Improved Online Learning and Modeling for Feature-Rich Discriminative Machine Translation

Vladimir A. Eidelman
Dissertation Defense

November 5, 2013
Motivation

• Machine Translation

```
french
  ↑
  ↓
  ↓
  ↓
decoder
  ↑
  ↓
  ↓
  ↓
english
```
Motivation

• Machine Translation

\[ x \in \mathcal{X} \]

\[ \text{arg max}_{y \in \mathcal{Y}(x)} \text{score}(x, y) \]

\[ y \in \mathcal{Y}(x) \]
Motivation

• Machine Translation

\[ x \in \mathcal{X} \]

\[
\arg \max_{y \in \mathcal{Y}(x)} w^T f(x, y)
\]

\[ y \in \mathcal{Y}(x) \]
Motivation

- Machine Translation
Statistical Machine Translation

Parallel Corpus

Translation model

Language model

Monolingual

Discriminative Training

tuning set

decoder

french

english
Statistical Machine Translation

- Parallel Corpus
  - Translation model
    - Language model
    - Monolingual
  - Discriminative Training
    - Tuning set
    - French
    - English
    - Decoder
Statistical Machine Translation

Translation model

Language model

Monolingual

Discriminative Training

tuning set

french

decoder

english
Hierarchical Phrase-Based Translation
Hierarchical Phrase-Based Translation

**src:** morgen fliege ich nach Kanada zur Konferenz

**trg:** tomorrow I will fly to the conference in Canada
Hierarchical Phrase-Based Translation

src: morgen fliege ich nach Kanada zur Konferenz

trg: tomorrow I will fly to the conference in Canada

morgen tomorrow
Hierarchical Phrase-Based Translation

**src**: morgen fliege ich nach Kanada zur Konferenz

**trg**: tomorrow I will fly to the conference in Canada
Synchronous Parsing

morgen fliege ich nach Kanada zur Konferenz
tomorrow I will fly to the conference in Canada

morgen fliege ich nach Kanada zur Konferenz

tomorrow I will fly to the conference in Canada
tomorrow I will fly to the conference in Canada

morgen    fliege ich   nach  Kanada  zur Konferenz

morgen    fliege ich   nach  Kanada  zur Konferenz
tomorrow I will fly to the conference in Canada

tomorrow I will fly in Canada to the conference

morgen fliege ich nach Kanada zur Konferenz

tomorrow I will fly

S₁ X₂ S₁ X₂

in Canada to the conference

S₁ X₂ S₁ X₂

in Canada to the conference

nach X₁ in X₁

tomorrow I will fly

X₁ ich I X₁

I will fly

morgen tomorrow fliege will fly

morgen fliege ich nach Kanada zur Konferenz
tomorrow I will fly to the conference in Canada
tomorrow I will fly to the conference in Canada

\[ f_1 = .68 \quad f_4 = 1 \]

\[ f_1 = 1.88 \quad f_2 = 1 \quad f_3 = -.43 \]

\[ f_1 = -2.19 \quad f_2 = 1 \quad f_3 = -.86 \]

\[ f_1 = -.17 \quad f_3 = -.43 \]

\[ f_1 = 1.28 \quad f_4 = 1 \]

\[ f_1 = 1.6 \quad f_3 = -.5 \]

\[ f_1 = 1.4 \quad f_3 = -.9 \]

\[ f_1 = 1.2 \quad f_3 = -.5 \]

\[ morgen \quad tomorrow \]

\[ fliege \quad will fly \]

\[ Kanada \quad Canada \]

\[ Konferenz \quad the Conference \]

\[ X_1 \quad ich \]

\[ I \quad X_1 \]

\[ nach \quad X_1 \quad zur \quad X_2 \]

\[ to \quad X_2 \quad in \quad X_1 \]
 tomorrow I will fly to the conference in Canada

\[ w^T f_1 = 0.68 \quad f_4 = 1 \]

\[ w^T f_1 = 1.28 \quad f_4 = 1 \]

\[ w^T f_1 = -1.7 \quad f_3 = -0.43 \]

\[ w^T f_1 = 1.28 \quad f_4 = 1 \]

\[ w^T f_1 = 1.6 \quad f_3 = -0.3 \]

\[ w^T f_1 = -1.6 \quad f_3 = -0.3 \]

\[ w^T f_1 = 1.4 \quad f_3 = -0.9 \]

\[ w^T f_1 = -1.2 \quad f_3 = -0.5 \]
tomorrow I will fly to the conference in Canada

score=1.5
Derivation Scoring

tomorrow I will fly to the conference in Canada

score = 1.5

\[
\text{deriv} = \sum \text{score} = -7.6
\]
tomorrow I will fly to the conference in Canada
Optimization Problem

• Have way of getting translations, but how do we know if they’re good?

- tomorrow I will fly to Canada to the conference
- tomorrow I will fly to Canada to the conference
- tomorrow I fly to the conference in Canada
- tomorrow I fly to Canada to the conference
- tomorrow I fly to the conference in Canada
- tomorrow I will fly in Canada to the conference

Values:
-11.7
-10.8
-7.6
-11.9
-8.7
-11.7
Optimization Problem

• Have way of getting translations, but how do we know if they’re good?

1. Define good features
2. Learn good feature weights $w$ from data
Machine Translation

Parallel Corpus

Translation model

Language model

Monolingual

Discriminative Training

features

tuning set

features

french

decoder

english
Machine Translation

Parallel Corpus

Translation model

Language model

Monolingual

Discriminative Training

tuning set

features

decoder

english

french

features

w
Machine Translation

Parallel Corpus

Translation model

Language model

Monolingual

Discriminative Training

features

features

monolingual decoder

tuning set

Machine Learning

english

french

features

Machine Learning
Motivation

• Machine Translation is hard 😞
Motivation

• Machine Translation is hard 😞

• More features
  – External: Syntactic (parser, tagger), Semantic (SRL)
  – Internal: lexical pairs, rule identities, etc.
Motivation

• Machine Translation is hard 😞

• More features
  – External: Syntactic (parser, tagger), Semantic (SRL)
  – Internal: lexical pairs, rule identities, etc.

• More data to tune on
  – Bitext tuning
Motivation

• Machine Translation is hard 😞

• More features
  – External: Syntactic (parser, tagger), Semantic (SRL)
  – Internal: lexical pairs, rule identities, etc.

• More data to tune on
  – Bitext tuning
Motivation

• Machine Translation is **hard 😞**

• More **features**
  – External: Syntactic (parser, tagger), Semantic (SRL)
  – Internal: lexical pairs, rule identities, etc.

