CS388: Natural Language Processing

Lecture 16: Machine Translation 1

Star Wars The Third Gathers: The Backstroke of the West
(subtitles machine translated from Chinese)

Greg Durrett

TEXAS
The University of Texas at Austin

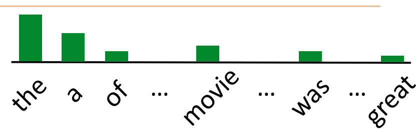
Some slides adapted from Dan Klein, UC Berkeley

Administrivia

- Project 2 due in a week

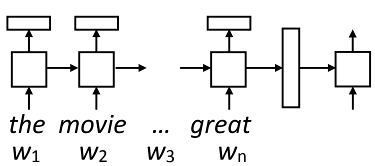
Recall: Pointer Networks

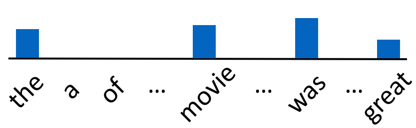
$$P(y_i|\mathbf{x}, y_1, \ldots, y_{i-1}) = \mathrm{softmax}(W[c_i; \bar{h}_i])$$



- Standard decoder (P_{vocab}): softmax over vocabulary, all words get >0 prob
- Pointer network: predict from *source words* instead of *target vocab*

$$P_{\text{pointer}}(y_i|\mathbf{x}, y_1, \ldots, y_{i-1}) \propto \begin{cases} \exp(h_j^\top V \bar{h}_i) & \text{if } y_i = w_j \\ 0 \text{ otherwise} & \end{cases}$$



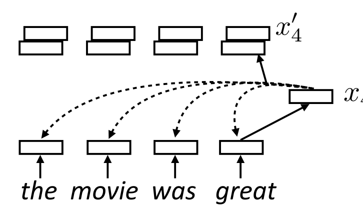


Recall: Self-Attention/Transformers

- Each word forms a "query" which then computes attention over each word

$$\alpha_{i,j} = \mathrm{softmax}(x_i^\top x_j) \quad \text{scalar}$$

$$x'_i = \sum_{j=1}^{n} \alpha_{i,j} x_j \quad \text{vector = sum of scalar * vector}$$



- Multiple "heads" analogous to different convolutional filters. Use parameters W_k and V_k to get different attention values + transform vectors

$$\alpha_{k,i,j} = \mathrm{softmax}(x_i^\top W_k x_j) \qquad x'_{k,i} = \sum_{j=1}^{n} \alpha_{k,i,j} V_k x_j$$

Vaswani et al. (2017)

This Lecture

- MT basics, evaluation
- Word alignment
- Phrase-based decoders
- Syntax-based decoders

MT Basics

MT Ideally

- *I have a friend* => `∃x friend(x,self)` => *J'ai un ami*

  J'ai une amie (friend is female)

  - May need information you didn't think about in your representation
  - Hard for semantic representations to cover everything
- Everyone has a friend => `∃x∀y friend(x,y)` `∀x∃y friend(x,y)` => Tous a un ami
  - Can often get away without doing all disambiguation — same ambiguities may exist in both languages

MT in Practice

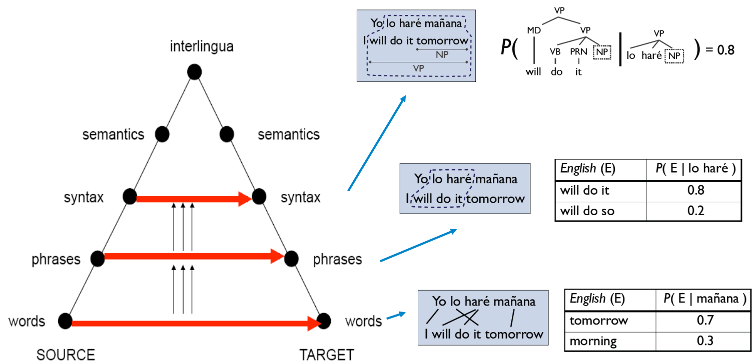
- Bitext: this is what we learn translation systems from. What can you learn?

Je fais un bureau	I'm making a desk
Je fais une soupe	I'm making soup
Je fais un bureau	I make a desk
Qu'est-ce que tu fais?	What are you making?

