Online Relative Margin Maximization for Statistical Machine Translation

Vladimir Eidelman, Yuval Marton and Philip Resnik

ACL 2013
Motivation

• Machine Translation is useful😊
Motivation

• Machine Translation is useful 😊, but hard 😞
Motivation

• Machine Translation is *useful* 😊, but *hard* 😞
Motivation

• Machine Translation is useful 😊, but hard ☹️

[Diagram showing Parallel Corpus, Parameter Estimation, Dev set, decoder, french, and english connections]
Motivation

• Machine Translation is useful 😊, but hard 😞

• More features
  – External: parser, tagger, etc.
  – Internal: lexical pairs, rule identities, etc.
Motivation

• Machine Translation is useful 😊, but hard 😞

• More features
  – External: parser, tagger, etc.
  – Internal: lexical pairs, rule identities, etc.

• More data to tune on
  – Bitext tuning
Motivation

• Machine Translation is useful 😊, but hard 😞

• More features
  – External: parser, tagger, etc.
  – Internal: lexical pairs, rule identities, etc.

• More data to tune on
  – Bitext tuning
Motivation

• Machine Translation is useful 😊, but hard 😞

• More features
  – External: parser, tagger, etc.
  – Internal: lexical pairs, rule identities, etc.

• More data to tune on
  – Bitext tuning

high-dimensional feature space
online learning
Outline

• Presenting an online learning method for structured prediction with sparse feature that uses higher-order information to improve generalization ability
Optimization

• How to learn parameter vector $\mathbf{w}$
  – External evaluation metric
  – High-dimensional feature representation
  – Online
Optimization

• How to learn parameter vector $w$
  – External evaluation metric
  – High-dimensional feature representation
  – Online
Online Large-Margin Training

• MIRA *(Crammer et al., 2003, 2006)*
  – Passive-Aggressive update
    • Performing dual coordinate descent
    • Closed-form update similar to subgradient descent
Online Large-Margin Training

• MIRA (Crammer et al., 2003, 2006)
  – Passive-Aggressive update
    • Performing dual coordinate descent
    • Closed-form update similar to subgradient descent

• Adaptation to MT
Online Large-Margin Training

- Optimization problem:

  Training Instance: $(x_i, y_i)$
  cost: external error based on truth
Online Large-Margin Training

- Optimization problem:

  Training Instance: $\left(x_i, y_i\right)$
  
  cost: external error based on truth

  don’t change $\mathbf{w}$ too much

  $$
  \mathbf{w}_{t+1} = \arg\min_{\mathbf{w}} \frac{1}{2} \|\mathbf{w} - \mathbf{w}_t\|^2
  $$

  s.t. $\text{score}(x_i, y_i) - \text{score}(x_i, y') \geq \text{cost}(y_i, y')$

  $\forall y' \neq y_i$

  make margin as big as the cost
Online Large-Margin Training

- Optimization problem:

  Training Instance: \((x_i, y_i)\)  
  \[\text{cost: external error based on truth}\]

  \[w_{t+1} = \arg\min_w \frac{1}{2}||w - w_t||^2\]

  \[\text{s.t. } \Delta\text{score}(x_i, y_i, y') \geq \text{cost}(y_i, y')\]

  \[\forall y' \neq y_i\]

  make margin as big as the cost
Online Large-Margin Training

• Optimization problem:

Training Instance: \((x_i, y_i)\)  

\[
\begin{align*}
\text{cost: external error based on truth} \\
\text{don't change } w \text{ too much} \\
\end{align*}
\]

\[
w_{t+1} = \arg \min_w \frac{1}{2} \|w - w_t\|^2 + C \xi_i \\
\text{s.t. } \Delta \text{score}(x_i, y_i, y') \geq \text{cost}(y_i, y') - \xi_i \\
\forall y' \neq y_i
\]

make margin as big as the cost
Online Large-Margin Training

- Optimization problem:

Training Instance: \((x_i, y_i)\)

Cost: external error based on truth

\[
\min_{\mathbf{w}} \frac{1}{2} \| \mathbf{w} - \mathbf{w}_t \|^2 + C \xi_i
\]

\[
\text{s.t. } \Delta \text{score}(x_i, y_i, y') \geq \text{cost}(y_i, y') - \xi_i
\]

Don’t change \(\mathbf{w}\) too much

Make margin as big as the cost
Online Large-Margin Training

• Optimization problem:

Training Instance: \((x_i, y_i)\)  

\[ \ell_h = \max_{y' \in \mathcal{Y}(x_i)} \left( \text{cost}(y_i, y') - \Delta \text{score}(x_i, y_i, y') \right) \]

loss > 0 only if cost > margin
Online Large-Margin Training

- Optimization problem:

Training Instance: \((x_i, y_i)\)  

Cost: external error based on truth

\[
\ell_h = \max_{y' \in Y(x_i)} \left( \text{cost}(y_i, y') - \Delta \text{score}(x_i, y_i, y') \right)
\]

loss > 0 only if cost > margin

\[w \leftarrow w + \delta \left( \Delta \text{score}(x_i, y_i, y') \right)\]
PA Algorithm

1-cost

model score

*adapted from Chiang 2009
PA Algorithm

1-cost

model output

model score

*adapted from Chiang 2009
PA Algorithm

1-cost

model score

*adapted from Chiang 2009
PA Algorithm

![Diagram of PA Algorithm with points plotted on a 1-cost vs. model score axis.

*adapted from Chiang 2009*
$$\ell_h = \max_{y' \in \mathcal{Y}(x_i)} (\text{cost}(y_i, y') - \Delta \text{score}(x_i, y_i, y'))$$

*adapted from Chiang 2009*
PA Algorithm

\[ \ell_h = \max_{y' \in \mathcal{Y}(x_i)} (\text{cost}(y_i, y') - \Delta \text{score}(x_i, y_i, y')) \]
PA Algorithm

\[ \ell_h = \max_{y' \in \mathcal{Y}(x_i)} (\text{cost}(y_i, y') - \Delta \text{score}(x_i, y_i, y')) \]

*adapted from Chiang 2009*
PA Algorithm

\[ \ell_h = \max_{y' \in \mathcal{V}(x_i)} \left( \text{cost}(y_i, y') - \Delta \text{score}(x_i, y_i, y') \right) \]
\[ \ell_h = \max_{y' \in \mathcal{Y}(x_i)} \left( \text{cost}(y_i, y') - \Delta \text{score}(x_i, y_i, y') \right) \]
PA Algorithm

\[ \ell_h = \max_{y' \in \mathcal{Y}(x_i)} \left( \text{cost}(y_i, y') - \Delta \text{score}(x_i, y_i, y') \right) \]
PA Algorithm

\[ \ell_h = \max_{y' \in \mathcal{Y}(x_i)} \left( \text{cost}(y_i, y') - \Delta \text{score}(x_i, y_i, y') \right) \]
PA Algorithm

\[ \ell_h = \max_{y' \in Y(x_i)} \left( \text{cost}(y_i, y') - \Delta \text{score}(x_i, y_i, y') \right) \]

*adapted from Chiang 2009*
PA Algorithm

\[ \ell_h = \max_{y' \in Y(x_i)} (\text{cost}(y_i, y') - \Delta \text{score}(x_i, y_i, y')) \]

*adapted from Chiang 2009*
Relative Margin Motivation

- Can use lots of features (yay!)
- Want better generalization in high-dimensional spaces
Relative Margin Motivation

• Can use *lots* of features (yay!)
• Want better *generalization* in high-dimensional spaces
• Include *higher order information*
Relative Margin Motivation