• More **data** to tune on
  – Bitext tuning

  high-dimensional feature space

  scalable methods
Goal

• Want efficient learning in high-dimensions with good generalization for structured latent output
Goal

• Want efficient learning in high-dimensions with good generalization for structured latent output

Contribution

• Present generalized form of cost-augmented objectives as family of loss functions
• Conduct comprehensive empirical analysis of optimization performance
• Develop tool for large-scale large-margin training and show its practicability
Goal

- Want efficient learning in high-dimensions with good generalization for structured latent output

Contribution

- Introduce loss for structured relative margin with cost-augmented inference
- Derive an online gradient based solver
- Introduce method for dynamic domain adaptation
  - Develop unsupervised domain adaptation features
Goal

• Want

  efficient learning in high-dimensions with good generalization for structured latent output

Contribution

• Define unified representation of structured latent objectives
• Introduce novel loss for latent large-margin learning
• Develop optimization procedure for maximum probability translation learning and inference
Outline

• Efficient learning in high-dimensions
  – Online Large-Margin Learning
    (Eidelman, 2012@WMT; Eidelman et al., 2013@WMT; Eidelman et al., 2013@ACL)

• Good generalization
  – Online Relative Margin Maximization
  – Adaptation with Topic Models

• Structured Latent Output
  – Latent Large-Margin Learning

• Contributions
Optimization

• How to learn parameter vector $\mathbf{w}$
Optimization

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

• How to learn parameter vector $w$
  – External evaluation metric
  – High-dimensional feature representation
  – Fast convergence

• MERT unable to scale
  – can handle < 30 features
Optimization

• How to learn parameter vector $w$
  – External evaluation metric
  – High-dimensional feature representation
  – Fast convergence

• MERT unable to scale
  – can handle < 30 features
Optimization

• How to learn parameter vector \( \mathbf{w} \)
  – External evaluation metric
  – High-dimensional feature representation
  – Fast convergence

• MERT unable to scale
  – can handle < 30 features
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: \((x_i, y_i)\)  

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

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

\( \forall y' \neq y_i \)

\[ \text{make margin as big as the cost} \]
Online Large-Margin Training

- Optimization problem:

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

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

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

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

don’t change \(w\) too much

\[
\begin{align*}
\mathbf{w}_{t+1} &= \arg \min_{\mathbf{w}} \frac{1}{2} \| \mathbf{w} - \mathbf{w}_t \|^2 + C \xi_i \\
\text{s.t. } \Delta \text{score}(x_i, y_i, y') &\geq \text{cost}(y_i, y') - \xi_i \\
\forall y' &\neq y_i
\end{align*}
\]

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

• Optimization problem:

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

\[
\text{cost: external error based on truth}
\]

\[
\text{don't change } w \text{ too much}
\]

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

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

\[
\forall y' \neq y_i
\]

\[
\text{make margin as big as the cost}
\]
Online Large-Margin Training

• Optimization problem:

\[ \ell_h = -\mathbf{w}^\top \mathbf{f}(x_i, y_i) + \max_{y' \in \mathcal{Y}(x_i)} \left( \mathbf{w}^\top \mathbf{f}(x_i, y') + \text{cost}(y_i, y') \right) \]

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

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

• Optimization problem:

\[ \ell_h = -w^\top f(x_i, y_i) + \max_{y' \in \mathcal{Y}(x_i)} \left( w^\top f(x_i, y') + \text{cost}(y_i, y') \right) \]

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

loss > 0 only if cost > margin

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

*adapted from Chiang 2009
PA Algorithm

\[ \text{model output} \rightarrow \]

*adapted from Chiang 2009*
PA Algorithm

1-cost

model score

*adapted from Chiang 2009
PA Algorithm

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

*adapted from Chiang 2009*
PA Algorithm

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

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

*adapted from Chiang 2009*
PA Algorithm

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

*adapted from Chiang 2009*
PA Algorithm

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

*adapted from Chiang 2009*
PA Algorithm

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

*adapted from Chiang 2009*
PA Algorithm

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

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

• **Unlike** other structured prediction tasks
• Disconnect between machine learning and MT
  – Supervised learning $\rightarrow (x_i, y_i)$
Challenges in SMT

• Unlike other structured prediction tasks
• Disconnect between machine learning and MT
  – Supervised learning $\rightarrow (x_i, y_i)$

1. Many valid translations
Challenges in SMT

• Unlike other structured prediction tasks
• Disconnect between machine learning and MT
  – Supervised learning $\rightarrow (x_i, y_i)$

1. Many valid translations
2. Many paths to the same output translation
  – Derivation modeled as a latent variable
Challenges in SMT

• Unlike other structured prediction tasks
• Disconnect between machine learning and MT
  – Supervised learning $\rightarrow (x_i, y_i)$

1. Many valid translations
2. Many paths to the same output translation
   – Derivation modeled as a latent variable
3. Model may not be able to produce correct (reference) translation (i.e. $y_i \notin \mathcal{Y}(x_i)$)
tomorrow I will fly to the conference in Canada
tomorrow I will fly to the conference in Canada
morgen fliege ich nach Kanada zur Konferenz

tomorrow I will fly to the conference in Canada
(un)reachable references

src: morgen fliege ich nach Kanada zur Konferenz

ref: tomorrow I will fly to the conference in Canada

oracle: tomorrow I will fly in Canada to the conference
Application to SMT

• Why is that a problem?
Application to SMT

• Why is that a problem?

\[ \ell_h = -\mathbf{w}^\top \mathbf{f}(x_i, y_i) + \max_{y' \in \mathcal{Y}(x_i)} \left( \mathbf{w}^\top \mathbf{f}(x_i, y') + \text{cost}(y_i, y') \right) \]

need to be able to compute model score on \( Y_i \)

\( y_i: \) tomorrow I will fly to the conference in Canada
Application to SMT

Why is that a problem?

\[ \ell_h = -\mathbf{w}^\top \mathbf{f}(x_i, \text{?}) + \max_{y' \in \mathcal{Y}(x_i)} \left( \mathbf{w}^\top \mathbf{f}(x_i, y') + \text{cost}(y_i, y') \right) \]

surrogate reference (hope)

\[ y^+: \text{tomorrow I will fly in Canada to the conference} \]

\[ y_i: \text{tomorrow I will fly to the conference in Canada} \]
Application to SMT

- Why is that a problem?

\[ \ell_h = -\mathbf{w}^\top \mathbf{f}(x_i, ?) + \max_{y' \in \mathcal{Y}(x_i)} \left( \mathbf{w}^\top \mathbf{f}(x_i, ?) + \text{cost}(y_i, ?) \right) \]

surrogate reference (hope) \[ y^+ \]

worst violator (fear) \[ y^- \]

good model score, but actually bad

(Chiang 2008, 2012)
Loss Functions
Loss Functions

- What to optimize? Family of cost-augmented losses

\[
\ell_r = - \max_{y^+ \in \mathcal{Y}(x_i)} \left( \gamma^+ w^\top f(x_i, y^+) - \beta^+ \text{cost}(y_i, y^+) \right) \\
+ \max_{y^- \in \mathcal{Y}(x_i)} \left( \gamma^- w^\top f(x_i, y^-) + \beta^- \text{cost}(y_i, y^-) \right)
\]
Loss Functions

