Phrase-Based Models

Philipp Koehn

11 May 2015
Motivation

• Word-Based Models translate *words* as atomic units

• Phrase-Based Models translate *phrases* as atomic units

• Advantages:
  – many-to-many translation can handle non-compositional phrases
  – use of local context in translation
  – the more data, the longer phrases can be learned

• ”Standard Model”, used by Google Translate and others
Phrase-Based Model

- Foreign input is segmented in phrases
- Each phrase is translated into English
- Phrases are reordered

\[
\begin{align*}
\text{natuerlich} & \rightarrow \text{of course} \\
\text{hat} & \rightarrow \text{john} \\
\text{john} & \rightarrow \text{has} \\
\text{spass am} & \rightarrow \text{fun with the} \\
\text{spiel} & \rightarrow \text{game}
\end{align*}
\]
Phrase Translation Table

- Main knowledge source: table with phrase translations and their probabilities

- Example: phrase translations for natuerlich

| Translation       | Probability $\phi(e|f)$ |
|-------------------|------------------------|
| of course         | 0.5                    |
| naturally         | 0.3                    |
| of course ,       | 0.15                   |
| , of course ,     | 0.05                   |
Real Example

- Phrase translations for den Vorschlag learned from the Europarl corpus:

<table>
<thead>
<tr>
<th>English</th>
<th>$\phi(\bar{e} \vert f)$</th>
<th>English</th>
<th>$\phi(\bar{e} \vert f)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>the proposal</td>
<td>0.6227</td>
<td>the suggestions</td>
<td>0.0114</td>
</tr>
<tr>
<td>'s proposal</td>
<td>0.1068</td>
<td>the proposed</td>
<td>0.0114</td>
</tr>
<tr>
<td>a proposal</td>
<td>0.0341</td>
<td>the motion</td>
<td>0.0091</td>
</tr>
<tr>
<td>the idea</td>
<td>0.0250</td>
<td>the idea of</td>
<td>0.0091</td>
</tr>
<tr>
<td>this proposal</td>
<td>0.0227</td>
<td>the proposal ,</td>
<td>0.0068</td>
</tr>
<tr>
<td>proposal</td>
<td>0.0205</td>
<td>its proposal</td>
<td>0.0068</td>
</tr>
<tr>
<td>of the proposal</td>
<td>0.0159</td>
<td>it</td>
<td>0.0068</td>
</tr>
<tr>
<td>the proposals</td>
<td>0.0159</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

- lexical variation (proposal vs suggestions)
- morphological variation (proposal vs proposals)
- included function words (the, a, ...)
- noise (it)
Linguistic Phrases?

- Model is not limited to linguistic phrases (noun phrases, verb phrases, prepositional phrases, ...)

- Example non-linguistic phrase pair
  
  spass am → fun with the

- Prior noun often helps with translation of preposition

- Experiments show that limitation to linguistic phrases hurts quality
modeling
Noisy Channel Model

- We would like to integrate a language model

- Bayes rule

\[
\arg\max_e p(e|f) = \arg\max_e \frac{p(f|e) p(e)}{p(f)} = \arg\max_e p(f|e) p(e)
\]
Noisy Channel Model

- Applying Bayes rule also called noisy channel model
  - we observe a distorted message R (here: a foreign string \( f \))
  - we have a model on how the message is distorted (here: translation model)
  - we have a model on what messages are probably (here: language model)
  - we want to recover the original message S (here: an English string \( e \))
More Detail

• Bayes rule

\[ e_{\text{best}} = \arg\max_e p(e|f) \]
\[ = \arg\max_e p(f|e) p_{\text{LM}}(e) \]

– translation model \( p(e|f) \)
– language model \( p_{\text{LM}}(e) \)

• Decomposition of the translation model

\[ p(f_1^I|e_1^I) = \prod_{i=1}^{I} \phi(f_i|e_i) \ d(start_i - end_{i-1} - 1) \]

– phrase translation probability \( \phi \)
– reordering probability \( d \)
Distance-Based Reordering

### Scoring function:
\[ d(x) = \alpha^{|x|} \] — exponential with distance

### Table

<table>
<thead>
<tr>
<th>phrase</th>
<th>translates</th>
<th>movement</th>
<th>distance</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1–3</td>
<td>start at beginning</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>6</td>
<td>skip over 4–5</td>
<td>+2</td>
</tr>
<tr>
<td>3</td>
<td>4–5</td>
<td>move back over 4–6</td>
<td>-3</td>
</tr>
<tr>
<td>4</td>
<td>7</td>
<td>skip over 6</td>
<td>+1</td>
</tr>
</tbody>
</table>
training
Learning a Phrase Translation Table

