Statistical NLP
Fall 2016

Lecture 6:
Phrase-Based Translation

Slav Petrov – Google

Slides loosely based on slides from Philipp Koehn and Chris Dyer

Some Results

- [Och and Ney 03]

<table>
<thead>
<tr>
<th>Model</th>
<th>Training scheme</th>
<th>0.5K</th>
<th>8K</th>
<th>128K</th>
<th>1.47M</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dice</td>
<td></td>
<td>50.9</td>
<td>43.4</td>
<td>39.6</td>
<td>38.9</td>
</tr>
<tr>
<td>Dice+C</td>
<td></td>
<td>46.3</td>
<td>37.6</td>
<td>35.0</td>
<td>34.0</td>
</tr>
<tr>
<td>Model 1</td>
<td>$i^5$</td>
<td>40.6</td>
<td>33.6</td>
<td>28.6</td>
<td>25.9</td>
</tr>
<tr>
<td>Model 2</td>
<td>$i^22^5$</td>
<td>46.7</td>
<td>29.3</td>
<td>22.0</td>
<td>19.5</td>
</tr>
<tr>
<td>HMM</td>
<td>$i^5H^5$</td>
<td>26.3</td>
<td>23.3</td>
<td>15.0</td>
<td>10.8</td>
</tr>
<tr>
<td>Model 3</td>
<td>$i^22^53^3$</td>
<td>43.6</td>
<td>27.5</td>
<td>20.5</td>
<td>18.0</td>
</tr>
<tr>
<td></td>
<td>$i^1H^33^3$</td>
<td>27.5</td>
<td>22.5</td>
<td>16.6</td>
<td>13.2</td>
</tr>
<tr>
<td>Model 4</td>
<td>$i^22^33^4$</td>
<td>41.7</td>
<td>25.1</td>
<td>17.3</td>
<td>14.1</td>
</tr>
<tr>
<td></td>
<td>$i^3H^33^4$</td>
<td>26.1</td>
<td>20.2</td>
<td>13.1</td>
<td>9.4</td>
</tr>
<tr>
<td></td>
<td>$i^4H^3$</td>
<td>26.3</td>
<td>21.8</td>
<td>13.3</td>
<td>9.3</td>
</tr>
<tr>
<td>Model 5</td>
<td>$i^3H^34^3$</td>
<td>26.5</td>
<td>21.5</td>
<td>13.7</td>
<td>9.6</td>
</tr>
<tr>
<td></td>
<td>$i^3H^33^5$</td>
<td>26.5</td>
<td>20.4</td>
<td>13.4</td>
<td>9.4</td>
</tr>
<tr>
<td>Model 6</td>
<td>$i^3H^34^6$</td>
<td>26.0</td>
<td>21.6</td>
<td>12.8</td>
<td>8.8</td>
</tr>
<tr>
<td></td>
<td>$i^3H^33^44^6$</td>
<td>25.9</td>
<td>20.3</td>
<td>12.5</td>
<td>8.7</td>
</tr>
</tbody>
</table>

Bidirectional Alignment

Consistent Phrases

- All words of the phrase have to align to each other (or to nothing)
Extracting Phrases

(Maria, Mary), (no, did not), (slap, daba una bofetada), (a la, the), (bruja, witch), (verde, green), (Maria no, Mary did not), (no daba una bofetada, did not slap), (daba una bofetada a la, slap the), (bruja verde, green witch), (Maria no daba una bofetada, Mary did not slap), (no daba una bofetada a la, did not slap the), (a la bruja verde, the green witch), (Maria no daba una bofetada a la, Mary did not slap the), (daba una bofetada a la bruja verde, slap the green witch), (no daba una bofetada a la bruja verde, did not slap the green witch), (Maria no daba una bofetada a la bruja verde, Mary did not slap the green witch)

Phrase Weights

How the MT community estimates $P(f|\bar{e})$

Parallel training sentences provide phrase pair counts.

Gracias, lo haré de muy buen grado.

Io haré $\Leftrightarrow$ I shall do so
gladly.

44 times in the corpus

All phrase pairs are counted, and counts are normalized.

Gracias, lo haré de muy buen grado.

Thank you, I shall do so gladly.

$P(f|\bar{e}) = \frac{\text{count}(f, \bar{e})}{\text{count}(\bar{e})}$

Phrase Scoring

- Learning weights has been tried, several times:
  - [Marcu and Wong, 02]
  - [DeNero et al, 06]
  - ... and others

- Seems not to work well, for a variety of partially understood reasons

- Main issue: big chunks get all the weight, obvious priors don’t help
  - Though, [DeNero et al 08]

Alignment Heuristics

<table>
<thead>
<tr>
<th>Learning weights has been tried, several times:</th>
</tr>
</thead>
<tbody>
<tr>
<td>[Marcu and Wong, 02]</td>
</tr>
<tr>
<td>[DeNero et al, 06]</td>
</tr>
<tr>
<td>... and others</td>
</tr>
</tbody>
</table>

- Seems not to work well, for a variety of partially understood reasons

- Main issue: big chunks get all the weight, obvious priors don’t help
  - Though, [DeNero et al 08]
Phrase Size

- Phrases do help
  - But they don’t need to be long
  - Why should this be?