- What to optimize? Family of cost-augmented losses
  - Different choices for hope and
  - Given by $\gamma = \{0,1\}$ and $\beta = \{0,1\}$
- Unified characterization:

$$\ell_r = -\max_{y^+ \in \mathcal{Y}(x_i)} \left( \gamma^+ w^\top f(x_i, y^+) - \beta^+ \text{cost}(y_i, y^+) \right)$$

$$+ \max_{y^- \in \mathcal{Y}(x_i)} \left( \gamma^- w^\top f(x_i, y^-) + \beta^- \text{cost}(y_i, y^-) \right)$$
Loss Functions

• What to optimize? Family of cost-augmented losses
  – Different choices for hope and
  – Given by $\gamma = \{0, 1\}$ and $\beta = \{0, 1\}$

• Unified characterization:

$$\ell_r = -\max_{y^+ \in \mathcal{Y}(x_i)} \left( \gamma^+ w^\top f(x_i, y^+) - \beta^+ \text{cost}(y_i, y^+) \right)$$

$$+ \max_{y^- \in \mathcal{Y}(x_i)} \left( \gamma^- w^\top f(x_i, y^-) + \beta^- \text{cost}(y_i, y^-) \right)$$
Loss Functions

• What to optimize? Family of cost-augmented losses
  – Different choices for hope and fear
  – Given by $\gamma = \{0, 1\}$ and $\beta = \{0, 1\}$
• Unified characterization:

$$
\ell_r = - \max_{y^+ \in \mathcal{Y}(x_i)} \left( \gamma^+ w^\top f(x_i, y^+) - \beta^+ \text{cost}(y_i, y^+) \right) + \max_{y^- \in \mathcal{Y}(x_i)} \left( \gamma^- w^\top f(x_i, y^-) + \beta^- \text{cost}(y_i, y^-) \right)
$$
Loss Functions

\[
\ell_r = - \max_{y^+ \in \mathcal{Y}(x_i)} \left( \gamma^+ w^\top f(x_i, y^+) - \beta^+ \text{cost}(y_i, y^+) \right) \\
+ \max_{y^- \in \mathcal{Y}(x_i)} \left( \gamma^- w^\top f(x_i, y^-) + \beta^- \text{cost}(y_i, y^-) \right)
\]

• Every combination of choices leads to different update
  – Many possible losses
    • Solvers: cutting-plane vs. passive-aggressive
    • Parallelization methods
Take Away
Take Away

• Presented unified framework for different cost-augmented loss functions
Take Away

• Presented unified framework for different cost-augmented loss functions
• Extensively empirically analyzed optimization performance of hope and fear choices and corresponding updates
  – Leads to simple margin-based PA training algorithm for SMT
  – Choice of candidates is important role for stability and effectiveness
  – Best combination of stability and generalization from
    \[
    y^+ \leftarrow \arg \max_{y \in y(x_i)} w^T f(x_i, y) - \text{cost}(y_i, y)
    \]
    \[
    y^- \leftarrow \arg \max_{y \in y(x_i)} w^T f(x_i, y) + \text{cost}(y_i, y)
    \]
  – Good performance in low and high dimensions for French, Czech, German, and Russian
Take Away

- Presented unified framework for different cost-augmented loss functions
- Extensively empirically analyzed optimization performance of hope and fear choices and corresponding updates
  - Leads to simple margin-based PA training algorithm for SMT
  - Choice of candidates is important role for stability and effectiveness
  - Best combination of stability and generalization from
    $y^+ \leftarrow \arg\max_{y \in \gamma(x_i)} w^T f(x_i, y) - \text{cost}(y_i, y)$
    $y^- \leftarrow \arg\max_{y \in \gamma(x_i)} w^T f(x_i, y) + \text{cost}(y_i, y)$
  - Good performance in low and high dimensions for French, Czech, German, and Russian
- Developed capability for practical large scale training
  - Scalable large-margin learning on MapReduce
  - Open source
Outline

• Efficient learning in high-dimensions
  – Online Large-Margin Learning

• Good generalization
  – Online Relative Margin Maximization
    (Eidelman, 2013@ACL)
  – Adaptation with Topic Models

• Structured Latent Output
  – Latent Large-Margin Learning

• Contributions
Relative Margin Motivation

- Can use lots of features (yay!)
- How can we improve generalization in high-dimensional spaces?
Relative Margin Motivation

• Can use lots of features (yay!)
• How can we improve generalization in high-dimensional spaces?
• Include higher order information
Relative Margin Motivation

• Can use lots of features (yay!)
• How can we improve 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

\[ w^\top f(x_i, y') \]
Structured Relative Margin

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

Spread

Margin

28
Structured Relative Margin

\[ f_1(x, y) \]

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

Large Margin

Relative Margin

Spread

Margin

Spread

Margin
Structured Relative Margin

Spread

Margin

Large Margin

Relative Margin

Spread

Margin
Structured Relative Margin

\[ f_1(x, y) \]

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

$f_1(x, y)$

Relative Margin

$f_2(x, y)$

Large Margin

test input
Relative Margin Machine

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

• Measure spread of data after projection defined by $w$
  – projection given by 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
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

• We introduce online gradient-based approach
  – Developed online update
    • Cutting Plane and PA version
    • Iterate between satisfying margin and bounding constraints
RM for SMT

• What to optimize

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

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

make margin as big as the cost
RM for SMT

• What to optimize

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

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

- make margin as big as the cost

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

- bound distance between correct and min score
RM for SMT

• What to optimize

$$w_{t+1} = \underset{w}{\text{argmin}} \frac{1}{2} \| w - w_t \|^2 + C \xi_i + D \tau_i$$

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

make margin as big as the cost

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

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

make margin as big as the cost

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

Bounding Constraint
RM for SMT

- What to optimize

\[
\mathbf{w}_{t+1} = \arg\min_{\mathbf{w}} \frac{1}{2} \| \mathbf{w} - \mathbf{w}_t \|^2 + C \xi_i + D \tau_i
\]

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

Margin Constraint

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

1-cost vs. model score graph with data points and a shaded region.
RM Algorithm

1-cost vs model score

1-cost to cost

margin

$y^+$ and $y^-$
RM Algorithm

1-cost

model score

1 - cost

cost

margin

B

$y^+$

$y^-$
RM Algorithm

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

1-cost

model score

B
Evaluation

• **Chinese-English** (1.6M sentence pairs)
  – NIST MT06 tune, MT03 and MT05 test
• **Arabic-English** (1M sentence pairs)
  – NIST MT06 tune, MT05, and MT08 test
• **4-gram LM** (600M words)
Experimental Setup

• Feature Sets:
  – Baseline
  – Sparse

• Baseline Optimizers
  – MERT
  – MIRA
  – RAMPION
  – PRO
Experimental Setup

• Baseline: 11 features (Koehn, 2010)
  – 4 Penalties:
    • Pass Through
    • Glue
    • Target Word
    • Source Word
  – Language Model
  – 5 Phrase table features
Experimental Setup

