Minimum Imputed Risk: Unsupervised Discriminative Training for Machine Translation

Zhifei Li, Jason Eisner, Ziyuan Wang, Sanjeev Khudanpur, and Brian Roark

Presentation not affiliated with actual authors

January 25, 2012
Typical MT System Training

- $y$: General Mono Data
- $(x, y)$: General Parallel Data
- $p(y|x)$: Translation Model(s)
- $\theta$: Model Parameters
- $L(y, y')$: Loss Function
- $x$: In-Domain Parallel Data (source)
- $y$: In-Domain Parallel Data (target)
- $y'$: Translation Hypotheses
Roadblock: No Parallel In-Domain Data
Solution: Fill in the Missing Data

\[ p(y|x) \]

\[ (x, y) \]

\[ \theta \]

\[ L(y, y') \]

\[ p(x|y) \]

\[ x' \]

\[ y \]

\[ p(x|y) \]

Reverse Model

In-Domain Source Data (guess)

In-Domain Target Data

General Mono Data

General Parallel Data

Translation Model(s)

Translation Hypotheses

Model Parameters

Loss Function
When is this Possible?

Requirements:

1. Enough general parallel data to build two MT systems: 
   \( p_\theta(y|x) \) and \( p_\phi(x|y) \)

2. A small amount of parallel in-domain data to tune the few parameters \( \phi \)

3. A large amount of in-domain target side monolingual data

For example: want to build syntactic MT system, only have enough parallel data to train very simple system.
Supervised Discriminative Training

Translating source sentences $x$ to target hypotheses $y'$:

$$\delta_{\theta}(x) = y'$$
Supervised Discriminative Training

Translating source sentences $x$ to target hypotheses $y'$:

$$\delta_\theta(x) = y'$$

Select loss function $L$ (usually BLEU) to score against correct translations $y$:

$$L(y', y)$$
Supervised Discriminative Training

Translating source sentences $x$ to target hypotheses $y'$:

$$\delta_\theta(x) = y'$$

Select loss function $L$ (usually BLEU) to score against correct translations $y$:

$$L(y', y)$$

Goal: find $\theta$ with low Bayes Risk. For MT tuning, use empirical risk:

$$\theta^* = \arg \min_\theta \frac{1}{N} \sum_{i=1}^{N} L(\delta_\theta(x_i), y_i)$$
Unsupervised Discriminative Training

We have $y_i$ but not $x_i$, so loss function becomes “round trip” cost:

$$L(\delta_\theta(x_i), y_i) \quad \text{becomes} \quad \sum_x p_\phi(x|y_i) L(\delta_\theta(x), y_i)$$
Unsupervised Discriminative Training

We have \( y_i \) but not \( x_i \), so loss function becomes “round trip” cost:

\[
L(\delta_\theta(x_i), y_i) \quad \text{becomes} \quad \sum_x p_\phi(x | y_i) L(\delta_\theta(x), y_i)
\]

Plug into objective function to minimize *imputed empirical risk*:

\[
\theta^* = \arg \min_{\theta} \frac{1}{N} \sum_{i=1}^{N} \sum_x p_\phi(x | y_i) L(\delta_\theta(x), y_i)
\]

How do we sum over all possible translations \( x \)?
Reverse Prediction Model

Model $p_\phi(x|y)$ translates from target to source

- Advantage: can use in-domain monolingual source data $x$
Reverse Prediction Model

Model $p_\phi(x|y)$ translates from target to source

- Advantage: can use in-domain monolingual source data $x$

$\delta_\theta$ and $p_\phi$ are not symmetric:

- $\delta_\theta$ is a function that produces the single best translation
- $p_\phi$ is a probability distribution over possible values of missing input sentence

Ideal: Train $\phi$ to match underlying conditional distribution, having low cross-entropy $H(X|Y)$. Approximate with:

$$-\frac{1}{M} \sum_{j=1}^{N} \log p_\phi(x_j|y_j) + \frac{1}{2} \sigma^2 ||\phi||_2^2$$
Reverse Prediction Model

