Sparse Feature Learning

Philipp Koehn

1 March 2016
Multiple Component Models
Component Weights

- Translation Model: 0.26
- Language Model: 0.19
- Reordering Model: 0.21

Weights:
- Translation Model: 0.26
- Language Model: 0.19
- Reordering Model: 0.21
- Other weights: 0.04, 0.05, 0.06, 0.1, 0.1

Philipp Koehn
Even More Numbers Inside

Translation Model

Language Model

Reordering Model

- $p(a \mid to) = 0.18$
- $p(casa \mid house) = 0.35$
- $p(azur \mid blue) = 0.77$
- $p(la \mid the) = 0.32$
Grand Vision

• There are millions of parameters
  – each phrase translation score
  – each language model n-gram
  – etc.

• Can we train them all discriminatively?

• This implies optimization over the entire training corpus
aligned corpus

rule scores

millions
A diagram showing the iterative n-best search space. Key points include:

- **MERT** [Och et al., 2003]
- **PRO** [Hopkins/May 2011]
- **MIRA** [Chiang 2007]
- **SampleRank** [Haddow et al., 2011]
- **MaxViolation** [Yu et al., 2013]
- **Leave One Out** [Wuebker et al., 2012]

The x-axis represents the aligned corpus size, ranging from "a handful" to "millions". The y-axis represents the n-best space. The diagram illustrates how the size of the corpus influences the search space, with smaller corpora requiring less computational resources but potentially lower quality translations, and larger corpora allowing for more comprehensive search but at a higher computational cost.
Strategy and Core Problems

- Process each sentence pair in the training corpus

- Optimize parameters towards producing the reference translation

- Reference translation may not be producible by model
  - optimize towards most similar translation
  - or, only process sentence pair partially

- Avoid overfitting

- Large corpora require efficient learning methods
Sentence Level vs. Corpus Level Error Metric

- Optimizing BLEU requires optimizing over the entire training corpus

\[
\text{BLEU}(\{e_{\text{best}}^i = \arg\max_{e_i} \sum_j h_j(e_i, f_i) \lambda_i \}, \{e_{i}^{\text{ref}}\})
\]

- Life would be easier, if we could sum over sentence level scores

\[
\sum_i \text{BLEU}'(\arg\max_{e_i} \sum_j (h_j(e_i, f_i) \lambda_i), e_{i}^{\text{ref}})
\]

- For instance, BLEU+1
features
Core Rule Properties

- Frequency of phrase (binned)

- Length of phrase
  - number of source words
  - number of target words
  - number of source and target words

- Unaligned / added (content) words in phrase pair

- Reordering within phrase pair
Lexical Translation Features

- $\text{lex}(e)$ fires when an output word $e$ is generated
- $\text{lex}(f, e)$ fires when an output word $e$ is generated aligned to a input word $f$
- $\text{lex}(\text{NULL}, e)$ fires when an output word $e$ is generated unaligned
- $\text{lex}(f, \text{NULL})$ fires when an input word $e$ is dropped
- Could also be defined on part of speech tags or word classes
Lexicalized Reordering Features

- Replacement of lexicalized reordering model

- Features differ by
  - lexicalized by first or last word of phrase (source or target)
  - word representation replaced by word class
  - orientation type
Domain Features

- Indicator feature that the rule occurs in one specific domain
- Probability that the rule belongs to one specific domain
- Domain-specific lexical translation probabilities
Syntax Features

• If we have syntactic parse trees, many more features
  – number of nodes of a particular kind
  – matching of source and target constituents
  – reordering within syntactic constituents

• Parse trees are a by-product of syntax-based models

• More on that in future lectures
Every Number in Model

- Phrase pair indicator feature
- Target n-gram feature
- Phrase pair orientation feature
perceptron algorithm
Optimizing Linear Model

- We consider each sentence pair \((e_i, f_i)\) and its alignment \(a_i\)

- To simplify notation, we define derivation \(d_i = (e_i, f_i, a_i)\)

- Model score is weighted linear combination of feature values \(h_j\) and weights \(\lambda_j\)

\[
\text{score}(\lambda, d_i) = \sum_j \lambda_j h_j(d_i)
\]