• Can use lots of features (yay!)
• Want better generalization in high-dimensional spaces
• Include higher order information
  – Relative Margin Machine
Structured Relative Margin

\[ f_1(x, y) \]

\[ f_2(x, y) \]
Structured Relative Margin

\[ f_1(x, y) \]

\[ f_2(x, y) \]
Structured Relative Margin

\[ f_1(x, y) \]

\[ f_2(x, y) \]
Structured Relative Margin
Structured Relative Margin

\[ w^T f(x_i, y') \]
Structured Relative Margin

\[ \text{score}(x_i, y') \]
Structured Relative Margin
Structured Relative Margin
Structured Relative Margin

\[ f_1(x, y) \]

\[ f_2(x, y) \]
Structured Relative Margin
Structured Relative Margin
Structured Relative Margin
Structured Relative Margin

Large Margin

Relative Margin
Structured Relative Margin
Structured Relative Margin

Spread

Margin

Large Margin

Relative Margin

Margin

Spread
Structured Relative Margin

Spread

Margin

Large Margin

Relative Margin

Spread
Structured Relative Margin

\[ f_1(x, y) \]

\[ f_2(x, y) \]
Structured Relative Margin

\[ f_1(x, y) \]

\[ f_2(x, y) \]

unseen input

Large Margin

Relative Margin
Relative Margin Machine

- Measure **spread** of data after projection defined by \( w \)
  - projection given by \( \text{score}(x,y) \)

(Shivaswamy and Jebara; 2008, 2009)
Relative Margin Machine

• Measure **spread** of data after projection defined by \( w \)
  – projection given by \( \text{score}(x,y) \)

• Learn large-margin **relative** to the spread
  – relative margin = ratio of max-margin to spread

(Shivaswamy and Jebara; 2008, 2009)
Relative Margin Machine

• Measure **spread** of data after projection defined by $\mathbf{w}$
  – projection given by $\text{score}(x, y)$

• Learn large-margin **relative** to the spread
  – relative margin = ratio of max-margin to spread

• Create max-margin while **bounding** the spread

(Shivaswamy and Jebara; 2008, 2009)
RM for SMT

• What to optimize

\[ w_{t+1} = \arg \min_w \frac{1}{2} \| w - w_t \|^2 + C \xi_i \]

s.t. \( \Delta \text{score}(x_i, y^+, y^-) \geq \text{cost}_i(y^+, y^-) - \xi_i \)
RM for SMT

• What to optimize

\[ w_{t+1} = \arg \min_{w} \frac{1}{2} \| w - w_t \|^2 + C \xi_i \]

s.t. \( \Delta \text{score}(x_i, y^+, y^-) \geq \text{cost}_i(y^+, y^-) - \xi_i - B \leq \Delta \text{score}(x_i, y^+, y^w) \leq B \)

bound distance between correct and min score
RM for SMT

• What to optimize

\[ w_{t+1} = \arg \min_w \frac{1}{2} \| w - w_t \|^2 + C \xi_i \]

s.t. \( \Delta \text{score}(x_i, y^+, y^-) \geq \text{cost}_i(y^+, y^-) - \xi_i \)

\( -B \leq \Delta \text{score}(x_i, y^+, y^w) \leq B \)

bound distance between correct and min score
RM for SMT

• What to optimize

\[ w_{t+1} = \arg \min_w \frac{1}{2} \| w - w_t \|^2 + C\xi_i + D\tau_i \]

s.t. \( \Delta \text{score}(x_i, y^+, y^-) \geq \text{cost}_i(y^+, y^-) - \xi_i \)

\[-B - \tau_i \leq \Delta \text{score}(x_i, y^+, y^w) \leq B + \tau_i \]

bound distance between correct and min score
RM for SMT

• What to optimize

\[ w_{t+1} = \arg \min_w \frac{1}{2} \| w - w_t \|^2 + C \xi_i + D \tau_i \]

s.t. \[ \Delta \text{score}(x_i, y^+, y^-) \geq \text{cost}_i(y^+, y^-) - \xi_i \]

\[ -B - \tau_i \leq \Delta \text{score}(x_i, y^+, y^w) \leq B + \tau_i \]