Model $p_\phi(x|y)$ translates from target to source

- Advantage: can use in-domain monolingual source data $x$

$\delta_\theta$ and $p_\phi$ are not symmetric:

- $\delta_\theta$ is a function that produces the single best translation
- $p_\phi$ is a probability distribution over possible values of missing input sentence

Ideal: Train $\phi$ to match underlying conditional distribution, having low cross-entropy $H(X|Y)$. Approximate with:

$$- \frac{1}{M} \sum_{j=1}^{N} \log p_\phi(x_j|y_j) + \frac{1}{2\sigma^2} ||\phi||_2^2$$
Forward Translation

Simple (deterministic) decoding: $\delta_\theta(x) = \arg \max_y p_\theta(y|x)$

- Equivalent to MERT on imputed data when $L$ is negated BLEU
- Objective function not differentiable, line search does not scale
Forward Translation

Simple (deterministic) decoding: \( \delta_\theta(x) = \arg \max_y p_\theta(y|x) \)
- Equivalent to MERT on imputed data when \( L \) is negated BLEU
- Objective function not differentiable, line search does not scale

Randomized decoding: system outputs \( y \) with probability \( p_\theta(y|x) \)

Minimum imputed empirical risk:

\[
\theta^* = \arg \min_{\theta} \frac{1}{N} \sum_{i=1}^{N} \sum_{x,y} p_\phi(x|y_i)p_\theta(y|x)L(y, y_i)
\]

Now differentiable, can optimize with gradient-based methods
Approximating $p_\theta(x|y_i)$
Approximating $p_\theta(x|y_i)$

Exhaustive?

- Computationally infeasible
Approximating $p_\theta(x|y_i)$

Exhaustive?
- Computationally infeasible

$k$-best?
- Extract top $k$ highest scoring translations, rescale probability to 1
Approximating $p_\theta(x|y_i)$

Exhaustive?

- Computationally infeasible

$k$-best?

- Extract top $k$ highest scoring translations, rescale probability to 1

Sampling?

- Take $k$ independent samples with weight $\frac{1}{k}$ from $p_\phi(x|y_i)$ for each $y_i$
Approximating $p_\theta(x|y_i)$

Exhaustive?
- Computationally infeasible

$k$-best?
- Extract top $k$ highest scoring translations, rescale probability to 1

Sampling?
- Take $k$ independent samples with weight $\frac{1}{k}$ from $p_\phi(x|y_i)$ for each $y_i$

Lattice?
- Theoretical contribution: efficient exact computation under certain conditions using dynamic programming
Approximating $p_\theta(x|y_i)$

Rule-level?

- For Hiero systems, require complete isomorphism of SCFG trees for forward and reverse translations
- Forward translations decompose according to existing parse tree of $x_i$
- Exploits structure sharing to score entire hypergraph (round-trip translate at the rule level)
Approximating $p_\theta(x|y_i)$

Rule-level?

- For Hiero systems, require complete isomorphism of SCFG trees for forward and reverse translations
- Forward translations decompose according to existing parse tree of $x_i$
- Exploits structure sharing to score entire hypergraph (round-trip translate at the rule level)

Actually used:

- 1-best approximation
Experiments

Chinese-English Joshua (Hiero) system with large number of target-rule bigram features

IWSLT Task:
- 40K sentence pairs train, 503 dev, 506 test
- 16 references per sentence
- 551 features, 5-gram LM on parallel data only

NIST Task:
- 1M sentence pairs train, 919 dev, 1788 for unsupervised, 1082/1099 test
- 4 references per sentence
- 1033 features, 5-gram LM on 130M words from Gigaword
## Semi-Supervised Results

<table>
<thead>
<tr>
<th>IWSLT</th>
<th>Training</th>
<th>Test BLEU</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sup (200 zh-en)</td>
<td></td>
<td>47.6</td>
</tr>
<tr>
<td>+Unsup (101 en)</td>
<td></td>
<td>49.0</td>
</tr>
<tr>
<td>+Unsup (202 en)</td>
<td></td>
<td>48.9</td>
</tr>
<tr>
<td>+Unsup (303 en)</td>
<td></td>
<td>49.7</td>
</tr>
</tbody>
</table>

