Phrase-Based MT
Translational Equivalence

Ma hat die Prüfung bestanden, jedoch nur knapp

Ma insisted on the test, but just barely.
Ma passed the test, but just barely.

How do lexical translation models deal with contextual information?
Ma insisted on the test, but just barely.
Ma passed the test, but just barely.

<table>
<thead>
<tr>
<th>F</th>
<th>E</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>bestanden</td>
<td>insisted</td>
<td>-1.18</td>
</tr>
<tr>
<td>were</td>
<td></td>
<td>-1.18</td>
</tr>
<tr>
<td>existed</td>
<td></td>
<td>-1.36</td>
</tr>
<tr>
<td>was</td>
<td></td>
<td>-1.39</td>
</tr>
<tr>
<td>been</td>
<td></td>
<td>-1.43</td>
</tr>
<tr>
<td>passed</td>
<td></td>
<td>-1.52</td>
</tr>
<tr>
<td>consist</td>
<td></td>
<td>-1.87</td>
</tr>
</tbody>
</table>
Ma hat die Prüfung **bestanden**, jedoch nur knapp

Ma **insisted on** the test, but just barely.
Ma **passed** the test, but just barely.

**Lexical Translation**

**What is wrong with this?**

**How can we improve this?**
Translation model

• With a **latent variable**, we introduce a decomposition into **phrases** which translate **independently**:

\[
p(f, a \mid e) = p(a) \prod_{\langle \bar{e}, \bar{f} \rangle \in a} p(f \mid \bar{e})
\]

\[
f = \text{Morgen fliege ich nach Baltimore zur Konferenz}
\]

\[
e = \text{Tomorrow I will fly to the Konferenz in Baltimore}
\]
Translation Model

• What are the atomic units
  • Lexical translation: **words**
  • Phrase-based translation: **phrases**

• Benefits
  • many-to-many translation
  • use of local context in translation

• Downsides
  • Where do phrases comes from?

• Standard model used by Google, Microsoft ...
Translation model

- With a latent variable, we introduce a decomposition into phrases which translate independently:

\[
p(f, a \mid e) = p(a) \prod_{\langle \bar{e}, \bar{f} \rangle \in a} p(\bar{f} \mid \bar{e})
\]

\[
p(f \mid e) = \sum_a p(f, a \mid e)
\]

\[
p(a \mid e) = \sum_{f \mid e} p(f, a \mid e)
\]

\[
p(f) = \sum_{a \mid e} p(f, a \mid e)
\]

\[
p(a) = \sum_{f \mid e} p(f, a \mid e)
\]

\[
p(e) = \sum_{f \mid e} p(f, a \mid e)
\]

\[
p(f, a, e) = p(f \mid e) p(a \mid e)
\]

\[
p(f) p(a) = \sum_{e} p(f, a, e)
\]

\[
p(f) = \sum_{a \mid e} p(f, a, e)
\]

\[
p(a) = \sum_{f \mid e} p(f, a, e)
\]

\[
p(e) = \sum_{f \mid e} p(f, a, e)
\]

\[
p(f \mid e) = \sum_{a \mid e} p(f, a, e)
\]

\[
p(a \mid e) = \sum_{f \mid e} p(f, a, e)
\]

\[
p(e) = \sum_{f \mid e} p(f, a, e)
\]

\[
p(f, a, e) = p(f \mid e) p(a \mid e)
\]

\[
p(f) p(a) = \sum_{e} p(f, a, e)
\]

\[
p(f) = \sum_{a \mid e} p(f, a, e)
\]

\[
p(a) = \sum_{f \mid e} p(f, a, e)
\]

\[
p(e) = \sum_{f \mid e} p(f, a, e)
\]

\[
p(f) p(a) = \sum_{e} p(f, a, e)
\]

\[
p(f) = \sum_{a \mid e} p(f, a, e)
\]

\[
p(a) = \sum_{f \mid e} p(f, a, e)
\]