Bounding Constraint
RM for SMT

• What to optimize

\[ w_{t+1} = \arg \min_w \frac{1}{2} \| w - w_t \|^2 + C\xi_i + D\tau_i \]

s.t. \[ \Delta \text{score}(x_i, y^+, y^-) \geq \text{cost}_i(y^+, y^-) - \xi_i \]

\[ -B - \tau_i \leq \Delta \text{score}(x_i, y^+, y^w) \leq B + \tau_i \]

Margin Constraint
RM Learning

- Shivaswamy developed batch optimization with off-the-shelf QP solver
  - Not a practical solution here
RM Learning

• Shivaswamy developed batch optimization with off-the-shelf QP solver
  – Not a practical solution here

• Introduce online gradient-based approach
  – Developed online update
    • Cutting Plane and PA version
    • Iterate between satisfying margin and bounding constraints
RM Algorithm

1-cost

model score
RM Algorithm

1-cost

model score

1-cost

margin

cost

$y^+$

$y^-$
RM Algorithm

1-cost

model score

B

\( y^+ \)

\( y^- \)

margin

18
RM Algorithm

\[ \Delta \text{score}(x, y^+, y^w) > B \]
Evaluation

• **Chinese-English** *(1.6M)*
  – NIST MT06 tune, MT03 and MT05 test

• **Arabic-English** *(1M)*
  – NIST MT06 tune, MT05, and MT08 test

• 4-gram LM *(600M words)*
Experimental Setup

- HPB decoder: cdec
- Feature Sets:
  - Baseline
  - Sparse
- Baseline Optimizers
  - MERT
  - MIRA
  - RAMPION
  - PRO
Experimental Setup

• Baseline: 11 features
  – 4 Penalties:
    • Pass Through
    • Glue
    • Target Word
    • Source Word
  – Language Model
  – 5 Phrase table features
Experimental Setup

• Sparse:
  – Structural Distortion
  – Lexicalized Features
    • Rule Identity
    • Insertion / Deletion
    • Word pairs
  – Rule Shape
<table>
<thead>
<tr>
<th>Optimizer</th>
<th>Dense feature set</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tune</td>
<td></td>
</tr>
<tr>
<td></td>
<td>BLEU</td>
</tr>
<tr>
<td>MERT</td>
<td>35.4</td>
</tr>
<tr>
<td>MIRA</td>
<td>35.5</td>
</tr>
<tr>
<td>PRO</td>
<td>34.1</td>
</tr>
<tr>
<td>RAMPION</td>
<td>35.1</td>
</tr>
</tbody>
</table>
## Chinese-English Results

<table>
<thead>
<tr>
<th>Optimizer</th>
<th>Dense feature set</th>
<th>Sparse feature set</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Tune</td>
<td>MT03</td>
</tr>
<tr>
<td></td>
<td>BLEU</td>
<td>BLEU</td>
</tr>
<tr>
<td>MERT</td>
<td>35.4</td>
<td>35.8</td>
</tr>
<tr>
<td>MIRA</td>
<td>35.5</td>
<td>35.8</td>
</tr>
<tr>
<td>PRO</td>
<td>34.1</td>
<td>36</td>
</tr>
<tr>
<td>RAMPION</td>
<td>35.1</td>
<td>36.5</td>
</tr>
</tbody>
</table>
## Chinese-English Results

