Decoding in Phrase-Based Machine Translation

(Building the translation)

Not required for the homework
Phrase Extraction
Phrase-Based MT

Unlabeled English data

Phrase table $P(f|e)$

Language model $P(e)$

\[
P(e|f) \propto P(f|e)P(e)
\]

Noisy channel model: combine scores from translation model + language model to translate foreign to English

“Translate faithfully but make fluent English”
Phrase-Based Decoding

- Noisy channel model: $P(e|f) \propto P(f|e) P(e)$ (ignore $P(f)$ term)
  - Translation model (TM)
  - Language model (LM)

- Inputs needed
  - Language model that scores $P(e_i|e_1, \ldots, e_{i-1}) \approx P(e_i|e_{i-n-1}, \ldots, e_{i-1})$
  - Phrase table: set of phrase pairs $(e, f)$ with probabilities $P(f|e)$

- What we want to find: $e$ produced by a series of phrase-by-phrase translations from an input $f$
Given an input sentence, look at our phrase table to find all possible translations of all possible spans

Monotonic translation: need to translate each word in order, explore paths in the lattice that don’t skip any words

Looks like Viterbi, but the scoring is more complicated

Koehn (2004)
Monotonic Translation

<table>
<thead>
<tr>
<th>Maria</th>
<th>no</th>
<th>dio</th>
<th>una</th>
<th>bofetada</th>
<th>a</th>
<th>la</th>
<th>bruja</th>
<th>verde</th>
</tr>
</thead>
</table>

- If we translate with beam search, what state do we need to keep in the beam?
- Score
- Where are we in the sentence
- What words have we produced so far (actually only need to remember the last 2 words when using a 3-gram LM)

\[
\text{arg max}_e \left[ \prod_{\langle e, f \rangle} P(f|e) \cdot \prod_{i=1}^{\left| e \right|} P(e_i|e_{i-1}, e_{i-2}) \right]
\]
Monotonic Translation

Beam state: where we’re at, what the current translation so far is, and score of that translation

Advancing state consists of trying each possible translation that could get us to this timestep
Monotonic Translation

<table>
<thead>
<tr>
<th>Maria</th>
<th>no</th>
<th>dio</th>
<th>una</th>
<th>bofetada</th>
<th>a</th>
<th>la</th>
<th>bruja</th>
<th>verde</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>Mary</em></td>
<td><em>not</em></td>
<td><em>give</em></td>
<td><em>a</em></td>
<td><em>slap</em></td>
<td><em>to</em></td>
<td><em>the</em></td>
<td><em>witch</em></td>
<td><em>green</em></td>
</tr>
<tr>
<td><em>did not</em></td>
<td><em>no</em></td>
<td><em>slap</em></td>
<td><em>by</em></td>
<td><em>green witch</em></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><em>did not give</em></td>
<td><em>to the</em></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><em>slap</em></td>
<td><em>to</em></td>
<td><em>the</em></td>
<td><em>the witch</em></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\[ \text{score} = \log P(\text{Mary}) \cdot P(\text{not} \mid \text{Mary}) \cdot P(\text{Maria} \mid \text{Mary}) \cdot P(\text{no} \mid \text{not}) \]

LM

TM

In reality: \[ \text{score} = \alpha \log P(\text{LM}) + \beta \log P(\text{TM}) \]

...and TM is broken down into several features
Moses

- Toolkit for machine translation due to Philipp Koehn + Hieu Hoang
  - Pharaoh (Koehn, 2004) is the decoder from Koehn’s thesis

- Moses implements word alignment, language models, and this decoder, plus a ton more stuff
  - Highly optimized and heavily engineered, could more or less build SOTA translation systems with this from 2007-2013
<table>
<thead>
<tr>
<th>SOURCE</th>
<th>Cela constituerait une solution transitoire qui permettrait de conduire à terme à une charte à valeur contraignante.</th>
</tr>
</thead>
<tbody>
<tr>
<td>HUMAN</td>
<td>That would be an interim solution which would make it possible to work towards a binding charter in the long term.</td>
</tr>
<tr>
<td>1x DATA</td>
<td>[this] [constituerait] [assistance] [transitoire] [who] [permettrait] [licences] [to] [terme] [to] [a] [charter] [to] [value] [contraignante] [.]</td>
</tr>
<tr>
<td>10x DATA</td>
<td>[it] [would] [a solution] [transitional] [which] [would] [of] [lead] [to] [term] [to a] [charter] [to] [value] [binding] [.]</td>
</tr>
<tr>
<td>100x DATA</td>
<td>[this] [would be] [a transitional solution] [which would] [lead to] [a charter] [legally binding] [.]</td>
</tr>
<tr>
<td>1000x DATA</td>
<td>[that would be] [a transitional solution] [which would] [eventually lead to] [a binding charter] [.]</td>
</tr>
</tbody>
</table>
Evaluating MT

- Fluency: does it sound good in the target language?
- Fidelity/adequacy: does it capture the meaning of the original?
- Automatic evaluation tries to approximate this...

- BLEU score: geometric mean of 1-, 2-, 3-, and 4-gram precision vs. a reference, multiplied by brevity penalty (penalizes short translations)
  - 1-gram precision: do you predict words that are in the reference?
  - 4-gram precision: to get this right, you need those words to be in the right order!

- Better metrics: human-in-the-loop variants
Syntactic MT
Syntactic MT

- Rather than use phrases, use a synchronous context-free grammar

NP → [DT₁ JJ₂ NN₃; DT₁ NN₃ JJ₂]

DT → [the, la]

DT → [the, le]

NN → [car, voiture]

JJ → [yellow, jaune]

Translation = parse the input with “half” of the grammar, read off the other half

Assumes parallel tree structures, but there can be reordering
Use lexicalized rules, look like “syntactic phrases”

Leads to HUGE grammars, parsing is slow

**Grammar**

\[
S \rightarrow \langle \text{VP} . ; \text{I VP} . \rangle \quad \text{or} \quad S \rightarrow \langle \text{VP} . ; \text{you VP} . \rangle
\]

\[
\text{VP} \rightarrow \langle \text{lo haré ADV} ; \text{will do it ADV} \rangle
\]

\[
S \rightarrow \langle \text{lo haré ADV} . ; \text{I will do it ADV} . \rangle
\]

\[
\text{ADV} \rightarrow \langle \text{de muy buen grado} ; \text{gladly} \rangle
\]

Slide credit: Dan Klein