• Sparse: ~100k features
  – Rule identity
  – Lexicalized
    • Insertion / Deletion
    • Target bigram
    • Contextual word pairs
  – Structural distortion
  – Rule shape
## Chinese-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></td>
<td></td>
</tr>
<tr>
<td><strong>BLEU</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MERT</td>
<td>35.4</td>
<td>63.9</td>
</tr>
<tr>
<td>MIRA</td>
<td>35.5</td>
<td>64.6</td>
</tr>
<tr>
<td>PRO</td>
<td>34.1</td>
<td>64.1</td>
</tr>
<tr>
<td>RAMPION</td>
<td>35.1</td>
<td>61.3</td>
</tr>
<tr>
<td>RM</td>
<td>31.3</td>
<td>32.1</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>
<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>Tune MT03 BLEU</td>
<td>Tune MT03 BLEU</td>
</tr>
<tr>
<td>MERT</td>
<td></td>
<td>MT03 TER BLEU</td>
</tr>
<tr>
<td>MIRA</td>
<td>36.6 35.9 60.6</td>
<td>35.7 34.8 56.1</td>
</tr>
<tr>
<td>PRO</td>
<td>35.7 34.8 56.1</td>
<td>31.4 59.1</td>
</tr>
<tr>
<td>RAMPION</td>
<td>36.7 36.9 57.7</td>
<td>33.3 60.6</td>
</tr>
<tr>
<td>RM</td>
<td>33.2 37.5 54.6</td>
<td>34 57.5</td>
</tr>
</tbody>
</table>

0.6-1.9 BLEU gain over MIRA
6-6.6 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>Tune</td>
<td>MT03</td>
<td>MT05</td>
</tr>
<tr>
<td>BLEU</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
### Arabic-English Results

<table>
<thead>
<tr>
<th>Optimizer</th>
<th>Sparse feature set</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tune</td>
<td>MT03</td>
</tr>
<tr>
<td>BLEU</td>
<td>BLEU</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>MERT</td>
<td></td>
</tr>
<tr>
<td>MIRA</td>
<td></td>
</tr>
<tr>
<td>PRO</td>
<td></td>
</tr>
<tr>
<td>RAMPION</td>
<td></td>
</tr>
<tr>
<td>RM</td>
<td></td>
</tr>
</tbody>
</table>

**0.1-1.9 BLEU gain over MIRA**

**1.8-2.6 TER gain over MIRA**

<table>
<thead>
<tr>
<th>Optimizer</th>
<th>Sparse feature set</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tune</td>
<td>MT03</td>
</tr>
<tr>
<td>BLEU</td>
<td>BLEU</td>
</tr>
<tr>
<td>MERT</td>
<td></td>
</tr>
<tr>
<td>MIRA</td>
<td></td>
</tr>
<tr>
<td>PRO</td>
<td></td>
</tr>
<tr>
<td>RAMPION</td>
<td></td>
</tr>
<tr>
<td>RM</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Optimizer</th>
<th>Sparse feature set</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tune</td>
<td>MT03</td>
</tr>
<tr>
<td>BLEU</td>
<td>BLEU</td>
</tr>
<tr>
<td>MERT</td>
<td></td>
</tr>
<tr>
<td>MIRA</td>
<td></td>
</tr>
<tr>
<td>PRO</td>
<td></td>
</tr>
<tr>
<td>RAMPION</td>
<td></td>
</tr>
<tr>
<td>RM</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Optimizer</th>
<th>Sparse feature set</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tune</td>
<td>MT03</td>
</tr>
<tr>
<td>BLEU</td>
<td>BLEU</td>
</tr>
<tr>
<td>MERT</td>
<td></td>
</tr>
<tr>
<td>MIRA</td>
<td></td>
</tr>
<tr>
<td>PRO</td>
<td></td>
</tr>
<tr>
<td>RAMPION</td>
<td></td>
</tr>
<tr>
<td>RM</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Optimizer</th>
<th>Sparse feature set</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tune</td>
<td>MT03</td>
</tr>
<tr>
<td>BLEU</td>
<td>BLEU</td>
</tr>
<tr>
<td>MERT</td>
<td></td>
</tr>
<tr>
<td>MIRA</td>
<td></td>
</tr>
<tr>
<td>PRO</td>
<td></td>
</tr>
<tr>
<td>RAMPION</td>
<td></td>
</tr>
<tr>
<td>RM</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Optimizer</th>
<th>Sparse feature set</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tune</td>
<td>MT03</td>
</tr>
<tr>
<td>BLEU</td>
<td>BLEU</td>
</tr>
<tr>
<td>MERT</td>
<td></td>
</tr>
<tr>
<td>MIRA</td>
<td></td>
</tr>
<tr>
<td>PRO</td>
<td></td>
</tr>
<tr>
<td>RAMPION</td>
<td></td>
</tr>
<tr>
<td>RM</td>
<td></td>
</tr>
</tbody>
</table>
## Are we actually bounding the spread?

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

avg (std)
Take Away
Take Away

• Extended relative margin methods to SMT
  – bound spread of the data
  – large-margin in better direction for generalization
Take Away

• Extended relative margin methods to SMT
  – bound spread of the data
  – large-margin in better direction for generalization
• 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
  – bound spread of the data
  – large-margin in better direction for generalization

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

• Substantially outperforms online large-margin solution and batch methods
Outline

• Efficient learning in high-dimensions
  – Online Large-Margin Learning

• Good generalization
  – Online Relative Margin Maximization
  – Adaptation with Topic Models
    (Eidelman, 2012@ACL)

• Structured Latent Output
  – Latent Large-Margin Learning

• Contributions
Domain Adaptation

Parallel Corpus

out

out

in

doc1
doc1
doc1

dev

w

test
Domain Lexical Weighting

(Chiang 2011)
## Domain Lexical Weighting

*(Chiang 2011)*

### Translation Table: nw

| Source       | Target          | P(e|f) |
|--------------|-----------------|-------|
| 粉丝很多       | lots of noodles | 0.41  |
| 粉丝很多       | lots of fans    | 0.32  |
## Domain Lexical Weighting

*(Chiang 2011)*

### Translation Table: nw

<table>
<thead>
<tr>
<th>Source</th>
<th>Target</th>
<th>$P_{s=nw}(e \mid f)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>粉丝很多</td>
<td>lots of noodles</td>
<td>.41</td>
</tr>
<tr>
<td>粉丝很多</td>
<td>lots of fans</td>
<td>.32</td>
</tr>
</tbody>
</table>

### Translation Table: Web

<table>
<thead>
<tr>
<th>Source</th>
<th>Target</th>
<th>$P_{s=wb}(e \mid f)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>粉丝很多</td>
<td>lots of noodles</td>
<td>.30</td>
</tr>
<tr>
<td>粉丝很多</td>
<td>lots of fans</td>
<td>.58</td>
</tr>
</tbody>
</table>
Aims

