:k}\right)$$

[Hopkins and May 2011]
Loss Function: Pairwise Ranking

\[ M(d_+) > M(d_-) \iff w \cdot (\phi(d_+) - \phi(d_-)) > 0 \]
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 \( -\nu \)
Loss Function: Pairwise Ranking

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

Loss formulation:

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

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

Binary classification problem between \( v \) and \( -v \)

Logistic loss: convex, differentiable

[Hopkins and May 2011]
Parallelization

Online algorithms are inherently sequential
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: \( O(\sqrt{T}) \)  
[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]
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

<table>
<thead>
<tr>
<th></th>
<th>Feature</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>Hierarchical lex. reordering</td>
</tr>
<tr>
<td>5</td>
<td>Moses phrase table features</td>
</tr>
<tr>
<td>1</td>
<td>Rule bitext count</td>
</tr>
<tr>
<td>1</td>
<td>Unique rule indicator</td>
</tr>
<tr>
<td>1</td>
<td>Word penalty</td>
</tr>
<tr>
<td>1</td>
<td>Linear distortion</td>
</tr>
<tr>
<td>1</td>
<td>LM</td>
</tr>
<tr>
<td>1</td>
<td>Unknown word</td>
</tr>
</tbody>
</table>

**Total:** 19
Sparse Feature Templates

Discriminative Phrase Table (PT)

Rule indicator:  \( \overline{\text{برنامج الفضاء}} \Rightarrow \text{space program} \)
Sparse Feature Templates

Discriminative Phrase Table (PT)

Rule indicator: BarButton الفضاء \( \Rightarrow \) space program

Discriminative Alignments (AL)

Source word deletion:

Word alignments:
Sparse Feature Templates

Discriminative Phrase Table (PT)

Rule indicator: \( \Rightarrow \) space program

Discriminative Alignments (AL)

Source word deletion: \( \Rightarrow \) space

Word alignments: \( \Rightarrow \) space

Discriminative Lex. Reordering (LO)

Phrase orientation: \( \Rightarrow \) (swap) space
Evaluation: NIST OpenMT

Small tuning set: MT06

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

BLEU-4 uncased, Four references
Evaluation: NIST OpenMT

Small tuning set: MT06

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

BLEU-4 uncased, Four references

Paper: mixed genre (bitext) experiments
## Results: Small Tuning Set (Dense)

<table>
<thead>
<tr>
<th></th>
<th>Ar–En</th>
<th></th>
<th>Zh–En</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Tune</td>
<td>Test Avg.</td>
<td>Tune</td>
<td>Test Avg.</td>
</tr>
<tr>
<td>MERT</td>
<td>45.08</td>
<td>50.51</td>
<td>33.73</td>
<td>34.49</td>
</tr>
<tr>
<td>This paper</td>
<td>43.16</td>
<td>50.11</td>
<td>32.20</td>
<td>35.25</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>Ar–En</th>
<th>Zh–En</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test Avg.</td>
<td>Test Avg.</td>
</tr>
<tr>
<td>MERT—mt06</td>
<td>50.51</td>
</tr>
<tr>
<td>MERT—mt0568</td>
<td>50.74</td>
</tr>
</tbody>
</table>

This paper + All—mt06 50.97 35.31 + All—mt0568 52.34 1.60 36.61 2.06

PRO + All worse than MERT—mt0568
## Results: Add More Data

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

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

*PRO: +All worse than MERT—mt0568*
## Results: Add More Data

<table>
<thead>
<tr>
<th></th>
<th>Ar–En</th>
<th>Zh–En</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test Avg.</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>
<tr>
<td>This paper</td>
<td></td>
<td></td>
</tr>
<tr>
<td>+All—mt06</td>
<td>50.97</td>
<td>35.31</td>
</tr>
<tr>
<td>+All—mt0568</td>
<td><strong>52.34</strong></td>
<td><strong>36.61</strong></td>
</tr>
<tr>
<td>PRO</td>
<td></td>
<td></td>
</tr>
<tr>
<td>+All worse than MERT—mt0568</td>
<td>+1.60</td>
<td>+2.06</td>
</tr>
</tbody>
</table>
**Analysis: Zh–En MT06 Tuning**

<table>
<thead>
<tr>
<th>(16 threads)</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>MERT</strong></td>
<td>Dense</td>
<td>22</td>
<td>180</td>
</tr>
</tbody>
</table>

This paper takes about 5 days to complete.
## 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 Dense</td>
<td>22</td>
<td>180</td>
</tr>
<tr>
<td>PRO +PT</td>
<td>25</td>
<td>35</td>
</tr>
<tr>
<td>kb-MIRA* +PT</td>
<td>26</td>
<td>25</td>
</tr>
<tr>
<td>This paper +PT</td>
<td>10</td>
<td>10</td>
</tr>
</tbody>
</table>

MERT—mt0568 tuning 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 Dense</td>
<td>22</td>
<td>180</td>
</tr>
<tr>
<td>PRO +PT</td>
<td>25</td>
<td>35</td>
</tr>
<tr>
<td>kb-MIRA* +PT</td>
<td>26</td>
<td>25</td>
</tr>
<tr>
<td>This paper +PT</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>PRO +All</td>
<td>13</td>
<td>100</td>
</tr>
<tr>
<td>This paper +All</td>
<td>5</td>
<td>15</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>MERT</td>
<td>Dense</td>
<td>22</td>
<td>180</td>
</tr>
<tr>
<td>PRO</td>
<td>+PT</td>
<td>25</td>
<td>35</td>
</tr>
<tr>
<td>kb-MIRA*</td>
<td>+PT</td>
<td>26</td>
<td>25</td>
</tr>
<tr>
<td>This paper</td>
<td>+PT</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>PRO</td>
<td>+All</td>
<td>13</td>
<td>100</td>
</tr>
<tr>
<td>This paper</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
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
Analysis: Reordering

Arabic matrix clauses often verb-initial

Manually selected 208 verb-initial segments (MT09)
Analysis: Reordering

Arabic matrix clauses often verb-initial

Manually selected 208 verb-initial segments (MT09)

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

<table>
<thead>
<tr>
<th>Category</th>
<th>Count</th>
<th>Percentage</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>
</tbody>
</table>

32
Analysis: Reordering

<table>
<thead>
<tr>
<th>Category</th>
<th>Count</th>
<th>Percentage</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>32</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
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>
Analysis: Domain Adaptation

برنامج ⇒ 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>
<tr>
<td>+PT rules: programme</td>
<td>353</td>
<td>79</td>
</tr>
<tr>
<td>+PT rules: program</td>
<td>9</td>
<td>31</td>
</tr>
</tbody>
</table>
Caveats and Next Steps

Single-reference setting

BLEU+1 is unreliable

Lexicalized features cause overfitting
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
Conclusion

Fast, adaptive, online tuning for MT

Easy to implement
Conclusion

Fast, adaptive, online tuning for MT

Easy to implement

Works as well as MERT for Dense
Conclusion

Fast, adaptive, online tuning for MT

Easy to implement

Works as well as MERT for Dense

Sane feature engineering
En–De Learning Curve

![Learning Curve Graph]

- **Model**
  - **dense**
  - **feature-rich**

- **BLEU newstest2008–2011**
- **Epoch**
Sparse Features: Negative Results

- **Discriminative LM**
  - Jane called Sally

- **Phrase boundary features**
  - Jane || called Sally

- **Alignment constellation**
  - 1-0 0-1

- **Target word insertion**
  - Jane called the Sally