Lecture 12: 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>1^5</td>
<td>40.6</td>
<td>33.6</td>
<td>28.6</td>
<td>25.9</td>
</tr>
<tr>
<td>Model 2</td>
<td>1^25</td>
<td>46.7</td>
<td>29.3</td>
<td>22.0</td>
<td>19.5</td>
</tr>
<tr>
<td>HMM</td>
<td>1^H5</td>
<td>26.3</td>
<td>23.3</td>
<td>15.0</td>
<td>10.8</td>
</tr>
<tr>
<td>Model 3</td>
<td>1^2^3^3^3^3</td>
<td>43.6</td>
<td>27.5</td>
<td>20.5</td>
<td>18.0</td>
</tr>
<tr>
<td></td>
<td>1^H^3^3^3^3</td>
<td>27.5</td>
<td>22.5</td>
<td>16.6</td>
<td>13.2</td>
</tr>
<tr>
<td>Model 4</td>
<td>1^2^3^4^4^4^4</td>
<td>41.7</td>
<td>25.1</td>
<td>17.3</td>
<td>14.1</td>
</tr>
<tr>
<td></td>
<td>1^H^3^4^4^4^4</td>
<td>26.1</td>
<td>20.2</td>
<td>13.1</td>
<td>9.4</td>
</tr>
<tr>
<td></td>
<td>1^H^3^4^3^3^3</td>
<td>26.3</td>
<td>21.8</td>
<td>13.3</td>
<td>9.3</td>
</tr>
<tr>
<td>Model 5</td>
<td>1^H^4^5^5^5^5</td>
<td>26.5</td>
<td>21.5</td>
<td>13.7</td>
<td>9.6</td>
</tr>
<tr>
<td></td>
<td>1^H^3^4^5^5^5</td>
<td>26.5</td>
<td>20.4</td>
<td>13.4</td>
<td>9.4</td>
</tr>
<tr>
<td>Model 6</td>
<td>1^H^4^6^6^6^6</td>
<td>26.0</td>
<td>21.6</td>
<td>12.8</td>
<td>8.8</td>
</tr>
<tr>
<td></td>
<td>1^H^3^4^6^6^6</td>
<td>25.9</td>
<td>20.3</td>
<td>12.5</td>
<td>8.7</td>
</tr>
</tbody>
</table>

---

**Bidirectional Alignment**

- English to Spanish
- Spanish to English

**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),
(vero, 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 $P(f|e)$

Parallel training sentences provide phrase pair counts.

Gracias, lo haré de muy buen grado.
Thank you, I shall do so gladly.

Io haré ⇔ I shall do so
44 times in the corpus

All phrase pairs are counted, and counts are normalized.

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

Phrase Scoring

$\phi_{new}(\tilde{e}_j|\tilde{f}_i) = \frac{c(\tilde{f}_i, \tilde{e}_j)}{c(\tilde{f}_i)}$

- 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

- 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]
**Phrase Size**

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

**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>

**Lexical Weighting**

\[
\phi(\bar{f}_i|\bar{e}_i) = \frac{\text{count}(\bar{f}_i, \bar{e}_i)}{\text{count}(\bar{e}_i)} \cdot p_w(\bar{f}_i|\bar{e}_i)
\]

**Phrase Table Example**

- Phrase translations for ‘der Vorschlag’:

| English       | \(\phi(\bar{e}|f)\) | English       | \(\phi(\bar{e}|f)\) |
|---------------|----------------------|---------------|----------------------|
| 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?)

it is not clear.

CE NE EST PAS CLAIR.