• Model Domain
  – Induce soft unsupervised domains
    • Latent Topics

• Apply to MT
  – Bias translation model
    • Introduce topic-dependent lexical weighting
Domain Adaptation

Parallel Corpus

doc1

doc4

doc4

doc4

dev

w

test
Lexical Weighting with Topic Models
Lexical Weighting with Topic Models

Translation Table: **Topic 1**

<table>
<thead>
<tr>
<th>Source</th>
<th>Target</th>
<th>$P_{\text{topic}=1}(e \mid f)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>粉丝很多</td>
<td>lots of noodles</td>
<td>.71</td>
</tr>
<tr>
<td>粉丝很多</td>
<td>lots of fans</td>
<td>.15</td>
</tr>
</tbody>
</table>
Lexical Weighting with Topic Models

Translation Table: Topic 1

| Source    | Target          | P_{\text{topic}=1}(e|f) |
|-----------|-----------------|--------------------------|
| 粉丝很多   | lots of noodles | .71                      |
| 粉丝很多   | lots of fans    | .15                      |

Translation Table: Topic 2

| Source    | Target          | P_{\text{topic}=2}(e|f) |
|-----------|-----------------|--------------------------|
| 粉丝很多   | lots of noodles | .41                      |
| 粉丝很多   | lots of fans    | .47                      |
## Lexical Weighting with Topic Models

### Translation Table: Topic 1

| Source | Target      | \(P_{\text{topic}=1}(e|f)\) |
|--------|-------------|-----------------------------|
| 粉丝很多 | lots of noodles | .71                         |
| 粉丝很多 | lots of fans  | .15                         |

### Translation Table: Topic 2

| Source | Target      | \(P_{\text{topic}=2}(e|f)\) |
|--------|-------------|-----------------------------|
| 粉丝很多 | lots of noodles | .41                         |
| 粉丝很多 | lots of fans  | .47                         |

### Translation Table: Topic 3

| Source | Target      | \(P_{\text{topic}=3}(e|f)\) |
|--------|-------------|-----------------------------|
| 粉丝很多 | lots of noodles | .21                         |
| 粉丝很多 | lots of fans  | .68                         |
Lexical Weighting Adaptation Features

Translation Table: Topic 1

<table>
<thead>
<tr>
<th>Source</th>
<th>Target</th>
<th>$P_{\text{topic}}(e \mid f)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>粉丝很多</td>
<td>lots of noodles</td>
<td>.71</td>
</tr>
<tr>
<td>粉丝很多</td>
<td>lots of fans</td>
<td>.15</td>
</tr>
</tbody>
</table>

Translation Table: Topic 2

<table>
<thead>
<tr>
<th>Source</th>
<th>Target</th>
<th>$P_{\text{topic}}(e \mid f)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>粉丝很多</td>
<td>lots of noodles</td>
<td>.41</td>
</tr>
<tr>
<td>粉丝很多</td>
<td>lots of fans</td>
<td>.47</td>
</tr>
</tbody>
</table>

Translation Table: Topic 3

<table>
<thead>
<tr>
<th>Source</th>
<th>Target</th>
<th>$P_{\text{topic}}(e \mid f)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>粉丝很多</td>
<td>lots of noodles</td>
<td>.21</td>
</tr>
<tr>
<td>粉丝很多</td>
<td>lots of fans</td>
<td>.68</td>
</tr>
</tbody>
</table>

Test sentence:

- Topic 1: 0.65
- Topic 2: 0.22
- Topic 3: 0.13
## Lexical Weighting Adaptation Features

**Translation Table: Topic 1**

| Source    | Target             | $P_{\text{topic}(e|f)}$ |
|-----------|--------------------|--------------------------|
| 粉丝很多   | lots of noodles   | .71                      |
| 粉丝很多   | lots of fans      | .15                      |

**Translation Table: Topic 2**

| Source    | Target             | $P_{\text{topic}(e|f)}$ |
|-----------|--------------------|--------------------------|
| 粉丝很多   | lots of noodles   | .41                      |
| 粉丝很多   | lots of fans      | .47                      |

**Translation Table: Topic 3**

| Source    | Target             | $P_{\text{topic}(e|f)}$ |
|-----------|--------------------|--------------------------|
| 粉丝很多   | lots of noodles   | .21                      |
| 粉丝很多   | lots of fans      | .68                      |
Lexical Weighting Adaptation Features

Translation Table: Topic 1

Translation Table: Topic 2

Translation Table: Topic 3

粉丝很多 | | lots of noodles | | $f_1(e|f) = .46$ $f_2(e|f) = .09$ $f_3(e|f) = .02$ $f_1(f|e) f_2(f|e) f_3(f|e)$
Large Margin Training

• Introduce $2k$ features
  – $k= \text{number of topics}$

• Need learning method capable of handling large feature space
Evaluation

• **Chinese-English** (1.6M)
  – NIST MT06 tune, MT03 and MT05 test
• 3-gram LM
• Baselines: no adaptation
  – MERT
  – MIRA
  – RM
• Adaptation using topic dependent lexical weights
  – MIRA
  – RM
Topic Adaptation Results

- BLEU scores for different topic adaptation methods and quantities:
  - MT03:
    - MIRA base: 34.31
    - MERT base: 34.6
    - MIRA +5 topics: 35.21
    - MIRA +10 topics: 35.32
  - MT05:
    - MIRA base: 30.63
    - MERT base: 30.53
    - MIRA +5 topics: 31.47
    - MIRA +10 topics: 31.56

These scores indicate the performance of the models with and without topic adaptation.
RM v. MIRA Results (10 topics)
Take Away

• **Topic modeling** for domain adaptation
  – No reliance on collection/genre annotation
  – Finer-grained domains
  – Biases translation toward topic
    • Lexical weighting adaptation with soft membership
    • Add $P_{\text{topic}}(e|f)$ and $P_{\text{topic}}(f|e)$ features to every rule
• Significantly *improves* translation performance
Outline

• Efficient learning in high-dimensions
  – Online Large-Margin Learning

• Good generalization
  – Online Relative Margin Maximization
  – Adaptation with Topic Models

• Structured Latent Output
  – Latent Large-Margin Learning

• Contributions
Motivation

ein kleines haus
Motivation

ein kleines haus
Motivation

ein kleines haus
Motivation

ein kleines Haus

S₁ X₂

ein a kleines little

ein kleines haus
Motivation

ein kleines haus

S₁ X₂ S₁ X₂

S₁ X₂ S₁ X₂

ein a kleines little haus house

ein kleines haus
Motivation

ein kleines haus $\rightarrow$ a little house
Motivation

• **Derivational** ambiguity
  – exponential number of derivations lead to **same** output string
Motivation