- Such models are also known as single layer perceptrons

![Diagram of a single layer perceptron](image)
Reference and Model Best

• Besides the reference derivation $d_i^{\text{ref}}$ for sentence pair $i$ and its score

$$\text{score}(\lambda, d_i^{\text{ref}}) = \sum_j \lambda_j h_j(d_i^{\text{ref}})$$

• We also have the model best translation

$$d_i^{\text{best}} = \arg\max_d \text{score}(\lambda, d_i) = \arg\max_d \sum_j \lambda_j h_j(d_i)$$

• ... and its model score

$$\text{score}(\lambda, d_i^{\text{best}}) = \sum_j \lambda_j h_j(d_i^{\text{best}})$$

• We can view the error in our model as a function of its parameters $\lambda$

$$\text{error}(\lambda, d_i^{\text{best}}, d_i^{\text{ref}}) = \text{score}(\lambda, d_i^{\text{best}}) - \text{score}(\lambda, d_i^{\text{ref}})$$
Follow the Direction of Gradient

- Assume that we can compute the gradient $\frac{d}{d\lambda} \text{error}(\lambda)$ at any point.
- If the error curve is convex, gradient points in the direction the optimum.
Move Relative to Steepness

- If the error curve is convex, size of gradient indicates speed of change
- Model update $\Delta \lambda = -\frac{d}{d\lambda} \text{error}(\lambda)$
Stochastic Gradient Descent

- We want to minimize the error
  \[
  \text{error}(\lambda, \text{d}^{\text{best}}_i, \text{d}^{\text{ref}}_i) = \text{score}(\lambda, \text{d}^{\text{best}}_i) - \text{score}(\lambda, \text{d}^{\text{ref}}_i)
  \]

- In stochastic gradient descent, we follow direction of gradient
  \[
  \frac{d}{d \lambda} \text{error}(\lambda, \text{d}^{\text{best}}_i, \text{d}^{\text{ref}}_i)
  \]

- For each \(\lambda_j\), we compute the gradient pointwise
  \[
  \frac{d}{d \lambda_j} \text{error}(\lambda_j, \text{d}^{\text{best}}_i, \text{d}^{\text{ref}}_i) = \frac{d}{d \lambda_j} \text{score}(\lambda, \text{d}^{\text{best}}_i) - \text{score}(\lambda, \text{d}^{\text{ref}}_i)
  \]
Stochastic Gradient Descent

• Gradient with respect to $\lambda_j$

$$\frac{d}{d \lambda_j} \text{error}(\lambda_j, d^\text{best}_i, d^\text{ref}_i) = \frac{d}{d \lambda_j} \sum_{j'} \lambda_{j'} h_{j'}(d^\text{best}_i) - \sum_{j'} \lambda_{j'} h_{j'}(d^\text{ref}_i)$$

• For $\lambda'_{j} \neq \lambda_{j}$, the terms $\lambda_{j'} h_{j'}(d_i)$ are constant, so they disappear

$$\frac{d}{d \lambda_j} \text{error}(\lambda_j, d^\text{best}_i, d^\text{ref}_i) = \frac{d}{d \lambda_j} \lambda_j h_j(d^\text{best}_i) - \lambda_j h_j(d^\text{ref}_i)$$

• The derivative of a linear function is its factor

$$\frac{d}{d \lambda_j} \text{error}(\lambda_j, d^\text{best}_i, d^\text{ref}_i) = h_j(d^\text{best}_i) - h_j(d^\text{ref}_i)$$

⇒ Our model update is $\lambda_{j}^{\text{new}} = \lambda_{j} - (h_j(d^\text{best}_i) - h_j(d^\text{ref}_i))$
Intuition

- Feature values in model best translation

- Feature values in reference translation

- Intuition:
  - promote features whose value is bigger in reference
  - demote features whose value is bigger in model best
Algorithm