Lexical Weighting

\[
\phi(f_i | e_i) = \frac{\text{count}(f_i, e_i)}{\text{count}(e_i)} \cdot p_w(f_i | e_i)
\]

\[
p_w(f_i | e_i) = p_w(f_1, f_2, f_3 | e_1, e_2, e_3, a) = w(f_1 | e_1) \times \frac{1}{2}(w(f_2 | e_2) + w(f_3 | e_3)) \times w(f_3 | \text{NULL})
\]

Sources of Alignments

<table>
<thead>
<tr>
<th>Method</th>
<th>Training corpus size</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>10k</td>
</tr>
<tr>
<td>AP</td>
<td>84k</td>
</tr>
<tr>
<td>Joint</td>
<td>125k</td>
</tr>
<tr>
<td>Syn</td>
<td>19k</td>
</tr>
</tbody>
</table>

Phrase Table Example

- Phrase translations for ‘der Vorschlag’:

| English          | \(\phi(e | f)\) | English          | \(\phi(e | f)\) |
|------------------|----------------|------------------|----------------|
| the proposal     | 0.6227         | the suggestions  | 0.0114         |
| 's proposal      | 0.1068         | the proposed     | 0.0114         |
| a proposal       | 0.0341         | the motion       | 0.0091         |
| the idea         | 0.0250         | the idea of      | 0.0091         |
| this proposal    | 0.0227         | the proposal ,   | 0.0068         |
| proposal         | 0.0205         | its proposal     | 0.0068         |
| of the proposal  | 0.0159         | it               | 0.0068         |
| the proposals    | 0.0159         | ...              | ...            |

- Lexical variation, morphology, function words
Decoding

- In these word-to-word models
  - Finding best alignments is easy
  - Finding translations is hard (why?)

Phrase-Based Decoding

Bag Generation as a TSP

- Imagine bag generation with a bigram LM
  - Words are nodes
  - Edge weights are $P(w|w')$
  - Valid sentences are Hamiltonian paths
- Not the best news for word-based MT!

IBM Decoding as a TSP
Decoding, Anyway

- Simplest possible decoder:
  - Enumerate sentences, score each with TM and LM

- Greedy decoding:
  - Assign each French word its most likely English translation
  - Operators:
    - Change a translation
    - Insert a word into the English (zero-fertile French)
    - Remove a word from the English (null-generated French)
    - Swap two adjacent English words
  - Do hill-climbing (or annealing)

Phrase-Based Systems

<table>
<thead>
<tr>
<th>Morgen</th>
<th>fliege</th>
<th>ich</th>
<th>nach Kanada</th>
<th>zur Konferenz</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tomorrow</td>
<td>I will fly</td>
<td>to the conference</td>
<td>in Canada</td>
<td></td>
</tr>
</tbody>
</table>

Phrase table (translation model)

Sentence-aligned corpus

Word alignments

Greedy Decoding

Phrase-Based Systems

Phrase table (translation model)

The Pharaoh “Model”

\[
P(c|g) = P(\{\tilde{g}_i}\|g) \prod_i \phi(c_i|\tilde{g}_i) d(a_i - b_{i-1})
\]

Segmentation

Translation

Distortion

[Coehn et al, 2003]
The Pharaoh “Model”

\[ P(f|e) = P(\{\vec{e}_i\}|e) \prod_{i=1}^{T} \phi(\vec{f}_i|\vec{e}_i) d(a_i - b_{i-1}) \]

\[ \frac{1}{K} \cdot \text{count}(\vec{f}_i, \vec{e}_i) \quad e^{a_i-b_{i-1}} \]

Where do we get these counts?

The Pharaoh Decoder

- Probabilities at each step include LM and TM

Decoding

Hypothesis Lattices

Drop weaker path when:
- last two English words match (matters for language model)
- foreign word coverage vectors match (affects future path)
Pruning

Problem: easy partial analyses are cheaper
Solution: Use hypothesis stacks per foreign subset

Stack Decoding

Partial Translations are Hard to Compare

Estimate Future Cost

Cost for each translation option:
- translation model: cost known
- language model: output words known, but not context -> estimate without context
- reordering model: unknown -> ignore
Combining Score and Future Cost

- Future costs make hypotheses comparable

Future costs:

\[
\begin{align*}
\text{the tourism initiative} & \quad \text{die touristische initiative} \\
\text{tm:} & -1.21, \text{lm:} -4.67 \\
\text{d:} & 0, \text{all:} -5.88 \\
\text{the first time} & \quad \text{das erste mal} \\
\text{tm:} & -0.56, \text{lm:} -2.81 \\
\text{d:} & -0.74, \text{all:} -4.11 \\
\text{this for } \ldots \text{ time} & \quad \text{für diese zeit} \\
\text{tm:} & -0.82, \text{lm:} -2.98 \\
\text{d:} & -1.06, \text{all:} -4.86 \\
\text{this for } \ldots \text{ time} & \quad \text{für diese zeit} \\
\text{tm:} & -0.82, \text{lm:} -2.98 \\
\text{d:} & -1.06, \text{all:} -4.86 \\
\end{align*}
\]

Phrase-Based Translation Summary

Input: Io haré rápidamente.
Translations: I’l do it quickly.