\[
p(e) = \sum_{f \mid e} p(f, a, e)
\]

\[
p(f \mid e) = \sum_{a \mid e} p(f, a, e)
\]

\[
p(a \mid e) = \sum_{f \mid e} p(f, a, e)
\]

\[
p(e) = \sum_{f \mid e} p(f, a, e)
\]

\[
p(f, a, e) = p(f \mid e) p(a \mid e)
\]

\[
p(f) p(a) = \sum_{e} p(f, a, e)
\]

\[
p(f) = \sum_{a \mid e} p(f, a, e)
\]

\[
p(a) = \sum_{f \mid e} p(f, a, e)
\]

\[
p(e) = \sum_{f \mid e} p(f, a, e)
\]

\[
p(f \mid e) = \sum_{a \mid e} p(f, a, e)
\]

\[
p(a \mid e) = \sum_{f \mid e} p(f, a, e)
\]

\[
p(e) = \sum_{f \mid e} p(f, a, e)
\]

\[
p(f, a, e) = p(f \mid e) p(a \mid e)
\]

\[
p(f) p(a) = \sum_{e} p(f, a, e)
\]

\[
p(f) = \sum_{a \mid e} p(f, a, e)
\]

\[
p(a) = \sum_{f \mid e} p(f, a, e)
\]

\[
p(e) = \sum_{f \mid e} p(f, a, e)
\]

\[
p(f \mid e) = \sum_{a \mid e} p(f, a, e)
\]

\[
p(a \mid e) = \sum_{f \mid e} p(f, a, e)
\]

\[
p(e) = \sum_{f \mid e} p(f, a, e)
\]

\[
p(f, a, e) = p(f \mid e) p(a \mid e)
\]

\[
p(f) p(a) = \sum_{e} p(f, a, e)
\]

\[
p(f) = \sum_{a \mid e} p(f, a, e)
\]

\[
p(a) = \sum_{f \mid e} p(f, a, e)
\]

\[
p(e) = \sum_{f \mid e} p(f, a, e)
\]

\[
p(f \mid e) = \sum_{a \mid e} p(f, a, e)
\]

\[
p(a \mid e) = \sum_{f \mid e} p(f, a, e)
\]

\[
p(e) = \sum_{f \mid e} p(f, a, e)
\]

\[
p(f, a, e) = p(f \mid e) p(a \mid e)
\]

\[
p(f) p(a) = \sum_{e} p(f, a, e)
\]

\[
p(f) = \sum_{a \mid e} p(f, a, e)
\]

\[
p(a) = \sum_{f \mid e} p(f, a, e)
\]

\[
p(e) = \sum_{f \mid e} p(f, a, e)
\]

\[
p(f \mid e) = \sum_{a \mid e} p(f, a, e)
\]

\[
p(a \mid e) = \sum_{f \mid e} p(f, a, e)
\]

\[
p(e) = \sum_{f \mid e} p(f, a, e)
\]

\[
p(f, a, e) = p(f \mid e) p(a \mid e)
\]

\[
p(f) p(a) = \sum_{e} p(f, a, e)
\]

\[
p(f) = \sum_{a \mid e} p(f, a, e)
\]

\[
p(a) = \sum_{f \mid e} p(f, a, e)
\]

\[
p(e) = \sum_{f \mid e} p(f, a, e)
\]

\[
p(f \mid e) = \sum_{a \mid e} p(f, a, e)
\]

\[
p(a \mid e) = \sum_{f \mid e} p(f, a, e)
\]

\[
p(e) = \sum_{f \mid e} p(f, a, e)
\]

\[
p(f, a, e) = p(f \mid e) p(a \mid e)
\]

\[
p(f) p(a) = \sum_{e} p(f, a, e)
\]

\[
p(f) = \sum_{a \mid e} p(f, a, e)
\]

\[
p(a) = \sum_{f \mid e} p(f, a, e)
\]