**Input:** set of sentence pairs \((e, f)\), set of features

**Output:** set of weights \(\lambda\) for each feature

1. \(\lambda_i = 0\) for all \(i\)
2. while not converged do
3.   for all foreign sentences \(f\) do
4.     \(d_{\text{best}} = \text{best derivation according to model}\)
5.     \(d_{\text{ref}} = \text{reference derivation}\)
6.     if \(d_{\text{best}} \neq d_{\text{ref}}\) then
7.       for all features \(h_i\) do
8.         \(\lambda_i += h_i(d_{\text{ref}}) - h_i(d_{\text{best}})\)
9.       end for
10.   end if
11. end for
12. end while
generating the reference
Failure to Generate Reference

- Reference translation may be anywhere in this box

```
all English sentences
produceable by model
covered by search
```

- If produceable by model → we can compute feature scores
- If not → we can not
Causes

- Reference translation in tuning set not literal
- Failure even if phrase pairs are extracted from same sentence pair
- Examples

  - alignment points too distant → phrase pair too big to extract
  - required reordering distance too large → exceeds distortion limit of decoder
Sentence Level BLEU

• BLEU+1
  – add one free n-gram count to statistics → avoids BLEU score of 0
  – however: wrong balance between 1-4 grams, too drastic brevity penalty

• BLEU impact
  – leave all other sentence translations fixed
  – collect n-gram matches and totals from them
  – add n-gram matches and total from current candidate
  → consider impact on overall BLEU score

• Incremental BLEU impact
  – maintain decaying statistics for n-gram matches, total n-grams

\[
\text{count}_t = \frac{9}{10} \text{count}_{t-1} + \text{current-count}_t
\]
Problems with Max-BLEU Training

- Consider the following Arabic sentence (written left-to-right in Buckwalter romanization) with English glosses:

\[
\text{sd} \quad \text{qTEp} \quad \text{mn} \quad \text{AlkEk} \quad \text{AlmmlH} \quad \text{"brytzl"} \quad \text{Hlqh} \quad .
\]

\text{blocked} \quad \text{piece} \quad \text{of} \quad \text{biscuit} \quad \text{salted} \quad \text{"pretzel"} \quad \text{his-throat}

- Very literal translation might be

A piece of a salted biscuit, a "pretzel," blocked his throat.

- But reference translation is

A pretzel, a salted biscuit, became lodged in his throat.

- Reference accurate, but major transformations

- Trying to approximate reference translation may lead to bad rules

note: example from Chiang (2012)
mira
Hope and Fear

- Bad: optimize towards utopian, away from n-best
- Good: optimize towards hope, away from fear
Hope and Fear Translations

- Hope translation
  \[ d^{\text{hope}} = \arg\max_d \text{BLEU}(d) + \text{score}(d) \]

- Finding the fear translation
  - Metric difference (should be big)
    \[ \Delta\text{BLEU}(d^{\text{hope}}, d) = \text{BLEU}(d^{\text{hope}}) - \text{BLEU}(d) \]
  - Score difference (should be small or negative)
    \[ \Delta\text{score}(\lambda, d^{\text{hope}}, d) = \text{score}(\lambda, d^{\text{hope}}) - \text{score}(\lambda, d) \]
  - Margin
    \[ v(\lambda, d^{\text{hope}}, d) = \Delta\text{BLEU}(d^{\text{hope}}, d) - \Delta\text{score}(\lambda, d^{\text{hope}}, d) \]
  - Fear translation
    \[ d^{\text{fear}} = \arg\max_d v(\lambda, d^{\text{hope}}, d) \]
Margin Infused Relaxed Algorithm (MIRA)

- Stochastic gradient descent update with learning weight $\delta_i$

$$
\lambda_j^{\text{new}} = \lambda_j - \delta_i \left( h_j(d_i^{\text{fear}}) - h_j(d_i^{\text{hope}}) \right)
$$

- Updates should depend on margin

$$
\delta_i = \min \left( C, \frac{\Delta \text{BLEU}(d_i^{\text{hope}}, d_i^{\text{fear}}) - \Delta \text{score}(d_i^{\text{hope}}, d_i^{\text{fear}})}{||\Delta h||^2} \right)
$$