• Derivational ambiguity
  – exponential number of derivations lead to same output string

• Most (~all) systems train and decode toward best single derivation

\[(y^*, d^*) = \arg \max_{(y,d) \in \mathcal{Y}(x), \mathcal{D}(x,y)} w^\top f(x, y, d)\]
Motivation

• Derivational ambiguity
  – exponential number of derivations lead to same output string

• Most (~all) systems train and decode toward best single derivation

\[ y^* = \arg \max_{y \in \mathcal{Y}(x)} \sum_{d \in \mathcal{D}(x, y)} \exp(w^T f(x, y, d)) \]
Motivation

• Derivational ambiguity
  – exponential number of derivations lead to same output string

• Most (~all) systems train and decode toward best single derivation

$$y^* = \arg \max_{y \in \mathcal{Y}(x)} \sum_{d \in \mathcal{D}(x,y)} \exp(w^T f(x, y, d))$$

• Maximum probability translation is NP-hard
Derivation as Translation

1-cost

model score
Derivation as Translation

1-cost

model score

$(y^+, d^+)$
Derivation as Translation

1-cost vs. model score

$(y^+, d^+)$

$(y^-, d^-)$
Derivation as Translation

\[ \sum_{d \in D(x, y^+)} (y^+, d) \]

\[ \sum_{d \in D(x, y^-)} (y^-, d) \]
Why not derivations?

• Features we update toward come from best derivation
  – throwing away (possibly exponential) amount of information
  – may have good translation but bad derivation
Why not derivations?

• Features we update toward come from best derivation
  – throwing away (possibly exponential) amount of information
  – may have good translation but bad derivation

So...want to use all derivations
Latent Large-Margin

- Explicitly model the latent derivation in learning
Latent Large-Margin

• Explicitly model the latent derivation in learning

Hope translation \((y^+, d^+\) \)  
Fear translation \((y^-, d^-)\)
Latent Large-Margin

- Explicitly model the latent derivation in learning

Hope translation: $(y^+, d^+)$

Fear translation: $(y^-, d^-)$

\[
w^\top f(x_i, y^+, d^+) - w^\top f(x_i, y^-, d^-)\]
Latent Large-Margin

- Explicitly model the latent derivation in learning

Hope translation

\[ \mathbf{w}^\top \mathbb{E}_{p(d|x_i,y^+)} \left[ f(\mathbf{x}_i, y^+) \right] - \mathbf{w}^\top \mathbb{E}_{p(d|x_i,y^-)} \left[ f(\mathbf{x}_i, y^-) \right] \]

Fear translation
Two Problems

• Maximum Translation Learning

• Maximum Translation Decoding
Learning with Latent Variables

$$\ell = -\mathbf{w}^\top \mathbf{f}(x_i, y_i) + \frac{1}{\beta_y} \log \sum_{y \in \mathcal{Y}(x_i)} \exp \left[ \beta_y \left\{ \mathbf{w}^\top \mathbf{f}(x_i, y) + \gamma \Delta_i(y) \right\} \right]$$
Learning with Latent Variables

\[ \ell = \frac{1}{\beta_y} \log \sum_{y \in Y(x_i)} \exp \left[ \beta_y \left\{ w^T f(x_i, y) + \gamma \Delta_i(y) \right\} \right] \]

\[ \ell_{LV} = -\frac{1}{\eta_d} \log \sum_{d \in D(x, y_i)} \exp \left[ \eta_d \left\{ w^T f(x_i, y_i, d) \right\} \right] + \frac{1}{\beta_y \eta_d} \log \sum_{y' \in Y(x_i)} \sum_{d \in D(x_i, y')} \exp \left[ \beta_y \eta_d \left\{ w^T f(x_i, y', d) + \gamma \Delta_i(y') \right\} \right] \]
Learning with Latent Variables

\[ \ell = -w^T f(x_i, y_i) + \frac{1}{\beta_y} \log \sum_{y \in \mathcal{Y}(x_i)} \exp \left[ \beta_y \left\{ w^T f(x_i, y) + \gamma \Delta_i(y) \right\} \right] \]

\[ \ell_{LV} = -\frac{1}{\eta_d} \log \sum_{d \in \mathcal{D}(x, y_i)} \exp \left[ \eta_d \left\{ w^T f(x_i, y, d) \right\} \right] + \frac{1}{\beta_y \eta_d} \log \sum_{y' \in \mathcal{Y}(x_i)} \sum_{d \in \mathcal{D}(x_i, y')} \exp \left[ \beta_y \eta_d \left\{ w^T f(x_i, y', d) + \gamma \Delta_i(y') \right\} \right] \]
Learning with Latent Variables

\[ \ell_{LV} = -\frac{1}{\eta_d} \log \sum_{d \in \mathcal{D}(x,y_i)} \exp \left[ \eta_d \left\{ \mathbf{w}^\top \mathbf{f}(x_i, y_i, d) \right\} \right] + \frac{1}{\beta_y} \frac{1}{\eta_d} \log \sum_{y' \in \mathcal{Y}(x_i)} \sum_{d \in \mathcal{D}(x_i, y')} \exp \left[ \beta_y \eta_d \left\{ \mathbf{w}^\top \mathbf{f}(x_i, y', d) + \gamma \Delta_i(y') \right\} \right] \]

\[ \ell_{LSVM} \quad \bullet \text{Maximizes over latent derivations, while others marginalize} \]

\[ \ell_{HCRF} \quad \bullet \text{Maximizes log-likelihood, and does not account for external loss} \]
  \bullet \text{Computes margin between } y^+ \text{ and all others} \]

\[ \ell_{LSMM} \quad \bullet \text{Extends HCRF with external cost} \]
Learning with Latent Variables

\[ \ell_{LV} = \frac{1}{\eta_d} \log \sum_{d \in \mathcal{D}(x, y_i)} \exp \left[ \eta_d \left\{ \mathbf{w}^\top \mathbf{f}(x_i, y_i, d) \right\} \right] + \frac{1}{\beta_y \eta_d} \log \sum_{y' \in \mathcal{Y}(x_i)} \sum_{d \in \mathcal{D}(x_i, y')} \exp \left[ \beta_y \eta_d \left\{ \mathbf{w}^\top \mathbf{f}(x_i, y', d) + \gamma \Delta_i(y') \right\} \right] \]

\[ \ell_{LMM} = \begin{array}{l}
\beta_y \rightarrow \infty \\
\gamma = \eta_d = 1
\end{array} \]

\[ - \log \sum_{d \in \mathcal{D}(x, y_i)} \exp \left\{ \mathbf{w}^\top \mathbf{f}(x_i, y_i, d) \right\} + \max_{y' \in \mathcal{Y}(x_i)} \log \sum_{d \in \mathcal{D}(x_i, y')} \exp \left\{ \mathbf{w}^\top \mathbf{f}(x_i, y', d) + \Delta_i(y') \right\} \]
Learning with Latent Variables