Small data scenario (40K sent)

<table>
<thead>
<tr>
<th>NIST</th>
<th>Training</th>
<th>MT05</th>
<th>MT06</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sup (919 zh-en)</td>
<td></td>
<td>32.4</td>
<td>30.6</td>
</tr>
<tr>
<td>+Unsup (1788 en)</td>
<td></td>
<td>33.0</td>
<td>31.1</td>
</tr>
</tbody>
</table>

Medium data scenario (1M sent)
Unsupervised Results (All IWSLT)

<table>
<thead>
<tr>
<th>Data size</th>
<th>Chinese BLEU</th>
<th>English BLEU</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>WLM</td>
<td>NLM</td>
</tr>
<tr>
<td>101</td>
<td>11.8</td>
<td>3.0</td>
</tr>
<tr>
<td>202</td>
<td>11.7</td>
<td>3.2</td>
</tr>
<tr>
<td>303</td>
<td>13.4</td>
<td>3.5</td>
</tr>
</tbody>
</table>

Varying strength of reverse prediction system

<table>
<thead>
<tr>
<th>$k$-best size</th>
<th>Test BLEU</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>48.5</td>
</tr>
<tr>
<td>2</td>
<td>48.4</td>
</tr>
<tr>
<td>3</td>
<td>48.9</td>
</tr>
<tr>
<td>4</td>
<td>48.5</td>
</tr>
<tr>
<td>5</td>
<td>48.4</td>
</tr>
</tbody>
</table>

Varying $k$-best size with 101 sentence dev set
EM:

E step, expected log-likelihood of complete data:

\[
\sum_{x} p_{\theta t}(x|y_i) \log p_{\theta}(x, y_i)
\]

M step, maximize:

\[
\theta_{t+1} = \arg \max_{\theta} \frac{1}{N} \sum_{i=1}^{N} \sum_{x} p_{\theta t}(x|y_i) \log p_{\theta}(x, y_i)
\]
Minimum Imputed Risk and EM

Minimum Imputed Risk:

Change $p_{\theta t}(x|y_i)$ to $p_{\phi}(x|y_i)$ and admit negative log-likelihood as objective function:

$$\theta^* = \arg \min_{\theta} \frac{1}{N} \sum_{i=1}^{N} \sum_x p_{\phi}(x|y_i) L(\delta_{\theta}(x), y_i)$$

Advantages over EM:

- Discriminative, incorporates loss function in training
- Training joint models is expensive, MIR works with conditional models
Discussion

Advantages:

- Use large monolingual data on both source and target side
- Idea could be used to enrich existing MT systems

Issues:

- IWSLT: 200 dev sentences < 551 features
  - Significant improvement expected from adding (degraded) dev sentences

Additional Experiments?

- Semi-supervised vs fully supervised? How close is the result?
- Generate additional dev sentences for existing data sets? Improve via paraphrasing effect?
Discussion

Advantages:
- Use large monolingual data on both source and target side
- Idea could be used to enrich existing MT systems

Issues:
- IWSLT: 200 dev sentences < 551 features
- Significant improvement expected from adding (degraded) dev sentences
Discussion

Advantages:

- Use large monolingual data on both source and target side
- Idea could be used to enrich existing MT systems

Issues:

- IWSLT: 200 dev sentences < 551 features
- Significant improvement expected from adding (degraded) dev sentences

Additional Experiments?:

- Semi-supervised vs fully supervised? How close is the result?
- Generate additional dev sentences for existing data sets? Improve via paraphrasing effect?
Minimum Imputed Risk: Unsupervised Discriminative Training for Machine Translation
Zhifei Li, Jason Eisner, Ziyuan Wang, Sanjeev Khudanpur, and Brian Roark

Presentation not affiliated with actual authors

January 25, 2012