- The math behind this is a bit complicated
Different Learning Rates for Features

• For some features, we have a lot of evidence (coarse features)

• Others occur only rarely (sparse features)

• After a while, we do not want to change coarse features too much

⇒ Adaptive Regularization of Weights (AROW)
   - record confidence in weights over time
   - include this in the learning rate for each feature
Parallelization

• Training is computationally expensive

⇒ Break up training data into batches

• After processing all the batches, average the weights

• Not only a speed-up, also seems to improve quality

• Allows parallel processing, but requires inter-process communication
Sample Rank

- Generating hope and fear translations is expensive

- Sample good/bad by random walk through alignment space
  - use operations as in Gibbs samples
  - vary one translation option choice
  - vary one reordering decision
  - vary one phrase segmentation decision
  - adopt new translation based on relative score

- Compare current translation against its neighbors

→ apply MIRA update if more costly translation has higher BLEU
**Batch MIRA**

- MIRA requires translation of each sentence on demand
  - repeated decoding needed
  - computationally very expensive

- Batch MIRA
  - n-best list or search graph (lattice)
  - straightforward parallelization
  - does not seem to harm performance
pro
• Reference translation: he does not go home

• N-best list

<table>
<thead>
<tr>
<th>Translation</th>
<th>Feature values</th>
<th>BLEU+1</th>
</tr>
</thead>
<tbody>
<tr>
<td>it is not under house</td>
<td>-32.22</td>
<td>-9.93</td>
</tr>
<tr>
<td></td>
<td>-19.00</td>
<td>-5.08</td>
</tr>
<tr>
<td></td>
<td>-8.22</td>
<td>-5</td>
</tr>
<tr>
<td>he is not under house</td>
<td>-34.50</td>
<td>-7.40</td>
</tr>
<tr>
<td></td>
<td>-16.33</td>
<td>-5.01</td>
</tr>
<tr>
<td></td>
<td>-8.15</td>
<td>-5</td>
</tr>
<tr>
<td>it is not a home</td>
<td>-28.49</td>
<td>-12.74</td>
</tr>
<tr>
<td></td>
<td>-19.29</td>
<td>-3.74</td>
</tr>
<tr>
<td></td>
<td>-8.42</td>
<td>-5</td>
</tr>
<tr>
<td>it is not to go home</td>
<td>-32.53</td>
<td>-10.34</td>
</tr>
<tr>
<td></td>
<td>-20.87</td>
<td>-4.38</td>
</tr>
<tr>
<td></td>
<td>-13.11</td>
<td>-6</td>
</tr>
<tr>
<td>it is not for house</td>
<td>-31.75</td>
<td>-17.25</td>
</tr>
<tr>
<td></td>
<td>-20.43</td>
<td>-4.90</td>
</tr>
<tr>
<td></td>
<td>-6.90</td>
<td>-5</td>
</tr>
<tr>
<td>he is not to go home</td>
<td>-35.79</td>
<td>-10.95</td>
</tr>
<tr>
<td></td>
<td>-18.20</td>
<td>-4.85</td>
</tr>
<tr>
<td></td>
<td>-13.04</td>
<td>-6</td>
</tr>
<tr>
<td>he does not home</td>
<td>-32.64</td>
<td>-11.84</td>
</tr>
<tr>
<td></td>
<td>-16.98</td>
<td>-3.67</td>
</tr>
<tr>
<td></td>
<td>-8.76</td>
<td>-4</td>
</tr>
<tr>
<td>it is not packing</td>
<td>-32.26</td>
<td>-10.63</td>
</tr>
<tr>
<td></td>
<td>-17.65</td>
<td>-5.08</td>
</tr>
<tr>
<td></td>
<td>-9.89</td>
<td>-4</td>
</tr>
<tr>
<td>he is not packing</td>
<td>-34.55</td>
<td>-8.10</td>
</tr>
<tr>
<td></td>
<td>-14.98</td>
<td>-5.01</td>
</tr>
<tr>
<td></td>
<td>-9.82</td>
<td>-4</td>
</tr>
<tr>
<td>he is not for home</td>
<td>-36.70</td>
<td>-13.52</td>
</tr>
<tr>
<td></td>
<td>-17.09</td>
<td>-6.22</td>
</tr>
<tr>
<td></td>
<td>-7.82</td>
<td>-5</td>
</tr>
</tbody>
</table>