\[ \ell_{LMM} = - \log \sum_{d \in \mathcal{D}(x_i, y_i)} \exp \left\{ w^T f(x_i, y_i, d) \right\} + \max_{y' \in \mathcal{Y}(x_i)} \log \sum_{d \in \mathcal{D}(x_i, y')} \exp \left\{ w^T f(x_i, y', d) + \Delta_i(y') \right\} \]

\[ \ell_{LMM2} = - \max_{y^+ \in \mathcal{Y}(x_i)} \log \sum_{d \in \mathcal{D}(x_i, y^+)} \exp \left\{ w^T f(x_i, y^+, d) - \Delta_i(y^+) \right\} + \max_{y^- \in \mathcal{Y}(x_i)} \log \sum_{d \in \mathcal{D}(x_i, y^-)} \exp \left\{ w^T f(x_i, y^-, d) + \Delta_i(y^-) \right\} \]
Learning with Latent Variables

\[ \ell_{LMM} = - \log \sum_{d \in \mathcal{D}(x_i, y_i)} \exp \left\{ \mathbf{w}^\top \mathbf{f}(x_i, y_i, d) \right\} + \]

\[ \max_{y' \in \mathcal{Y}(x_i)} \log \sum_{d \in \mathcal{D}(x_i, y')} \exp \left\{ \mathbf{w}^\top \mathbf{f}(x_i, y', d) + \Delta_i(y') \right\} \]

\[ \ell_{LMM_2} = - \max_{y^+ \in \mathcal{Y}(x_i)} \log \sum_{d \in \mathcal{D}(x_i, y^+)} \exp \left\{ \mathbf{w}^\top \mathbf{f}(x_i, y^+, d) - \Delta_i(y^+) \right\} \]

\[ + \max_{y^- \in \mathcal{Y}(x_i)} \log \sum_{d \in \mathcal{D}(x_i, y^-)} \exp \left\{ \mathbf{w}^\top \mathbf{f}(x_i, y^-, d) + \Delta_i(y^-) \right\} \]
Two Problems

• **Maximum Translation Learning**
  – What is objective, how to optimize

• **Maximum Translation Decoding**
  – How to get maximum probability translation
Maximum Translation Sampling

• New use of inside-outside sampling (Goodman)
  – Blunsom (2008) adapted from CFG sampling
  – We use for cost-augmented inference
Maximum Translation Sampling

• New use of inside-outside sampling (Goodman)
  – Blunsom (2008) adapted from CFG sampling
  – We use for cost-augmented inference

*ein kleines haus*
Maximum Translation Sampling

• New use of inside-outside sampling (Goodman)
  – Blunsom (2008) adapted from CFG sampling
  – We use for cost-augmented inference

\[ \text{ein kleines haus} \]

\[ \text{some small house} \mid \text{a little house} \mid \text{my little house} \]
Computing Expectations

• Three steps:
Computing Expectations

• Three steps:
  1. Unconstrained decoding
     • Produce the whole translation space of $x$
     • Cost-augmented decoding to obtain $y^+$ and $y^-$
       1) derivations or 2) translation sampling
Computing Expectations

• Three steps:
  1. Unconstrained decoding
     • Produce the whole translation space of $x$
     • Cost-augmented decoding to obtain $y^+$ and $y^-$
   1) derivations or 2) translation sampling
  2. Constrained decoding $<x, y^+>$
Computing Expectations

- Three steps:
  1. Unconstrained decoding
     - Produce the whole translation space of $x$
     - Cost-augmented decoding to obtain $y^+$ and $y^-$
       1) derivations or 2) translation sampling
  2. Constrained decoding $<x, y^+>$
     - Compute expected features for hope
Computing Expectations

• Three steps:
  1. Unconstrained decoding
     • Produce the whole translation space of $x$
     • Cost-augmented decoding to obtain $y^+$ and $y^-$
       1) derivations or 2) translation sampling
  2. Constrained decoding $<x, y^+>$
     • Compute expected features for hope
  3. Constrained decoding $<x, y^->$
     • Compute expected features for fear
Computing Expectations

• Three steps:
  1. Unconstrained decoding
      • Produce the whole translation space of $x$
      • Cost-augmented decoding to obtain $y^+$ and $y^-$
         1) derivations or 2) translation sampling
  2. Constrained decoding $<x, y^+>$
      • Compute expected features for hope
  3. Constrained decoding $<x, y^->$
      • Compute expected features for fear

• Compute update as usual using expectations
Evaluation

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

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

• 4-gram LM (600M words)
# Chinese Results (k-best v. forest)

<table>
<thead>
<tr>
<th>Training</th>
<th>MT03</th>
<th>MT05</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>deriv</td>
<td>deriv</td>
</tr>
<tr>
<td></td>
<td>BLEU</td>
<td>TER</td>
</tr>
<tr>
<td>MIRA k-best</td>
<td>35.8</td>
<td>61.1</td>
</tr>
<tr>
<td>derivation</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MIRA forest</td>
<td>36.2</td>
<td>59.2</td>
</tr>
<tr>
<td>derivation</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Chinese Dense Results

<table>
<thead>
<tr>
<th>Training</th>
<th>MT03</th>
<th>MT05</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>deriv</td>
<td>trans</td>
</tr>
<tr>
<td></td>
<td>BLEU</td>
<td>TER</td>
</tr>
<tr>
<td>MIRA derivation</td>
<td>36.2</td>
<td>59.2</td>
</tr>
<tr>
<td>LSMM translation</td>
<td>32.8</td>
<td>62.4</td>
</tr>
<tr>
<td>LMM derivation</td>
<td>36.5</td>
<td>58.2</td>
</tr>
<tr>
<td>LMM translation</td>
<td>36.9</td>
<td>57.1</td>
</tr>
</tbody>
</table>

**0.9-1.2** BLEU gain over MIRA  
**2.8-3.4** TER gain over MIRA
## Chinese Dense Results

<table>
<thead>
<tr>
<th>Training</th>
<th>MT03 deriv</th>
<th>MT03 trans</th>
<th>MT05 deriv</th>
<th>MT05 trans</th>
</tr>
</thead>
<tbody>
<tr>
<td>MIRA derivation</td>
<td>36.2 59.2</td>
<td>36.7 58.7</td>
<td>32.6 62.7</td>
<td>32.9 61.8</td>
</tr>
<tr>
<td>LSMM translation</td>
<td>32.8 62.4</td>
<td>33.6 61.8</td>
<td>29.1 66.1</td>
<td>29.5 65.6</td>
</tr>
<tr>
<td>LMM derivation</td>
<td>36.5 58.2</td>
<td>36.9 57.7</td>
<td>32.7 61.1</td>
<td>33.0 60.9</td>
</tr>
<tr>
<td>LMM translation</td>
<td>36.9 57.1</td>
<td>37.1 56.4</td>
<td>33.5 59.7</td>
<td>33.8 59.3</td>
</tr>
</tbody>
</table>