- What makes this hard? Not word-to-word translation

  Multiple translations of a single source (ambiguous)

Levels of Transfer: Vauquois Triangle



English (E)	*P*(E | lo haré)
will do it	0.8
will do so	0.2

English (E)	*P*(E | mañana)
tomorrow	0.7
morning	0.3

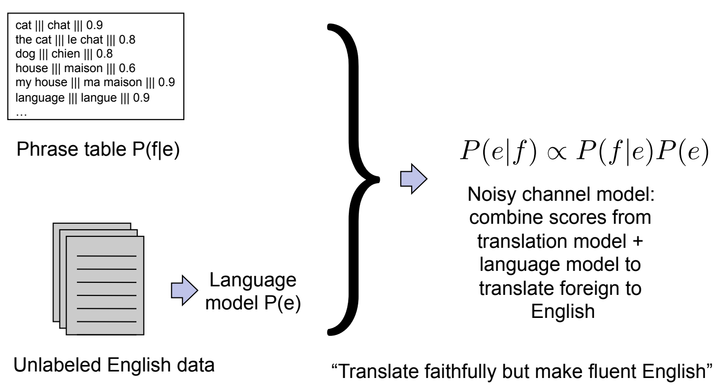
- Today: mostly phrase-based, some syntax

Slide credit: Dan Klein

Phrase-Based MT

- Key idea: translation works better the bigger chunks you use
- Remember phrases from training data, translate piece-by-piece and stitch those pieces together to translate
  - How to identify phrases? Word alignment over source-target bitext
  - How to stitch together? Language model over target language
  - Decoder takes phrases and a language model and searches over possible translations
- NOT like standard discriminative models (take a bunch of translation pairs, learn a ton of parameters in an end-to-end way)

Phrase-Based MT



"Translate faithfully but make fluent English"

Evaluating MT

- Fluency: does it sound good in the target language?
- Fidelity/adequacy: does it capture the meaning of the original?
- BLEU score: geometric mean of 1-, 2-, 3-, and 4-gram *precision* vs. a reference, multiplied by brevity penalty (penalizes short translations)

$$\text{BLEU} = \text{BP} \cdot \exp\left(\sum_{n=1}^{N} w_n \log p_n\right)$$

Typically n = 4, w_i = 1/4

$$\text{BP} = \begin{cases} 1 & \text{if } c > r \\ e^{(1-r/c)} & \text{if } c \leq r \end{cases}$$

r = length of reference
c = length of prediction

Word Alignment

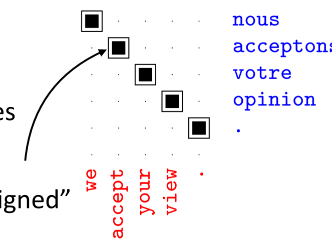
Word Alignment

- Input: a bitext, pairs of translated sentences

  nous acceptons votre opinion . ||| we accept your view

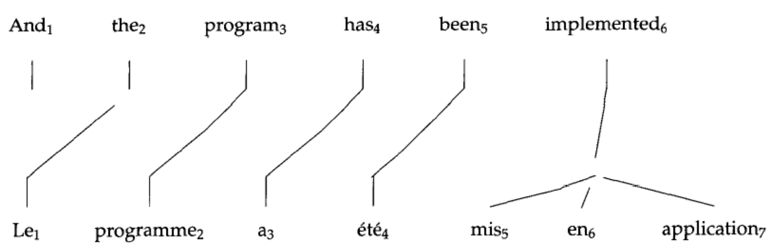
  nous allons changer d'avis ||| we are going to change our minds

- Output: alignments between words in each sentence
  - We will see how to turn these into phrases



"accept and acceptons are aligned"

1-to-Many Alignments



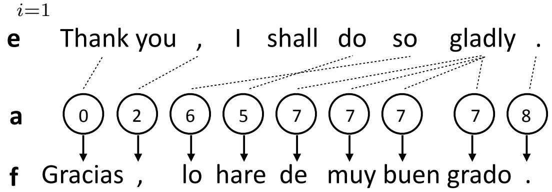
Word Alignment

- Models P(**f**|**e**): probability of "French" sentence being generated from "English" sentence according to a model
- Latent variable model: $P(\mathbf{f}|\mathbf{e}) = \sum_{\mathbf{a}} P(\mathbf{f},\mathbf{a}|\mathbf{e}) = \sum_{\mathbf{a}} P(\mathbf{f}|\mathbf{a},\mathbf{e})P(\mathbf{a})$
- Correct alignments should lead to higher-likelihood generations, so by optimizing this objective we will learn correct alignments

