Decoding in Phrase-Based Machine Translation
(Building the translation)
Not required for the homework

Phrase Extraction

Phrase-Based MT

Unlabeled English data
Phrase table $P(\text{f|e})$
Language model $P(\text{e})$

"Translate faithfully but make fluent English"

Phrase-Based Decoding

- Noisy channel model: $P(\text{e|f}) \propto P(\text{f|e}) P(\text{e})$ (ignore $P(\text{f})$ term)
  - Translation model (TM)
  - Language model (LM)
- Inputs needed
  - Language model that scores $P(\text{e}_i|\text{e}_{i-1},\ldots,\text{e}_1) \approx P(\text{e}_i|\text{e}_{i-n},\ldots,\text{e}_{i-1})$
  - Phrase table: set of phrase pairs $(\text{e}, \text{f})$ with probabilities $P(\text{f|e})$
- What we want to find: $\text{e}$ produced by a series of phrase-by-phrase translations from an input $\text{f}$
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)

» 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

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)
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

---

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

NP → [DT1 JJ2 NN3; DT1 NN3 JJ2]
DT → [the, la]
DT → [the, le]
NN → [car, voiture]
JJ → [yellow, jaune]

the yellow car la voiture 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