<table>
<thead>
<tr>
<th>Optimizer</th>
<th>Dense feature set</th>
<th>Sparse feature set</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Tune MT03 MT05</td>
<td>Tune MT03 MT05</td>
</tr>
<tr>
<td></td>
<td>BLEU   BLEU   TER</td>
<td>BLEU   TER</td>
</tr>
<tr>
<td>MERT</td>
<td>35.4   35.8  60.8</td>
<td>32.4  63.9</td>
</tr>
<tr>
<td>MIRA</td>
<td>35.5   35.8  61.1</td>
<td>32.1  64.6</td>
</tr>
<tr>
<td>PRO</td>
<td>34.1   36    60.2</td>
<td>31.7  63.4</td>
</tr>
<tr>
<td>RAMPION</td>
<td>35.1   36.5  58.6</td>
<td>33    61.3</td>
</tr>
<tr>
<td>RM</td>
<td>31.3   36.5  56.4</td>
<td>33.6  59.3</td>
</tr>
</tbody>
</table>
Chinese-English Results

<table>
<thead>
<tr>
<th>Optimizer</th>
<th>Dense feature set</th>
<th>Sparse feature set</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Tune</td>
<td>MT03</td>
</tr>
<tr>
<td></td>
<td>BLEU</td>
<td>BLEU</td>
</tr>
<tr>
<td>MERT</td>
<td>35.4</td>
<td>35.8</td>
</tr>
<tr>
<td>MIRA</td>
<td>35.5</td>
<td>35.8</td>
</tr>
<tr>
<td>PRO</td>
<td>34.1</td>
<td>36</td>
</tr>
<tr>
<td>RAMPION</td>
<td>35.1</td>
<td>36.5</td>
</tr>
<tr>
<td>RM</td>
<td>31.3</td>
<td>36.5</td>
</tr>
</tbody>
</table>

**0.7-1.5** BLEU gain over MIRA
## Chinese-English Results

<table>
<thead>
<tr>
<th>Optimizer</th>
<th>Dense feature set</th>
<th>MT03</th>
<th>MT05</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Tune</td>
<td>MT03</td>
<td>MT05</td>
</tr>
<tr>
<td>MERT</td>
<td>35.4</td>
<td>35.8</td>
<td>60.8</td>
</tr>
<tr>
<td>MIRA</td>
<td>35.5</td>
<td>35.8</td>
<td>61.1</td>
</tr>
<tr>
<td>PRO</td>
<td>34.1</td>
<td>36</td>
<td>60.2</td>
</tr>
<tr>
<td>RAMPION</td>
<td>35.1</td>
<td>36.5</td>
<td>58.6</td>
</tr>
<tr>
<td>RM</td>
<td>31.3</td>
<td>36.5</td>
<td>56.4</td>
</tr>
</tbody>
</table>

- **0.7-1.5** BLEU gain over MIRA
- **4.7-5.3** TER gain over MIRA
## Chinese-English Results

<table>
<thead>
<tr>
<th>Optimizer</th>
<th>Dense feature set</th>
<th>Sparse feature set</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Tune</td>
<td>MT03</td>
</tr>
<tr>
<td></td>
<td>BLEU</td>
<td>BLEU</td>
</tr>
<tr>
<td>MERT</td>
<td>35.4</td>
<td>35.8</td>
</tr>
<tr>
<td>MIRA</td>
<td>35.5</td>
<td>35.8</td>
</tr>
<tr>
<td>PRO</td>
<td>34.1</td>
<td>36</td>
</tr>
<tr>
<td>RAMPION</td>
<td>35.1</td>
<td>36.5</td>
</tr>
<tr>
<td>RM</td>
<td>31.3</td>
<td>36.5</td>
</tr>
</tbody>
</table>

0.7-1.5 BLEU gain over MIRA  
4.7-5.3 TER gain over MIRA
# Chinese-English Results

<table>
<thead>
<tr>
<th>Optimizer</th>
<th>Dense feature set</th>
<th>Sparse feature set</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Tuned MT03</td>
<td>MT05</td>
</tr>
<tr>
<td></td>
<td>BLEU</td>
<td>BLEU</td>
</tr>
<tr>
<td>MERT</td>
<td>35.4</td>
<td>35.8</td>
</tr>
<tr>
<td>MIRA</td>
<td>35.5</td>
<td>35.8</td>
</tr>
<tr>
<td>PRO</td>
<td>34.1</td>
<td>36</td>
</tr>
<tr>
<td>RAMPION</td>
<td>35.1</td>
<td>36.5</td>
</tr>
<tr>
<td>RM</td>
<td>31.3</td>
<td>36.5</td>
</tr>
</tbody>
</table>