Phrases-Based Decoding

<table>
<thead>
<tr>
<th>The sentence</th>
<th>Encoding</th>
<th>Decoding</th>
</tr>
</thead>
<tbody>
<tr>
<td>7 people</td>
<td></td>
<td></td>
</tr>
<tr>
<td>travelling</td>
<td></td>
<td></td>
</tr>
<tr>
<td>by plane</td>
<td></td>
<td></td>
</tr>
<tr>
<td>and visa</td>
<td></td>
<td></td>
</tr>
<tr>
<td>the reason</td>
<td></td>
<td></td>
</tr>
<tr>
<td>the case</td>
<td></td>
<td></td>
</tr>
<tr>
<td>this flight</td>
<td></td>
<td></td>
</tr>
<tr>
<td>this plane</td>
<td></td>
<td></td>
</tr>
<tr>
<td>the French</td>
<td></td>
<td></td>
</tr>
<tr>
<td>and the crew</td>
<td></td>
<td></td>
</tr>
<tr>
<td>of the crew</td>
<td></td>
<td></td>
</tr>
<tr>
<td>of the plane</td>
<td></td>
<td></td>
</tr>
<tr>
<td>from France</td>
<td></td>
<td></td>
</tr>
<tr>
<td>and Russia</td>
<td></td>
<td></td>
</tr>
<tr>
<td>to Russia</td>
<td></td>
<td></td>
</tr>
<tr>
<td>the astronaut</td>
<td></td>
<td></td>
</tr>
<tr>
<td>the astronaut</td>
<td></td>
<td></td>
</tr>
<tr>
<td>the astronaut</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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

- Sentence-aligned corpus
- Word alignments
- Phrase table (translation model)

Greedy Decoding

- Null well heard, it talks a great victory.
- 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)

The Pharaoh “Model”

- Sentence-aligned corpus
- Word alignments
- Phrase table (translation model)

Phrase-Based Systems

[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

- Sentence-aligned corpus
- Word alignments
- Phrase table (translation model)

Greedy Decoding

- Null well heard, it talks a great victory.
- 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)

The Pharaoh “Model”

- Sentence-aligned corpus
- Word alignments
- Phrase table (translation model)
The Pharaoh “Model”

\[
P(f|e) = P(\{\bar{c}_i\}|e) \prod_i \phi(\bar{f}_i|\bar{c}_i) d(a_i - b_i - 1) \frac{1}{K} \frac{\text{count}(\bar{f}_i, \bar{c}_i)}{\text{count}(\bar{c}_i)} \alpha^{a_i - b_i - 1} \]

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

Phrase-Based Translation Summary

The decoder...

Input: lo haré rápidamente.
Translations: I'll do it quickly.

More generally, define feature functions \( h: \)

\[ P(e|f), \text{word counts, phrase counts, etc.} \]

Need to learn how to set weights \( \lambda \)

- What function to maximize?
- How to maximize? Is it differentiable?

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:

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

\[
\text{score}(e,f) = \lambda \cdot h(e,f) + \sum_{i \neq j} \lambda_i \cdot h_i(e,f)
\]

\[
\text{score}(e,f) = A\lambda + B
\]
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/nt)</th>
<th>errors</th>
<th>translation errors (semantic and/or syntax)</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>BP</td>
<td>47.56</td>
<td>0</td>
<td>57</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</td>
<td>0.79</td>
<td>8</td>
<td>58</td>
<td>43</td>
<td>53</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>6</td>
<td>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>BP</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</td>
<td>5.07</td>
<td>20</td>
<td>75</td>
<td>24</td>
<td>57</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>15</td>
</tr>
<tr>
<td>8</td>
<td>greedy</td>
<td>4.66</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># searches</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
- Dependency-based reordering for English-Japanese

```
They solved the problem with statistics
PRON VERB DET NOUN ADP NOUN
```

Machine Translation Reordering

- Source-side reordering for machine translation
- Use source-side syntax to guide reordering decisions
- Dependency-based reordering for English-Japanese

```
They can pasta every day eat .
Sie kann Pasta jeden Tag essen .
```

```
She can eat pasta every day .
She glaubt dass sie Pasta jeden Tag essen kann .
```
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

---

GradBoost

Features:
- Head wording
- Children and position
- Words/tags between children

Pairs of the above.

GradBoost

**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!**

**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.**