Decoding in Phrase-Based Machine Translation

(Building the translation)

Not required for the homework

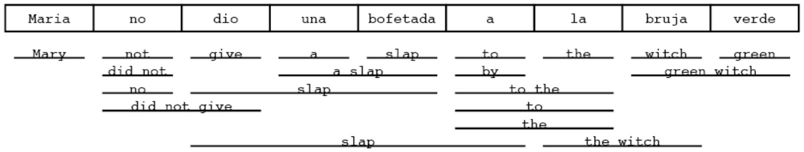
Phrase-Based Decoding

- Noisy channel model: P(**e**|**f**) ∝ 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**

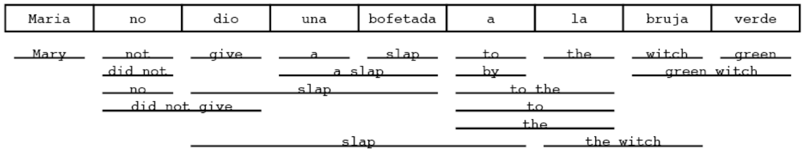
Phrase Lattice



- 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)

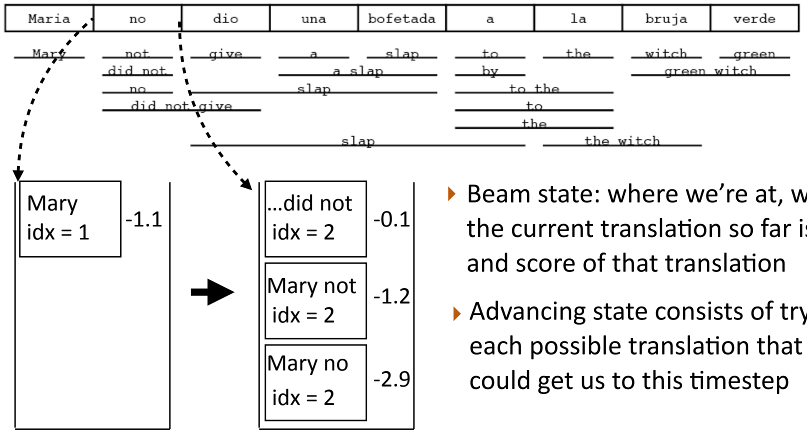
Monotonic Translation



- 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)

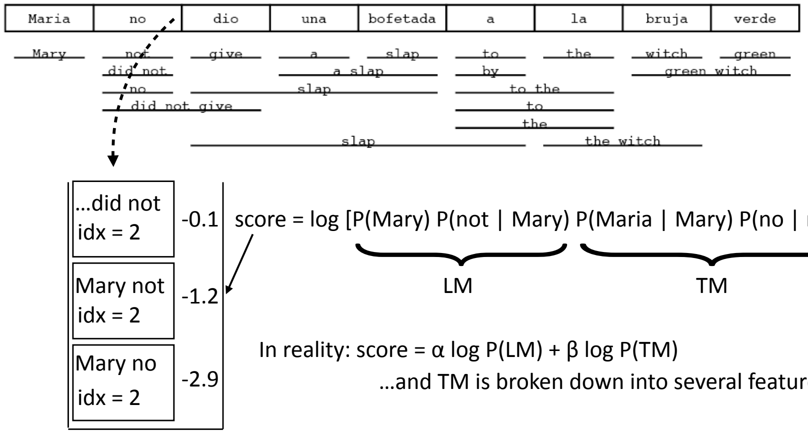
$$\arg\max_{\mathbf{e}} \left[\prod_{\langle \bar{e},\bar{f}\rangle} P(\bar{f}|\bar{e}) \cdot \prod_{i=1}^{|\mathbf{e}|} P(e_i|e_{i-1},e_{i-2}) \right]$$

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

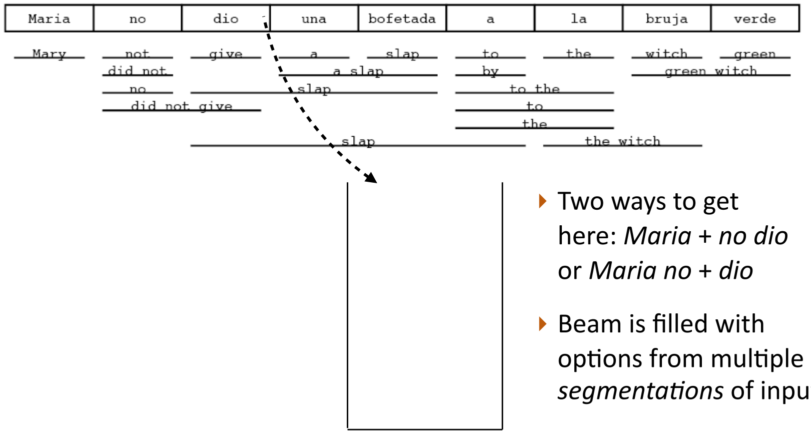


$$\text{score} = \log [\underbrace{P(\text{Mary})\, P(\text{not} \mid \text{Mary})}_{\text{LM}}\ \underbrace{P(\text{Maria} \mid \text{Mary})\, P(\text{no} \mid \text{not})}_{\text{TM}}]$$