**BLEU gain over MIRA**

- 0.7-1.5

**TER gain over MIRA**

- 4.7-5.3
- 6-6.6
## Arabic-English Results

<table>
<thead>
<tr>
<th>Optimizer</th>
<th>Dense feature set</th>
<th>Sparse feature set</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Tune</td>
<td>MT03</td>
</tr>
<tr>
<td></td>
<td>BLEU</td>
<td>BLEU</td>
</tr>
<tr>
<td>MERT</td>
<td>43.8</td>
<td>53.3</td>
</tr>
<tr>
<td>MIRA</td>
<td>43.0</td>
<td>52.8</td>
</tr>
<tr>
<td>PRO</td>
<td>41.5</td>
<td>51.3</td>
</tr>
<tr>
<td>RAMPION</td>
<td>42.4</td>
<td>52.0</td>
</tr>
<tr>
<td>RM</td>
<td>38.5</td>
<td>53.3</td>
</tr>
</tbody>
</table>
## Arabic-English Results

<table>
<thead>
<tr>
<th>Optimizer</th>
<th>Dense feature set</th>
<th>Sparse feature set</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tune</td>
<td>BLEU</td>
<td>BLEU</td>
</tr>
<tr>
<td>MERT</td>
<td>43.8</td>
<td>53.3</td>
</tr>
<tr>
<td>MIRA</td>
<td>43.0</td>
<td>52.8</td>
</tr>
<tr>
<td>PRO</td>
<td>41.5</td>
<td>51.3</td>
</tr>
<tr>
<td>RAMPION</td>
<td>42.4</td>
<td>52.0</td>
</tr>
<tr>
<td>RM</td>
<td>38.5</td>
<td>53.3</td>
</tr>
</tbody>
</table>

-0.7-0.5 BLEU gain over MIRA
### Arabic-English Results

<table>
<thead>
<tr>
<th>Optimizer</th>
<th>Dense feature set</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Tuned MT03</td>
<td>MT05</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>BLEU 53.3</td>
<td>BLEU 40.2</td>
<td>TER 41.0</td>
<td>TER 50.7</td>
</tr>
<tr>
<td>MERT</td>
<td>43.8</td>
<td>41.0</td>
<td>50.7</td>
<td></td>
</tr>
<tr>
<td>MIRA</td>
<td>43.0</td>
<td>41.3</td>
<td>50.6</td>
<td></td>
</tr>
<tr>
<td>PRO</td>
<td>41.5</td>
<td>39.4</td>
<td>51.5</td>
<td></td>
</tr>
<tr>
<td>RAMPION</td>
<td>42.4</td>
<td>40.0</td>
<td>50.8</td>
<td></td>
</tr>
<tr>
<td>RM</td>
<td>38.5</td>
<td>40.6</td>
<td>49.7</td>
<td></td>
</tr>
</tbody>
</table>

- **-0.7-0.5** BLEU gain over MIRA
- **0.9-1** TER gain over MIRA
# Arabic-English Results