- Task: learn the model from a parallel corpus

- Three stages:
  - word alignment: using IBM models or other method
  - extraction of phrase pairs
  - scoring phrase pairs
michael
assumes
that
he
will
stay
in
the
house

geht
davon
aus
dass
er
im
haus
bleibt
Extracting Phrase Pairs

extract phrase pair consistent with word alignment:

assumes that / geht davon aus , dass
All words of the phrase pair have to align to each other.
Phrase pair \((\bar{e}, \bar{f})\) consistent with an alignment \(A\), if all words \(f_1, \ldots, f_n\) in \(\bar{f}\) that have alignment points in \(A\) have these with words \(e_1, \ldots, e_n\) in \(\bar{e}\) and vice versa:

\[
(\bar{e}, \bar{f}) \text{ consistent with } A \iff \\
\forall e_i \in \bar{e} : (e_i, f_j) \in A \rightarrow f_j \in \bar{f} \\
\text{AND } \forall f_j \in \bar{f} : (e_i, f_j) \in A \rightarrow e_i \in \bar{e} \\
\text{AND } \exists e_i \in \bar{e}, f_j \in \bar{f} : (e_i, f_j) \in A
\]
Smallest phrase pairs:

- michael — michael
- assumes — geht davon aus / geht davon aus ,
- that — dass / , dass
- he — er
- will stay — bleibt
- in the — im
- house — haus

unaligned words (here: German comma) lead to multiple translations
Larger Phrase Pairs

Michael assumes — Michael geht davon aus / Michael geht davon aus,
assumes that — geht davon aus, dass; assumes that he — geht davon aus, dass er
that he — dass er, dass er; in the house — im haus
Michael assumes that — Michael geht davon aus, dass
Michael assumes that he — Michael geht davon aus, dass er
Michael assumes that he will stay in the house — Michael geht davon aus, dass er im haus bleibt
assumes that he will stay in the house — geht davon aus, dass er im haus bleibt
that he will stay in the house — dass er im haus bleibt; dass er im haus bleibt,
he will stay in the house — er im haus bleibt; will stay in the house — im haus bleibt
Scoring Phrase Translations

- Phrase pair extraction: collect all phrase pairs from the data
- Phrase pair scoring: assign probabilities to phrase translations
- Score by relative frequency:

\[
\phi(f|e) = \frac{\text{count}(e, f)}{\sum_{f_i} \text{count}(e, f_i)}
\]
EM Training of the Phrase Model

• We presented a heuristic set-up to build phrase translation table (word alignment, phrase extraction, phrase scoring)

• Alternative: align phrase pairs directly with EM algorithm
  – initialization: uniform model, all $\phi(\bar{e}, \bar{f})$ are the same
  – expectation step:
    * estimate likelihood of all possible phrase alignments for all sentence pairs
  – maximization step:
    * collect counts for phrase pairs $(\bar{e}, \bar{f})$, weighted by alignment probability
    * update phrase translation probabilities $p(\bar{e}, \bar{f})$

• However: method easily overfits
  (learns very large phrase pairs, spanning entire sentences)
Size of the Phrase Table

- Phrase translation table typically bigger than corpus
  ... even with limits on phrase lengths (e.g., max 7 words)

→ Too big to store in memory?

- Solution for training
  - extract to disk, sort, construct for one source phrase at a time

- Solutions for decoding
  - on-disk data structures with index for quick look-ups
  - suffix arrays to create phrase pairs on demand
advanced modeling
Weighted Model

- Described standard model consists of three sub-models
  - phrase translation model $\phi(\vec{f}|\vec{e})$
  - reordering model $d$
  - language model $p_{LM}(e)$

$$e_{best} = \arg\max_e \prod_{i=1}^I \phi(\vec{f}_i|\vec{e}_i) \cdot d(start_i - end_{i-1} - 1) \cdot \prod_{i=1}^{|e|} p_{LM}(e_i|e_1...e_{i-1})$$

- Some sub-models may be more important than others

- Add weights $\lambda_\phi$, $\lambda_d$, $\lambda_{LM}$

$$e_{best} = \arg\max_e \prod_{i=1}^I \phi(\vec{f}_i|\vec{e}_i)^{\lambda_\phi} \cdot d(start_i - end_{i-1} - 1)^{\lambda_d} \cdot \prod_{i=1}^{|e|} p_{LM}(e_i|e_1...e_{i-1})^{\lambda_{LM}}$$
Log-Linear Model

• Such a weighted model is a log-linear model:

\[ p(x) = \exp \sum_{i=1}^{n} \lambda_i h_i(x) \]

• Our feature functions
  – number of feature function \( n = 3 \)
  – random variable \( x = (e, f, \text{start}, \text{end}) \)
  – feature function \( h_1 = \log \phi \)
  – feature function \( h_2 = \log d \)
  – feature function \( h_3 = \log p_{\text{LM}} \)
Weighted Model as Log-Linear Model

\[ p(e, a | f) = \exp(\lambda_\phi \sum_{i=1}^{I} \log \phi(f_i | \bar{e}_i) + \lambda_d \sum_{i=1}^{I} \log d(a_i - b_{i-1} - 1) + \lambda_{LM} \sum_{i=1}^{\|e\|} \log p_{LM}(e_i | e_1...e_{i-1})) \]
More Feature Functions

- Bidirectional alignment probabilities: $\phi(\bar{e}|\bar{f})$ and $\phi(\bar{f}|\bar{e})$