In reality: score = $\alpha \log P(LM) + \beta \log P(TM)$

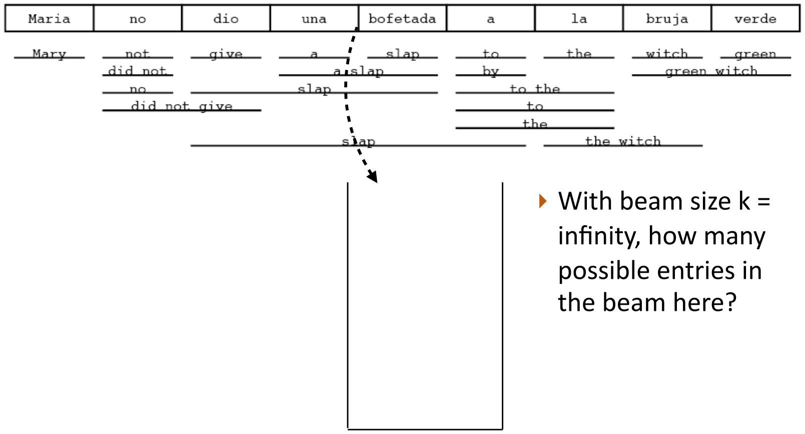
...and TM is broken down into several features

Monotonic Translation



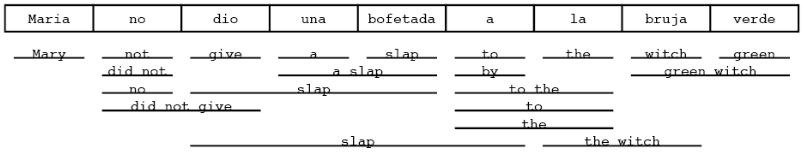
- Two ways to get here: *Maria + no dio* or *Maria no + dio*
- Beam is filled with options from multiple *segmentations* of input

Monotonic Translation

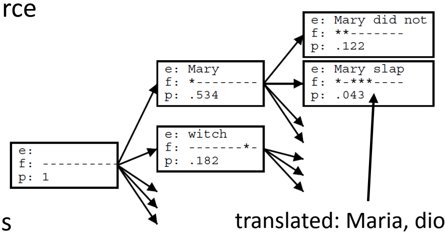


- With beam size k = infinity, how many possible entries in the beam here?

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



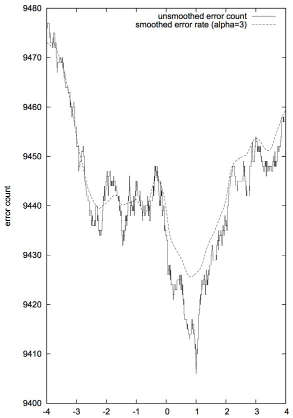
translated: Maria, dio, una, bofetada

Training Decoders

score = α log P(**t**) + β 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

Moses

SOURCE	Cela constituerait une solution transitoire qui permettrait de conduire à terme à une charte à valeur contraignante.
HUMAN	That would be an interim solution which would make it possible to work towards a binding charter in the long term .
1x DATA	[this] [constituerait] [assistance] [transitoire] [who] [permettrait] [licences] [to] [terme] [to] [a] [charter] [to] [value] [contraignante] [.]
10x DATA	[it] [would] [a solution] [transitional] [which] [would] [of] [lead] [to] [term] [to a] [charter] [to] [value] [binding] [.]
100x DATA	[this] [would be] [a transitional solution] [which would] [lead to] [a charter] [legally binding] [.]
1000x DATA	[that would be] [a transitional solution] [which would] [eventually lead to] [a binding charter] [.]

slide credit: Dan Klein

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

Syntactic MT

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

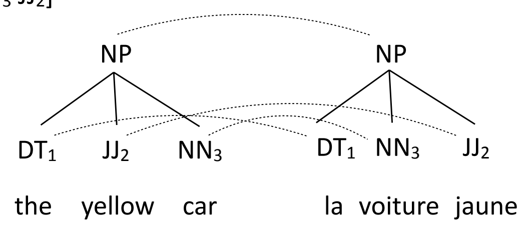
NP → [DT_1 JJ_2 NN_3; DT_1 NN_3 JJ_2]

DT → [the, la]

DT → [the, le]

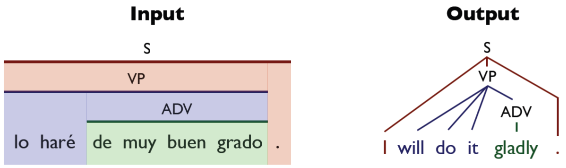
NN → [car, voiture]

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

Syntactic MT



- Use lexicalized rules, look like "syntactic phrases"
- Leads to HUGE grammars, parsing is slow

Grammar

S → ⟨ VP . ; I VP . ⟩ **OR** S → ⟨ VP . ; you VP . ⟩

VP → ⟨ lo haré ADV ; will do it ADV ⟩

S → ⟨ lo haré ADV . ; I will do it ADV . ⟩

ADV → ⟨ de muy buen grado ; gladly ⟩

Slide credit: Dan Klein