<table>
<thead>
<tr>
<th>Optimizer</th>
<th>Dense feature set</th>
<th>Sparse feature set</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Tuned MT03</td>
<td>Tuned MT05</td>
</tr>
<tr>
<td></td>
<td>Tuned MT05</td>
<td>Tuned MT05</td>
</tr>
<tr>
<td></td>
<td>MT03</td>
<td>MT05</td>
</tr>
<tr>
<td></td>
<td>MT05</td>
<td>MT05</td>
</tr>
<tr>
<td></td>
<td>BLEU</td>
<td>BLEU</td>
</tr>
<tr>
<td>MERT</td>
<td>43.8</td>
<td>53.3</td>
</tr>
<tr>
<td>MIRA</td>
<td>43.0</td>
<td>52.8</td>
</tr>
<tr>
<td>PRO</td>
<td>41.5</td>
<td>51.3</td>
</tr>
<tr>
<td>RAMPION</td>
<td>42.4</td>
<td>52.0</td>
</tr>
<tr>
<td>RM</td>
<td>38.5</td>
<td>53.3</td>
</tr>
</tbody>
</table>

-0.7 to -0.5 BLEU gain over MIRA
0.9 to 1 TER gain over MIRA
## Arabic-English Results

<table>
<thead>
<tr>
<th>Optimizer</th>
<th>Dense feature set</th>
<th>Sparse feature set</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Tune</td>
<td>MT03</td>
</tr>
<tr>
<td></td>
<td>BLEU</td>
<td>BLEU</td>
</tr>
<tr>
<td>MERT</td>
<td>43.8</td>
<td>53.3</td>
</tr>
<tr>
<td>MIRA</td>
<td>43.0</td>
<td>52.8</td>
</tr>
<tr>
<td>PRO</td>
<td>41.5</td>
<td>51.3</td>
</tr>
<tr>
<td>RAMPION</td>
<td>42.4</td>
<td>52.0</td>
</tr>
<tr>
<td>RM</td>
<td>38.5</td>
<td>53.3</td>
</tr>
</tbody>
</table>

**-0.7-0.5** BLEU gain over MIRA  
**0.9-1** TER gain over MIRA  
**0-1.9** BLEU gain over MIRA  
**1.8-2.6** TER gain over MIRA
# Spread Analysis

## Chinese-English

### Small Feature

<table>
<thead>
<tr>
<th></th>
<th>MIRA</th>
<th>RM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average</td>
<td>5.9 (20.5)</td>
<td>0.7 (2.9)</td>
</tr>
<tr>
<td>Avg (std)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Spread Analysis

### Chinese-English

<table>
<thead>
<tr>
<th>Feature Type</th>
<th>MIRA</th>
<th>RM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Small Feature</td>
<td>5.9 (20.5)</td>
<td>0.7 (2.9)</td>
</tr>
<tr>
<td>Large Feature</td>
<td>14 (31.1)</td>
<td>0.9 (2.4)</td>
</tr>
</tbody>
</table>
## Spread Analysis

<table>
<thead>
<tr>
<th></th>
<th>Chinese-English</th>
<th></th>
<th>Arabic-English</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Small Feature</td>
<td>Large Feature</td>
<td>Small Feature</td>
<td>Large Feature</td>
</tr>
<tr>
<td><strong>MIRA</strong></td>
<td>5.9 (20.5)</td>
<td>14 (31.1)</td>
<td>9.4 (26.8)</td>
<td>11.4 (22.1)</td>
</tr>
<tr>
<td><strong>RM</strong></td>
<td>0.7 (2.9)</td>
<td>0.9 (2.4)</td>
<td>0.7 (2.4)</td>
<td>0.8 (1.4)</td>
</tr>
</tbody>
</table>
Take Away
Take Away

• Extended relative margin methods to SMT
  – use of second-order information for generalization
  – bound spread of the data
Take Away

• Extended relative margin methods to SMT
  – use of second-order information for generalization
  – bound spread of the data
• Introduced online gradient-based update
  – PA and Cutting Plane version
  – Easily incorporate into any gradient based learning
Take Away

• Extended relative margin methods to SMT
  – use of second-order information for generalization
  – bound spread of the data

• Introduced online gradient-based update
  – PA and Cutting Plane version
  – Easily incorporate into any gradient based learning

• Significantly outperforms other online and batch methods
Take Away (for you)

• Will be available as part of cdec
• Can be applied to other structured prediction problems
Thank You!