Discriminative Training
for Phrase-Based Machine Translation

Abhishek Arun

19 April 2007
Overview

• Evolution from generative to discriminative models
• Discriminative training
• Model
• Learning schemes
• Featured representation
• The reference dilemma
• Experiments
• Future work
• Conclusion
The birth of SMT: generative models

• The definition of translation probability follows a mathematical derivation

\[
\arg\max_ep(e|f) = \arg\max_ep(f|e) p(e)
\]

• Occasionally, some independence assumptions are thrown in for instance IBM Model 1: word translations are independent of each other

\[
p(e|f, a) = \frac{1}{Z} \prod_i p(e_i|f_a(i))
\]

• Generative model leads to straight-forward estimation
  – maximum likelihood estimation of component probability distribution
  – EM algorithm for discovering hidden variables (alignment)
Log-linear models

• Alternative to Equation 1: Model **posterior probability directly**:

\[
p(e|f) = \frac{\exp[\sum_{m=1}^{M} \lambda_m h_m(e, f)]}{\sum_{e'} \exp[\sum_{m=1}^{M} \lambda_m h_m(e', f)]}
\]  

(2)

• Decision rule is now:

\[
\hat{e} = \arg\max_e p(e|f) = \arg\max_e \left[ \sum_{m=1}^{M} \lambda_m h_m(e, f) \right]
\]
Discriminative training

- **Modeling problem:**
  - Come up with sensible features.
- **Training problem:**
  - Come up with suitable lambdas.
- Most estimation procedures in NLP maximize likelihood of training data.
- However at test time model is evaluated wrt to some **loss function**
- Idea:
  - Minimize loss on training data
Discriminative training

Model

- generate n-best list
  - 1
  - 2
  - 3
  - 4
  - 5
  - 6

- score translations
  - 1
  - 2
  - 3
  - 4
  - 5
  - 6

- change feature weights
  - 1
  - 2
  - 3
  - 4
  - 5
  - 6

- find feature weights that move up good translations
Och’s minimum error rate training (MERT)

- **Line search** for best feature weights

```
given: sentences with n-best list of translations
iterate n times
    randomize starting feature weights
    iterate until convergences
    for each feature
        find best feature weight
        update if different from current
return best feature weights found in any iteration
```
BLEU error surface

- Varying one parameter: a ragged line with many local optima
Unstable outcomes: scores vary

- Even different scores with different runs (varying 0.40 on dev, 0.89 on test)

<table>
<thead>
<tr>
<th>run</th>
<th>iterations</th>
<th>dev score</th>
<th>test score</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>8</td>
<td>50.16</td>
<td>51.99</td>
</tr>
<tr>
<td>2</td>
<td>9</td>
<td>50.26</td>
<td>51.78</td>
</tr>
<tr>
<td>3</td>
<td>8</td>
<td>50.13</td>
<td>51.59</td>
</tr>
<tr>
<td>4</td>
<td>12</td>
<td>50.10</td>
<td>51.20</td>
</tr>
<tr>
<td>5</td>
<td>10</td>
<td>50.16</td>
<td>51.43</td>
</tr>
<tr>
<td>6</td>
<td>11</td>
<td>50.02</td>
<td>51.66</td>
</tr>
<tr>
<td>7</td>
<td>10</td>
<td>50.25</td>
<td>51.10</td>
</tr>
<tr>
<td>8</td>
<td>11</td>
<td>50.21</td>
<td>51.32</td>
</tr>
<tr>
<td>9</td>
<td>10</td>
<td>50.42</td>
<td>51.79</td>
</tr>
</tbody>
</table>
More features: more components

• We would like to add more components to our model
  – multiple language models
  – domain adaptation features
  – various special handling features
  – using linguistic information

→ MERT becomes even less reliable
  – runs many more iterations
  – fails more frequently
More features: factored models

- Factored translation models break up phrase mapping into smaller steps
  - multiple translation tables
  - multiple generation tables
  - multiple language models and sequence models on factors

→ Many more features
Millions of features

• Why mix of discriminative training and generative models?

• Discriminative training of all components
  – phrase table [Liang et al., 2006]
  – language model [Roark et al, 2004]
  – additional features

• Large-scale discriminative training
  – millions of features
  – training of full training set, not just a small development corpus
Model

SMT as a **structured prediction** task.

- Local score:
  \[ s(f_i, e_i) = w \cdot \Phi(f_i, e_i) \]

- Translation score:
  \[
  s(f, e) = \sum_{(f_i, e_i) \in e} s(f_i, e_i)
  = \sum_{(f_i, e_i) \in e} w \cdot \Phi(f_i, e_i)
  \]

- Decoding:
  \[ \hat{e} = \arg\max_e s(f, e) \]
Featured representation

\[ s(f_i, e_i) = w \cdot \Phi(f_i, e_i) \]

- \( \Phi \): multidimensional feature vector representation
- Can throw in arbitrary features in the model
  - Model can learn from negative evidence e.g. downweight “the the”
  - Complex interactions between features
Example

\[ \Phi_{100}(f, e) = \begin{cases} 
1 & \text{if } f_i = \text{“les expressions de”} \land e_i = \text{“expressions of”} \\
0 & \text{otherwise} 
\end{cases} \]

\[ \Phi_{241}(f, e) = \begin{cases} 
1 & \text{if distortion} = 0 \land f_{i-1} = \text{“START”} \land f_i = \text{“les expressions de”} \\
0 & \text{otherwise} 
\end{cases} \]
Example

\[ \Phi_{729}(f, e) = \begin{cases} 
1 & \text{if } \text{last2TgtWords} = \text{“of equality”} \\
0 & \text{otherwise} 
\end{cases} \]

\[ \Phi_{730}(f, e) = \begin{cases} 
1 & \text{if } \text{last3TgtWords} = \text{“expressions of equality”} \\
0 & \text{otherwise} 
\end{cases} \]
Example

\[ \Phi_{317}(f, e) = \begin{cases} 
1 & \text{if orientation} = \text{"MONO"} \land f_{i-1} = \text{"les expressions de"} \\
& \land f_i = \text{"parite"} \land e_{i-1} = \text{"expressions of"} \\
& \land e_i = \text{"equality"} \\
0 & \text{otherwise}
\end{cases} \]
Training regimes

\[ s(f, e) = \sum_{(f_i, e_i) \in e} w \cdot \Phi(f_i, e_i) \]