• Higher quality translation (BLEU+1) should rank higher
Pick 2 Translations at Random

- Reference translation: he does not go home
- N-best list

<table>
<thead>
<tr>
<th>Translation</th>
<th>Feature values</th>
<th>BLEU+1</th>
</tr>
</thead>
<tbody>
<tr>
<td>it is not under house</td>
<td>-32.22 -9.93 -19.00 -5.08 -8.22 -5</td>
<td>27.3%</td>
</tr>
<tr>
<td>he is not under house</td>
<td>-34.50 -7.40 -16.33 -5.01 -8.15 -5</td>
<td>30.2%</td>
</tr>
<tr>
<td>it is not a home</td>
<td>-28.49 -12.74 -19.29 -3.74 -8.42 -5</td>
<td>30.2%</td>
</tr>
<tr>
<td>it is not to go home</td>
<td>-32.53 -10.34 -20.87 -4.38 -13.11 -6</td>
<td>31.2%</td>
</tr>
<tr>
<td>it is not for house</td>
<td>-31.75 -17.25 -20.43 -4.90 -6.90 -5</td>
<td>27.3%</td>
</tr>
<tr>
<td>he is not to go home</td>
<td>-35.79 -10.95 -18.20 -4.85 -13.04 -6</td>
<td>31.2%</td>
</tr>
<tr>
<td>he does not home</td>
<td>-32.64 -11.84 -16.98 -3.67 -8.76 -4</td>
<td>36.2%</td>
</tr>
<tr>
<td>it is not packing</td>
<td>-32.26 -10.63 -17.65 -5.08 -9.89 -4</td>
<td>21.8%</td>
</tr>
<tr>
<td>he is not packing</td>
<td>-34.55 -8.10 -14.98 -5.01 -9.82 -4</td>
<td>24.2%</td>
</tr>
<tr>
<td>he is not for home</td>
<td>-36.70 -13.52 -17.09 -6.22 -7.82 -5</td>
<td>32.5%</td>
</tr>
</tbody>
</table>

- Higher quality translation (BLEU+1) should rank higher
One is Better than the Other

- Reference translation: he does not go home
- N-best list

<table>
<thead>
<tr>
<th>Translation</th>
<th>Feature values</th>
<th>BLEU+1</th>
</tr>
</thead>
<tbody>
<tr>
<td>it is not under house</td>
<td>-32.22 -9.93 -19.00 -5.08 -8.22 -5</td>
<td>27.3%</td>
</tr>
<tr>
<td>he is not under house</td>
<td>-34.50 -7.40 -16.33 -5.01 -8.15 -5</td>
<td>30.2%</td>
</tr>
<tr>
<td>it is not a home</td>
<td>-28.49 -12.74 -19.29 -3.74 -8.42 -5</td>
<td>30.2%</td>
</tr>
<tr>
<td>it is not to go home</td>
<td>-32.53 -10.34 -20.87 -4.38 -13.11 -6</td>
<td>31.2%</td>
</tr>
<tr>
<td><strong>it is not for house</strong></td>
<td><strong>-31.75 -17.25 -20.43 -4.90 -6.90 -5</strong></td>
<td><strong>27.3%</strong></td>
</tr>
<tr>
<td>he is not to go home</td>
<td>-35.79 -10.95 -18.20 -4.85 -13.04 -6</td>
<td>31.2%</td>
</tr>
<tr>
<td>he does not home</td>
<td>-32.64 -11.84 -16.98 -3.67 -8.76 -4</td>
<td>36.2%</td>
</tr>
<tr>
<td>it is not packing</td>
<td>-32.26 -10.63 -17.65 -5.08 -9.89 -4</td>
<td>21.8%</td>
</tr>
<tr>
<td>he is not packing</td>
<td>-34.55 -8.10 -14.98 -5.01 -9.82 -4</td>
<td>24.2%</td>
</tr>
<tr>
<td><strong>he is not for home</strong></td>
<td><strong>-36.70 -13.52 -17.09 -6.22 -7.82 -5</strong></td>
<td><strong>32.5%</strong></td>
</tr>
</tbody>
</table>