**0.9-1.2** BLEU gain over MIRA  
**2.8-3.4** TER gain over MIRA
# Chinese Dense Results

<table>
<thead>
<tr>
<th>Training</th>
<th>MT03</th>
<th>MT05</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>deriv</td>
<td>trans</td>
</tr>
<tr>
<td><strong>MIRA</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>derivation</td>
<td>36.2</td>
<td>59.2</td>
</tr>
<tr>
<td>translation</td>
<td>32.8</td>
<td>62.4</td>
</tr>
<tr>
<td><strong>LMM</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>derivation</td>
<td>36.5</td>
<td>58.2</td>
</tr>
<tr>
<td>translation</td>
<td>36.9</td>
<td>57.1</td>
</tr>
</tbody>
</table>

**0.9-1.2** BLEU gain over MIRA  **2.8-3.4** TER gain over MIRA
# Chinese Dense Results

<table>
<thead>
<tr>
<th>Training</th>
<th>MT03</th>
<th></th>
<th>MT05</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>deriv</td>
<td>trans</td>
<td>deriv</td>
<td>trans</td>
</tr>
<tr>
<td></td>
<td>BLEU</td>
<td>TER</td>
<td>BLEU</td>
<td>TER</td>
</tr>
<tr>
<td>MIRA</td>
<td>36.2</td>
<td>59.2</td>
<td>36.7</td>
<td>58.7</td>
</tr>
<tr>
<td>derivation</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LSMM</td>
<td>32.8</td>
<td>62.4</td>
<td>33.6</td>
<td>61.8</td>
</tr>
<tr>
<td>translation</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LMM</td>
<td>36.5</td>
<td>58.2</td>
<td>36.9</td>
<td>57.7</td>
</tr>
<tr>
<td>derivation</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LMM</td>
<td>36.9</td>
<td>57.1</td>
<td>37.1</td>
<td>56.4</td>
</tr>
<tr>
<td>translation</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**0.9-1.2** BLEU gain over MIRA  
**2.8-3.4** TER gain over MIRA
# Chinese Dense Results

<table>
<thead>
<tr>
<th>Training</th>
<th>MT03</th>
<th>MT05</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>deriv</td>
<td>trans</td>
</tr>
<tr>
<td>MIRA derivation</td>
<td>36.2</td>
<td>59.2</td>
</tr>
<tr>
<td></td>
<td>32.6</td>
<td>62.7</td>
</tr>
<tr>
<td>LSMM translation</td>
<td>32.8</td>
<td>62.4</td>
</tr>
<tr>
<td></td>
<td>29.1</td>
<td>66.1</td>
</tr>
<tr>
<td>LMM derivation</td>
<td>36.5</td>
<td>58.2</td>
</tr>
<tr>
<td></td>
<td>32.7</td>
<td>61.1</td>
</tr>
<tr>
<td>LMM translation</td>
<td>36.9</td>
<td>57.1</td>
</tr>
<tr>
<td></td>
<td>33.5</td>
<td>59.7</td>
</tr>
</tbody>
</table>

### BLEU gain over MIRA

- **0.9-1.2**

### TER gain over MIRA

- **2.8-3.4**
# Chinese Sparse Results

<table>
<thead>
<tr>
<th>Training</th>
<th>MT03</th>
<th>MT05</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>deriv</td>
<td>trans</td>
</tr>
<tr>
<td></td>
<td>BLEU</td>
<td>TER</td>
</tr>
<tr>
<td>MIRA</td>
<td>36.3</td>
<td>58.9</td>
</tr>
<tr>
<td>derivation</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LMM</td>
<td>36.7</td>
<td>57.3</td>
</tr>
<tr>
<td>derivation</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LMM</td>
<td>37.2</td>
<td>56.5</td>
</tr>
<tr>
<td>translation</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

0.9-1.1 BLEU gain over MIRA  
2.6-2.9 TER gain over MIRA
### Arabic Dense Results

<table>
<thead>
<tr>
<th>Training</th>
<th>MT05</th>
<th>MT08</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>deriv</td>
<td>trans</td>
</tr>
<tr>
<td>MIRA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>derivation</td>
<td>52.2</td>
<td>40.9</td>
</tr>
<tr>
<td>LSMM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>translation</td>
<td>48.7</td>
<td>44.6</td>
</tr>
<tr>
<td>LMM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>derivation</td>
<td>53.5</td>
<td>39.3</td>
</tr>
<tr>
<td>LMM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>translation</td>
<td>53.4</td>
<td>39.7</td>
</tr>
</tbody>
</table>

- **BLEU gain over MIRA**: 0.2-2.1
- **TER gain over MIRA**: 1.7-2
# Arabic Sparse Results

<table>
<thead>
<tr>
<th>Training</th>
<th>MT05 deriv</th>
<th>MT05 trans</th>
<th>MT08 deriv</th>
<th>MT08 trans</th>
</tr>
</thead>
<tbody>
<tr>
<td>MIRA derivation</td>
<td>53.1</td>
<td>40.1</td>
<td>53.5</td>
<td>39.8</td>
</tr>
<tr>
<td>LMM derivation</td>
<td>52.9</td>
<td>40.1</td>
<td>53.6</td>
<td>39.6</td>
</tr>
<tr>
<td>LMM translation</td>
<td>54.0</td>
<td>39.2</td>
<td>54.4</td>
<td>39.0</td>
</tr>
</tbody>
</table>

**0.7-1.3** BLEU gain over MIRA  
**0.9-1.1** TER gain over MIRA
Take Away

• Defined general framework of latent variable models
• Presented novel loss for latent large-margin
• Developed optimization procedure for maximum translation decoding and learning
Outline

• Efficient learning in high-dimensions
  – Online Large-Margin Learning
• Good generalization
  – Online Relative Margin Maximization
  – Adaptation with Topic Models
• Structured Output
  – Latent Large-Margin Learning
• Contributions
Summary of Contributions

• Presented generalized form of cost-augmented objectives as family of loss functions
• Conducted comprehensive empirical analysis of optimization performance
  – choice of loss, solver, and parallelization important for generalization performance and learning stability
• Developed tool for large-scale large-margin training and showed it practicability
• Introduced loss for structured relative margin with cost-augmented inference
  – derived an online gradient based solver
  – bounding the spread significantly improves performance over standard large-margin
Summary of Contributions

• Introduced method for dynamic domain adaptation
  – developed unsupervised domain adaptation features

• Defined unified representation of structured latent objectives
  – Previous losses emerge as special case
  – Introduced novel loss for latent large-margin learning
  – Developed optimization procedure for maximum probability translation learning and inference
Software

• Training methods in cdec
  – Cutting-Plane and PA MIRA
  – RM (shortly)

• Mr. MIRA
  – Decoder agnostic online large margin learning on MapReduce

• Topic Model Adaptation
  – part of grammar extraction in cdec
Thank You!