\[
p(e) = \sum_{f \mid e} p(f, a, e)
\]

\[
p(f \mid e) = \sum_{a \mid e} p(f, a, e)
\]

\[
p(a \mid e) = \sum_{f \mid e} p(f, a, e)
\]

\[
p(e) = \sum_{f \mid e} p(f, a, e)
\]

\[
p(f, a, e) = p(f \mid e) p(a \mid e)
\]

\[
p(f) p(a) = \sum_{e} p(f, a, e)
\]

\[
p(f) = \sum_{a \mid e} p(f, a, e)
\]

\[
p(a) = \sum_{f \mid e} p(f, a, e)
Translation model

• With a **latent variable**, we introduce a decomposition into **phrases** which translate independently:

\[
p(f, a \mid e) = p(a) \prod_{(\bar{e}, \bar{f}) \in a} p(\bar{f} \mid \bar{e})
\]

\[
p(f \mid e) = p(a) \prod_{(\bar{e}, \bar{f}) \in a} p(\bar{f} \mid \bar{e})
\]

We can then marginalize to get \(p(f \mid e)\):

\[
p(Morgen \mid Tomorrow) = p(f \mid e)
\]

\[
\text{(Morgen) \quad fliege ich nach Baltimore zur Konferenz}
\]

\[
\text{(Tomorrow) \quad I will fly to the Konferenz in Baltimore}
\]
Translation model

- With a **latent variable**, we introduce a decomposition into **phrases** which translate independently:

\[
p(f, a \mid e) = p(a) \prod_{\langle \bar{e}, \bar{f} \rangle \in a} p(\bar{f} \mid \bar{e})
\]

\[
p(f \mid e) = \sum_a p(a) \prod_{\langle \bar{e}, \bar{f} \rangle \in a} p(\bar{f} \mid \bar{e})
\]

We can then marginalize to get \(p(f \mid e)\):

\[
p(Morgen \mid Tomorrow) \times p(fliege \mid will fly)
\]
Translation model

- With a **latent variable**, we introduce a decomposition into **phrases** which translate independently:

\[
p(f, a \mid e) = p(a) \prod_{\langle \bar{e}, \bar{f} \rangle \in a} p(\bar{f} \mid \bar{e})
\]

\[
p(M) = p(\text{Morgen}) \times p(\text{fliege} \mid \text{will fly}) \times p(\text{ich} \mid I)
\]

\[
p(\text{Morgen} \mid \text{Tomorrow}) \times p(\text{fliege} \mid \text{will fly}) \times p(\text{ich} \mid I)
\]
Translation model

- With a **latent variable**, we introduce a decomposition into **phrases** which translate independently:

\[
p(f, a \mid e) = p(a) \prod_{(\bar{e}, \bar{f}) \in a} p(\bar{f} \mid \bar{e})
\]

We can then marginalize to get \( p(f \mid e) \):

\[
p(f \mid e) = \sum_a p(f, a \mid e)
\]

\[
p(Morgen \mid Tomorrow) \times p(\text{fliege} \mid \text{will fly}) \times p(\text{ich} \mid \text{I}) \times \ldots
\]
Translation model

- With a latent variable, we introduce a decomposition into phrases which translate independently:

\[
p(f, a \mid e) = p(a) \prod_{\langle \bar{e}, \bar{f} \rangle \in a} p(f \mid \bar{e})
\]

Marginalize to get \( p(f \mid e) \):

\[
p(f \mid e) = \sum_{a \in A} p(a) \prod_{\langle \bar{e}, \bar{f} \rangle \in a} p(f \mid \bar{e})
\]
Phrases

• Contiguous strings of words
• Phrases are not necessarily syntactic constituents
• Usually have maximum limits
• Phrases subsume words (words are phrases)
Linguistic Phrases

• Model is not limited to linguistic phrases (NPs, VPs, PPs, CPs...)