IBM Model 1

- Each French word is aligned to *at most* one English word

$$P(\mathbf{f},\mathbf{a}|\mathbf{e}) = \prod_{i=1}^{n} P(f_i|e_{a_i})P(a_i)$$



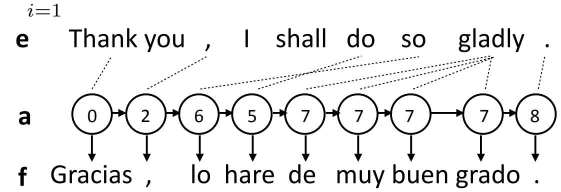
- Set P(a) uniformly (no prior over good alignments)
- $P(f_i|e_{a_i})$: word translation probability table. Learn with EM

Brown et al. (1993)

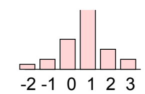
HMM for Alignment

- Sequential dependence between a's to capture monotonicity

$$P(\mathbf{f},\mathbf{a}|\mathbf{e}) = \prod_{i=1}^{n} P(f_i|e_{a_i})P(a_i|a_{i-1})$$



- Alignment dist parameterized by jump size: $P(a_j - a_{j-1})$

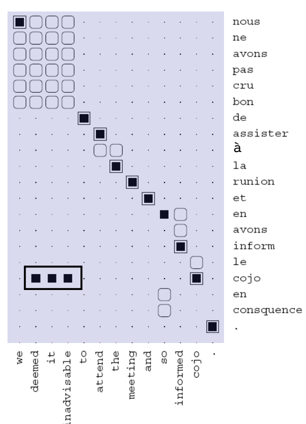


- $P(f_i|e_{a_i})$: same as before

Vogel et al. (1996)

HMM Model

- Alignments are generally monotonic (along diagonal)
- Some mistakes, especially when you have rare words (*garbage collection*)



Phrase Extraction

- Find contiguous sets of aligned words in the two languages that don't have alignments to other words

d'assister à la reunion et ||| to attend the meeting and

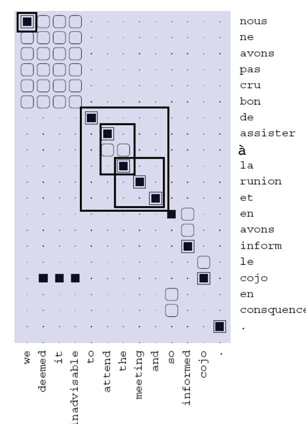
assister à la reunion ||| attend the meeting

la reunion and ||| the meeting and

nous ||| we

...

- Lots of phrases possible, count across all sentences and score by frequency



Decoding

Recall: *n*-gram Language Models

$P(\mathbf{w}) = P(w_1)P(w_2|w_1)P(w_3|w_1,w_2)\ldots$

- *n*-gram models: distribution of next word is a multinomial conditioned on previous *n*-1 words $P(w_i|w_1,\ldots,w_{i-1}) = P(w_i|w_{i-n+1},\ldots,w_{i-1})$

I visited San _____ put a distribution over the next word

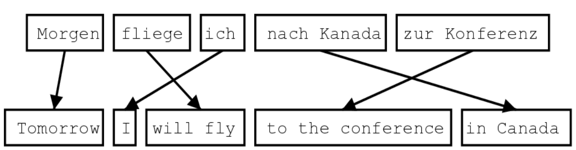
$$P(w|\text{visited San}) = \frac{\text{count}(\text{visited San}, w)}{\text{count}(\text{visited San})}$$

Maximum likelihood estimate of this 3-gram probability from a corpus

- Typically use ~5-gram language models for translation

Phrase-Based Decoding

- Inputs:
  - n-gram language model: $P(e_i|e_1,\ldots,e_{i-1}) \approx P(e_i|e_{i-n-1},\ldots,e_{i-1})$
  - 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**, possibly with reordering:



Phrase lattices are big!