Objective: \[ \arg \max_e [P(f|e) \cdot P(e)] \]

Minimum Error Rate Training

- Non-convex, non-differentiable objective: [Och ’03]
  - Generate n-best list
  - Line search 1 dir. at a time
  - Use random restarts

Each hypothesis in n-best list contributes a line:

\[ \text{score}(e, f) = \sum_{i=1}^{m} \lambda_i \cdot h_i(e, f) \]

“Tuning”

- More generally, define feature functions \( h \):
  \[
p(e|f) \propto \exp \sum_{k=1}^{m} \lambda_k \cdot h_k(e, f)
\]

- \( P(e) \) and \( P(f|e) \) are just two of many possible feature functions:
  - \( P(e|f) \), word counts, phrase counts, etc.

- Need to learn how to set weights \( \lambda \)
  - What function to maximize?
  - How to maximize? Is it differentiable?
Stack Decoding

- Stack decoding:
  - Beam search
  - Usually A* estimates for completion cost
  - One stack per candidate sentence length
- Other methods:
  - Dynamic programming decoders possible if we make assumptions about the set of allowable permutations

<table>
<thead>
<tr>
<th>sent length</th>
<th>decoder type</th>
<th>time (sec/seg)</th>
<th>search errors</th>
<th>translation errors (semantic and/or syntaxes)</th>
<th>NE</th>
<th>PME</th>
<th>DSE</th>
<th>FSE</th>
<th>HSE</th>
<th>CE</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>BPR</td>
<td>47.50</td>
<td>0</td>
<td>55</td>
<td>44</td>
<td>57</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>stack greedy</td>
<td>0.07</td>
<td>18</td>
<td>60</td>
<td>38</td>
<td>45</td>
<td>5</td>
<td>2</td>
<td>1</td>
<td>10</td>
</tr>
<tr>
<td>8</td>
<td>BPR</td>
<td>499.00</td>
<td>0</td>
<td>56</td>
<td>27</td>
<td>74</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>8</td>
<td>stack greedy</td>
<td>5.67</td>
<td>20</td>
<td>75</td>
<td>24</td>
<td>57</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>15</td>
</tr>
<tr>
<td>8</td>
<td>greedy</td>
<td>2.66</td>
<td>43</td>
<td>75</td>
<td>20</td>
<td>38</td>
<td>4</td>
<td>5</td>
<td>1</td>
<td>33</td>
</tr>
</tbody>
</table>

Exact Phrase-Based Decoding

- Exact decoding is NP-hard
- Lagrangian Relaxation:
  - Relax problem: translate \( n \) words (any \( n \) words)
  - Add constraints to enforce correct solution

<table>
<thead>
<tr>
<th>Beam size</th>
<th>time (sec.)</th>
<th>Fails</th>
<th># search errors</th>
<th>percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>0.3355</td>
<td>650/1,818</td>
<td>214/1,168</td>
<td>18.32%</td>
</tr>
<tr>
<td>200</td>
<td>0.4477</td>
<td>531/1,818</td>
<td>207/1,287</td>
<td>16.08%</td>
</tr>
<tr>
<td>1,000</td>
<td>4.1055</td>
<td>342/1,818</td>
<td>115/1,476</td>
<td>7.79%</td>
</tr>
<tr>
<td>10,000</td>
<td>42.9423</td>
<td>169/1,818</td>
<td>68/1,649</td>
<td>4.12%</td>
</tr>
</tbody>
</table>

Machine Translation Preordering

- English has Subject-Verb-Object word order, while Japanese has Subject-Object-Verb order
- Use hand-written or automatically learned rules to change word order prior to translation [Collins et al. '05]
- Dependency-based reordering for English-Japanese [Xu et al. '09]

- They solved the problem with statistics
- Sie kann Pasta jeden Tag essen.
- Sie glaubt dass sie Pasta jeden Tag essen kann.
The black cat climbed to the tree top.

---

**Classifier Preordering 1-Step**

- Predict the target word order by treating each permutation as a label in a multi-class classifier. Traverse the parse tree, reordering each family (head and children) and recursing.

- Problem: lots of permutations!

---

**Classifier Preordering 2-Step**

- Decompose the search space: first determine the position of every child relative to the head (pivot) and then order the children before and after the head. Think QuickSort without recursion.

---

**Classifier Reordering Results**

---

**Dutch Parser in Machine Translation**

- No preordering (Lerner and Petrov, 2013)
- Arc-eager preordering (L&P, 2013)
- Arc-eager preordering (updated baseline)
- Two-Registers preordering (updated baseline)