• Non-constituent phrases are useful

\textit{es gibt} | \textit{there is} | \textit{there are}

• Is a “good” phrase more likely to be 
  \( [P \text{ NP}] \) or \( [\text{governor} \ P] \)

Why? How would you figure this out?
# Phrase Tables

<table>
<thead>
<tr>
<th>$\bar{f}$</th>
<th>$\bar{e}$</th>
<th>$p(\bar{f} \mid \bar{e})$</th>
</tr>
</thead>
<tbody>
<tr>
<td>das Thema</td>
<td>the issue</td>
<td>0.41</td>
</tr>
<tr>
<td></td>
<td>the point</td>
<td>0.72</td>
</tr>
<tr>
<td></td>
<td>the subject</td>
<td>0.47</td>
</tr>
<tr>
<td></td>
<td>the thema</td>
<td>0.99</td>
</tr>
<tr>
<td>es gibt</td>
<td>there is</td>
<td>0.96</td>
</tr>
<tr>
<td></td>
<td>there are</td>
<td>0.72</td>
</tr>
<tr>
<td>morgen</td>
<td>tomorrow</td>
<td>0.9</td>
</tr>
<tr>
<td>fliege ich</td>
<td>will I fly</td>
<td>0.63</td>
</tr>
<tr>
<td></td>
<td>will fly</td>
<td>0.17</td>
</tr>
<tr>
<td></td>
<td>I will fly</td>
<td>0.13</td>
</tr>
</tbody>
</table>
Two responsibilities

- Divide the source sentence into phrases
  - Standard approach: uniform distribution over all possible segmentations
  - How many segmentations are there?
- Reorder the phrases
  - Standard approach: Markov model on phrases (parameterized with log-linear model)
Reordering Model

Scoring function: \( d(x) = \alpha^{|x|} \) — exponential with distance
Learning Phrases

- Latent segmentation variable
- Latent phrasal inventory
- Parallel data
- EM?

Computational problem: summing over all segmentations and alignments is \#P-complete

Modeling problem: MLE has a degenerate solution.
Learning Phrases

• Three stages
  • word alignment
  • extraction of phrases
  • estimation of phrase probabilities
Consistent Phrases

All words of the phrase pair have to align to each other.
<table>
<thead>
<tr>
<th>watashi</th>
<th>open</th>
<th>the</th>
<th>box</th>
</tr>
</thead>
<tbody>
<tr>
<td>wa</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>hako</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>wo</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>akemasu</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
I open the box

watashi
wa
hako
wo
akemasu

akemasu / open
I open the box

watashi wa / I

wa

hako

wo

akemasu

watashi wa / I
I open the box

watashi
wa
hako
wo
akemasu

watashi / I
### Phrase Extraction

<table>
<thead>
<tr>
<th>watashi</th>
<th>wa</th>
<th>hako</th>
<th>wo</th>
<th>akemasu</th>
</tr>
</thead>
<tbody>
<tr>
<td>✔️</td>
<td>✔️</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**I open the box**

**watashi wa / I**
I open the box

watashi wa hako wo akemasu

hako wo / box
**Phrase Extraction**

```
<table>
<thead>
<tr>
<th>watashi</th>
<th>wa</th>
<th>hako</th>
<th>wo</th>
<th>akemasu</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>open</td>
<td>the</td>
<td>box</td>
<td></td>
</tr>
</tbody>
</table>
```

**hako wo / the box**
<table>
<thead>
<tr>
<th>watashi</th>
<th>wa</th>
<th>hako</th>
<th>wo</th>
<th>akemamus</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>open</td>
<td>the</td>
<td>box</td>
<td></td>
</tr>
</tbody>
</table>

**Phrase Extraction**

I open the box

**hako wo / open the box**
<table>
<thead>
<tr>
<th>watashi</th>
<th>wa</th>
<th>hako</th>
<th>wo</th>
<th>akemasu</th>
</tr>
</thead>
<tbody>
<tr>
<td>I open</td>
<td>the box</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

I open the box

hako wo / open the box
Phrase Extraction

I open the box

watashi wa hako wo akemasu / open the box

hako wo akemasu / open the box
Estimating Probabilities

• What is the MLE?
  • Depends on the alignment model!