- Rare phrase pairs have unreliable phrase translation probability estimates → lexical weighting with word translation probabilities

\[
\text{lex}(\bar{e}|\bar{f}, a) = \prod_{i=1}^{\text{length}(\bar{e})} \frac{1}{|\{j|(i,j) \in a\}|} \sum_{\forall(i,j) \in a} w(e_i|f_j)
\]
More Feature Functions

• Language model has a bias towards short translations
  → word count: $wc(e) = \log |e|^\omega$

• We may prefer finer or coarser segmentation
  → phrase count $pc(e) = \log |I|^\rho$

• Multiple language models

• Multiple translation models

• Other knowledge sources
reordering
Lexicalized Reordering

- Distance-based reordering model is weak
  → learn reordering preference for each phrase pair
- Three orientations types: (m) monotone, (s) swap, (d) discontinuous

orientation ∈ \{m, s, d\}

\[ p_o(\text{orientation}|\vec{f}, \vec{e}) \]
Learning Lexicalized Reordering

- Collect orientation information during phrase pair extraction
  - if word alignment point to the top left exists → **monotone**
  - if a word alignment point to the top right exists → **swap**
  - if neither a word alignment point to top left nor to the top right exists → neither monotone nor swap → **discontinuous**
Learning Lexicalized Reordering

• Estimation by relative frequency

\[ p_o(\text{orientation}) = \frac{\sum_{\bar{f}} \sum_{\bar{e}} \text{count}(\text{orientation}, \bar{e}, \bar{f})}{\sum_o \sum_{\bar{f}} \sum_{\bar{e}} \text{count}(o, \bar{e}, \bar{f})} \]

• Smoothing with unlexicalized orientation model \( p(\text{orientation}) \) to avoid zero probabilities for unseen orientations

\[ p_o(\text{orientation} | \bar{f}, \bar{e}) = \frac{\sigma \ p(\text{orientation}) + \text{count}(\text{orientation}, \bar{e}, \bar{f})}{\sigma + \sum_o \text{count}(o, \bar{e}, \bar{f})} \]
operation sequence model
A Critique: Phrase Segmentation is Arbitrary

• If multiple segmentations possible - why choose one over the other?
  
  spass am spiel vs. spass am spiel

• When choose larger phrase pairs or multiple shorter phrase pairs?
  
  spass am spiel vs. spass am spiel vs. spass am spiel

• None of this has been properly addressed
A Critique: Strong Independence Assumptions

• Lexical context considered only within phrase pairs

\[
\text{spass am} \rightarrow \text{fun with}
\]

• No context considered between phrase pairs

\[
? \text{spass am} ? \rightarrow ? \text{fun with} ?
\]

• Some phrasal context considered in lexicalized reordering model
  ... but not based on the identity of neighboring phrases
Segmentation? Minimal Phrase Pairs

natürlich hat John Spaß am Spiel

of course John has fun with the game

⇓

natürlich hat John Spaß Spiel

of course John has fun game am with the
**Independence?**

**Consider Sequence of Operations**

<table>
<thead>
<tr>
<th>Step</th>
<th>Operation</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>o1</td>
<td>Generate(natürlich, of course)</td>
<td>natürlich ↓ of course</td>
</tr>
<tr>
<td>o2</td>
<td>Insert Gap</td>
<td>natürlich ↓ John</td>
</tr>
<tr>
<td>o3</td>
<td>Generate (John, John)</td>
<td>of course John</td>
</tr>
<tr>
<td>o4</td>
<td>Jump Back (1)</td>
<td>natürlich hat ↓ John</td>
</tr>
<tr>
<td>o5</td>
<td>Generate (hat, has)</td>
<td>of course John has</td>
</tr>
<tr>
<td>o6</td>
<td>Jump Forward</td>
<td>natürlich hat John ↓ of course John has</td>
</tr>
<tr>
<td>o7</td>
<td>Generate(natürlich, of course)</td>
<td>natürlich hat John Spaß ↓ of course John has fun</td>
</tr>
<tr>
<td>o8</td>
<td>Generate(am, with)</td>
<td>natürlich hat John Spaß am ↓ of course John has fun with the</td>
</tr>
<tr>
<td>o9</td>
<td>GenerateTargetOnly(the)</td>
<td>of course John has fun with the</td>
</tr>
<tr>
<td>o10</td>
<td>Generate(Spiel, game)</td>
<td>natürlich hat John Spaß am Spiel ↓ of course John has fun with the game</td>
</tr>
</tbody>
</table>
Operation Sequence Model

- Operations
  - generate (phrase translation)
  - generate target only
  - generate source only
  - insert gap
  - jump back
  - jump forward

- N-gram sequence model over operations, e.g., 5-gram model:

\[ p(o_1) \ p(o_2|o_1) \ p(o_3|o_1, o_2) \ldots p(o_{10}|o_6, o_7, o_8, o_9) \]
In Practice

• Operation Sequence Model used as additional feature function

• Significant improvements over phrase-based baseline

→ State-of-the-art systems include such a model
• Phrase Model

• Training the model
  – word alignment
  – phrase pair extraction
  – phrase pair scoring
  – EM training of the phrase model

• Log linear model
  – sub-models as feature functions
  – lexical weighting
  – word and phrase count features

• Lexicalized reordering model

• Operation sequence model