- Higher quality translation (BLEU+1) should rank higher
Learn from the Pairwise Sample

- **Pairwise sample**
  - \( \text{bad} = (-31.75, -17.25, -20.43, -4.90, -6.90, -5) \)
  - \( \text{good} = (-36.70, -13.52, -17.09, -6.22, -7.82, -5) \)

- **Learn a classifier**
  - \( \text{bad} \rightarrow \text{good} \rightarrow \frown \)
  - \( \text{good} \rightarrow \text{bad} \rightarrow \smile \)

- **Use off the shelf maximum entropy classifier to learn weights for each feature**
e.g., MegaM (http://www.umiacs.umd.edu/~hal/megam/)
Sampling

• Collect samples for each sentence pair in tuning set

• For each sentence, sample 1000-best list for 50 pairwise samples

• Reject samples if difference in BLEU+1 score is too small (≤ 0.05)

• Iterate process
  1. set default weights
  2. generate n-best list
  3. build classifier
  4. adopt classifier weights
  5. go to 2, unless converged
leave one out
Leave One Out Training

• Train initial baseline model

• Force translate the training data:
  require decoder to match the reference translation

• Collect statistics over translation rules used

• Leave one out:
  do not use translation rules originally collected from current sentence pair

• Related to jackknife
  – 90% of training data used for rule collection
  – 10% to validate rules
  – rotate
Translate Almost All Sentences

- Relaxed leave-one-out
  - allow rules originally collected from current sentence pair
  - very costly $\rightarrow$ only used, if everything else fails

- Allow single word translations (avoid OOV)

- Larger distortion limit

- Word deletion and insertion (very costly)
Model Re-Estimation

- Generate 100-best list
- Collect fractional counts from derivations

⇒ Much smaller model
⇒ Sometimes better model
max-violation perceptron
and
forced decoding
Early work on stochastic gradient descent over full training corpus unsuccessful

One reason: Search errors break theoretical properties of convergence

Are unreachable reference translations a problem?
  - yes: ignoring them leaves out large amounts of training data
  - no: data selection, non-literal translations are lower quality

Idea: update when partial reference derivation falls out of beam
Reachability

Reachability by distortion limit and sentence length
Chinese–English NIST [Yu et al., 2013]
Recall: Decoding

- Extend partial translations (=hypotheses) by adding translation options
- Organize hypotheses in stacks, prune out bad ones
Matching the Reference

- Some hypotheses match the reference translation
  
  he does not go home
Early Updating

- At some point the best reference derivation may fall outside the beam
- Early updating
  - perceptron update between partial derivations
  - best derivation vs. best reference derivation outside beam
- Note: a reference derivation may skip a bin (multi-word phrase translation)
  - only stop when no hope that reference derivation will be in a future stack
Max Violation

- Complete search process
- Keep best reference derivations
- Maximum violation update
  - find stack where maximal model score difference between
    * best derivation
    * best reference derivation
  - update between those two derivations
Max Violation

- Shown to be successful [Yu et al., 2013]
  - optimization over full training corpus
  - over 20 million features
  - relatively small data conditions (5-9 millions words)
  - gain: +2 BLEU points

- Features
  - rule id
  - word edge features (first and last word of phrase), defined over words, word clusters, or POS tags
  - combinations of word edge features
  - non-local features: ids of consecutive rules, rule id + last two English words

- Address overfitting: leave-one-out or singleton pruning
Summary

- MERT [Och et al. 2003]
- PRO [Hopkins/May 2011]
- MIRA [Chiang 2007]
- SampleRank [Haddow et al. 2011]
- MaxViolation [Yu et al., 2013]
- Leave One Out [Wuebker et al., 2012]

- search space
- iterative n-best
- n-best
- aligned corpus
- a handful
- thousands
- millions

Rule scores