• Two options
  • EM over restricted space
  • Assume all alignments equally likely - count and normalize phrase pairs
Adapted from Koehn (2006)
Maria no dio una bofetada a la bruja verde

Mary did not give a slap to the witch green

did not a slap by hag bawdy

no slap to the green witch

did not give the the witch

Adapted from Koehn (2006)
Maria no dio una bofetada a la bruja verde

Mary did not give a slap to the witch green

did not a slap by hag bawdy

no slap to the green witch

did not give the the witch

Adapted from Koehn (2006)
Decoding algorithm

- Translation as a search problem
- Partial hypothesis keeps track of
  - which source words have been translated (coverage vector)
  - $n-1$ most recent words of English (for LM!)
  - a back pointer list to the previous hypothesis + (e,f) phrase pair used
  - the (partial) translation probability
  - the estimated probability of translating the remaining words (precomputed, a function of the coverage vector)
- **Start state**: no translated words, $E=<s>$, $bp=nil$
- **Goal state**: all translated words
Decoding algorithm

- $Q[0] \leftarrow$ Start state
- for $i = 0$ to $|f| - 1$
  - Keep $b$ best hypotheses at $Q[i]$
  - for each hypothesis $h$ in $Q[i]$
    - for each untranslated span in $h.c$ for which there is a translation $<e,f>$ in the phrase table
      - $h' = h$ extend by $<e,f>$
      - Is there an item in $Q[|h'.c|]$ with $= LM$ state?
        - yes: update the item bp list and probability
        - no: $Q[|h'.c|] \leftarrow h'$
  - Find the best hypothesis in $Q[|f|]$, reconstruction translation by following back pointers
Maria no dio una bofetada a la bruja verde

...
f: Maria no dio una bofetada a la bruja verde

Q[0] Q[1] Q[2] ...

Mary

Mary

e: <s> Mary
c: *--------
p: 0.9

p: 1.0
f: Maria no dio una bofetada a la bruja verde

Q[0]         Q[1]         Q[2]       ...

Mary

Maria

Mari

no
dio
una
bofetada
da
la
bruja
verde

Tuesday, February 19, 13
f: Maria no dio una bofetada a la bruja verde
Q[0] Q[1] Q[2] ...
Maria no dio una bofetada a la bruja verde

Mary did not...
Maria no dio una bofetada a la bruja verde

Mary

Maria

Mary did not

did not
Mary did not slap the witch.
Reordering

- Language express words in different orders
  - bruja verde vs. green witch
- Phrase pairs can “memorize” some of these
- More general: in decoding, “skip ahead”
- Problem:
  - Won’t “easy parts” of the sentence be translated first?
- Solution:
  - **Future cost estimate**
  - For every coverage vector, estimate what it will cost to translate the remaining untranslated words
  - When pruning, use p * future cost!
f: Maria no dio una bofetada a la bruja verde

Q[0]  Q[1]  Q[2] ...

Mary
0.9  8.6e-9

Maria
0.3  8.6e-9

Maria no dio una bofetada a la bruja verde
f: Maria no dio una bofetada a la bruja verde

Q[0]   Q[1]   Q[2]   ...

Mary
0.9  8.6e-9

Maria
0.3  8.6e-9

Not
0.4  1.0e-9

Maria no dio una bofetada a la bruja verde.
f: Maria no dio una bofetada a la bruja verde

Future costs make these hypotheses comparable.
Decoding summary

- Finding the best hypothesis is NP-hard

- Even with no language model, there are an exponential number of states!

- Solution 1: limit reordering

- Solution 2: (lossy) pruning