- Supervised training: given training set \( T = \{(f_t, e_t)\}_{t=1}^T \), estimate \( w \)
  - Likelihood based models:
    * Expectations of features across the structure
  - Margin-based methods:
    * n-best or marginal distribution across graphical structure
    * Perceptron [Collins, 2002]: only need argmax computation
    * Approximate large margin: MIRA [Crammer and Singer, 2003]
Perceptron

Requirements:

• Training data: \( T = \{(f_t, e_t)\}_{t=1}^T \)

• \( \hat{e} = \arg\max_e s(f, e) \)
  
  - Exact computation intractable \( \rightarrow \) beam search

• \( \Phi(f_t, \hat{e}) \)

• \( \Phi(f_t, e_t) \)

Update rule: \( w^{(i+1)} = w^i + \Phi(f_t, e_t) - \Phi(f_t, \hat{e}) \)

Intuition:

• Boost features in correct output and penalise features in incorrect prediction
MIRA

Requirements:

- $T, \hat{e}, \Phi(f_t, \hat{e}), \Phi(f_t, e_t)$
- Loss function, $L(e_t, \hat{e}) \rightarrow$ measures goodness of prediction wrt to gold standard

Updates weighted by loss:

$$
\begin{align*}
\min |w_{i+1} - w_i| \\
\text{s.t.} & \quad s(f_t, e_t) - s(f_t, \hat{e}) \geq L(e_t, \hat{e}) \\
\forall \hat{e} & \quad \in \text{best}_k(f_t; w^{(i)})
\end{align*}
$$
Problem: overfitting

- Fundamental problem in machine learning
  - what works best for training data, may not work well in general
  - rare, unrepresentative features may get too much weight

- Especially severe problem in phrase-based models
  - long phrase pairs explain well individual sentences
  - ... but are less general, suspect to noise
  - EM training of phrase models [Marcu and Wong, 2002] has same problem
Solutions

• **Restrict to short phrases**, e.g., maximum 3 words (current approach)
  – limits the power of phrase-based models
  – ... but not very much [Koehn et al, 2003]

• **Restrict to short features**: window of 3 words

• **Jackknife**
  – collect phrase pairs from one part of corpus
  – optimize their feature weights on another part

• **IBM direct model**: **only one-to-many** phrases [Ittycheriah and Salim Roukos, 2007]
Problem: reference translation

- Supervised training assumes knowledge of gold standard, but...
- Reference translation may not be produceable by model

Diagram:
- All English sentences
- Produceable by model
- Covered by search
Problem: reference translation

- If produceable by model → we can compute feature scores
- If not → we can not
- Matching reference string not enough, we want to learn from good phrasal alignments too.

Multiple ways of going from source to target (if reachable). Is there a reference phrasal alignment?
- Let’s just ignore alignments for now...
Update strategies

- **Skip sentences**, for which reference can not be produced
  - invalidates large amounts of training data, biases model to shorter sentences

- Declare candidate translations closest to reference as **surrogate**
  - closeness measured for instance by smoothed BLEU score
  - may be not a very good translation: odd feature values, training is severely distorted
Update strategies

- **Local update:**
  - When including all sentences: surrogate reference picked from 1000-best list using maximum *smoothed BLEU score* with respect to reference translation.
  - **Dynamic reranking.**

- **Min Loss update:**
  - Modify regular decoder to use smoothed BLEU as scoring function.
  - Store min loss candidate for each training instance.
Experiments

Czech-English task - Prague Dependency treebank, 21K training sentences. Only binary features

- phrase table features
- lexicalized reordering features
- distortion features
- source and target phrase ngram
## Results

<table>
<thead>
<tr>
<th>Training scheme</th>
<th>BLEU</th>
<th>Length ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pharaoh - MERT</td>
<td>34.53</td>
<td>0.978</td>
</tr>
<tr>
<td>Perceptron - local</td>
<td>28.09</td>
<td>0.906</td>
</tr>
<tr>
<td>1-best MIRA - local</td>
<td>27.64</td>
<td>0.911</td>
</tr>
<tr>
<td>Perceptron - min loss</td>
<td>24.04</td>
<td>0.881</td>
</tr>
<tr>
<td>1-best MIRA - min loss</td>
<td>25.24</td>
<td>0.881</td>
</tr>
</tbody>
</table>
Discussion

- Min Loss performing much worse than local updates - why?
- Local updates more conservative than min loss update
- Loss function ignores alignments
- Can produce “good” translations using “dodgy” alignments.
- Loss function insensitive to paraphrasing

- Short output - model bias?
Summary

- Discriminative models allow us to incorporate lots of features
- Proposed model = millions of features (phrase pair, ngram, lexicalised reordering)
- Train on whole corpus
- Margin based learning algorithms
- Problems:
  - Discriminative training: Requires featured representation of gold standard
  - Featured representation of gold standard not always available
  - Model biased towards short output
Future work

• What is a good reference? Paraphrasing to extend reference set.
• Loss functions - sensitive to alignments, lexical choices etc
• mix of binary and real-valued features
• scaling up

More and more features are unavoidable, let’s deal with them