Decoding in Phrase-Based Machine Translation

(Building the translation)

Not required for the homework
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)
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)
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>Mary</td>
<td>not</td>
<td>give</td>
<td>a</td>
<td>slap</td>
<td>to</td>
<td>the</td>
<td>witch</td>
<td>green</td>
</tr>
<tr>
<td>did not</td>
<td></td>
<td></td>
<td>a slap</td>
<td>by</td>
<td></td>
<td>green witch</td>
<td></td>
<td></td>
</tr>
<tr>
<td>no</td>
<td></td>
<td>slap</td>
<td></td>
<td>to the</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>did not give</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

...did not give

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

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

...and TM is broken down into several features
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>Mary</td>
<td>not</td>
<td>give a slap to the green witch</td>
<td>did not give a slap to the green witch</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>did not</td>
<td>give a slap to the green witch</td>
<td>did not give</td>
<td>slap to the</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Two ways to get here: *Maria + no dio* or *Maria no + dio*
- Beam is filled with options from multiple *segmentations* of input
Monotonic Translation

<table>
<thead>
<tr>
<th>María</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>give</em></td>
<td><em>a slap</em></td>
<td><em>by</em></td>
<td><em>green</em></td>
<td><em>witch</em></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><em>no</em></td>
<td><em>slap</em></td>
<td><em>to the</em></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><em>did not give</em></td>
<td><em>to</em></td>
<td><em>the</em></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- With beam size $k = \infty$, how many possible entries in the beam here?
Non-Monotonic Translation

- More flexible model: can visit source sentence “out of order”
- State needs to describe which words have been translated and which haven’t
- Big enough phrases already capture lots of reorderings, so this isn’t as important as you think
Training Decoders

score = $\alpha \log P(t) + \beta \log P(s|t)$

...and $P(s|t)$ is in fact more complex

- Usually 5-20 feature weights to set, want to optimize for BLEU score which is not differentiable

- MERT (Och 2003): decode to get 1000-best translations for each sentence in a small training set (<1000 sentences), do line search on parameters to directly optimize for BLEU
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
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
Rather than use phrases, use a *synchronous context-free grammar*

\[
\text{NP} \rightarrow \text{[DT}_1\text{ JJ}_2\text{ NN}_3; DT}_1\text{ NN}_3\text{ JJ}_2]
\]

\[
\text{DT} \rightarrow \text{[the, la]}
\]

\[
\text{DT} \rightarrow \text{[the, le]}
\]

\[
\text{NN} \rightarrow \text{[car, voiture]}
\]

\[
\text{JJ} \rightarrow \text{[yellow, jaune]}
\]

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**

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

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

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

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

Slide credit: Dan Klein