Slide credit: Dan Klein

Phrase-Based Decoding

- Input — lo haré | rápidamente | .
- Translations — I'll do it | quickly | .
  — quickly | I'll do it | .

The decoder…

tries different segmentations,

translates phrase by phrase,

and considers reorderings.

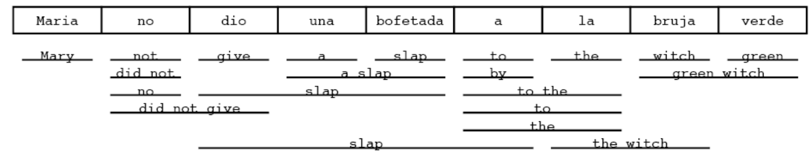
$$\arg\max_{\mathbf{e}} [P(\mathbf{f}|\mathbf{e}) \cdot P(\mathbf{e})]$$

- Decoding objective (for 3-gram LM)

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

Slide credit: Dan Klein

Monotonic Translation

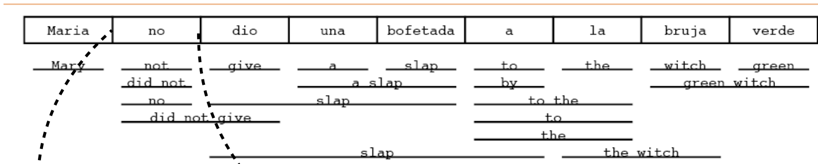


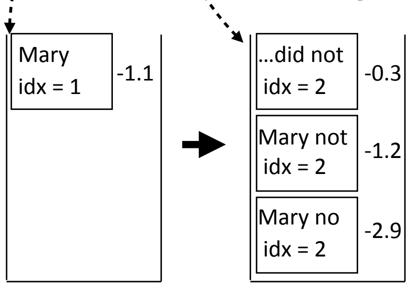
- If we translate with beam search, what state do we need to keep in the beam?
  - What have we translated so far?
  - What words have we produced so far? (need to remember the last 2 words for a 3-gram LM)

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

Koehn (2004)

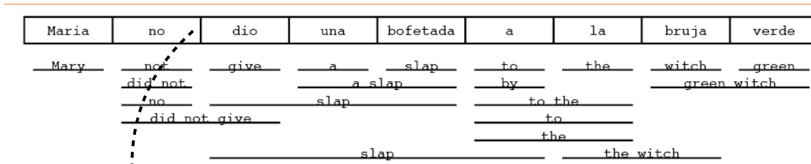
Monotonic Translation

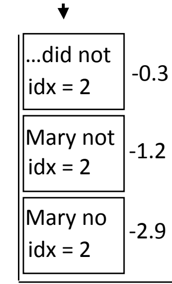




- Beam state: where we're at, what the current translation so far is, and score of that translation
- Advancing state consists of trying each possible translation that could get us to this timestep

Monotonic Translation





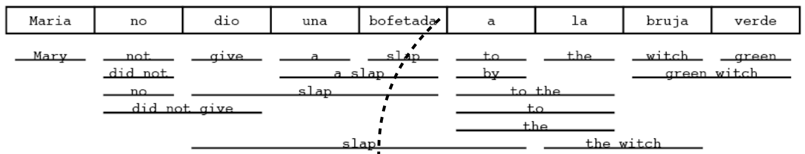
score = log [$\underbrace{\text{P(Mary) P(not | Mary)}}_{\text{LM}}$ $\underbrace{\text{P(Maria | Mary) P(no | not)}}_{\text{TM}}$]

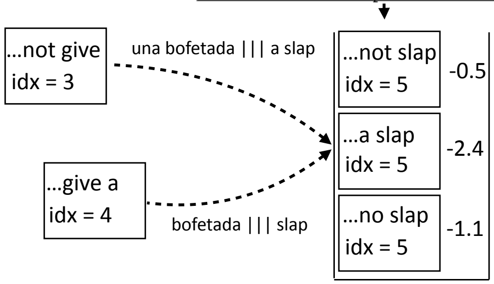
In reality: score = α log P(LM) + β log P(TM)

…and TM is broken down into several features

Koehn (2004)

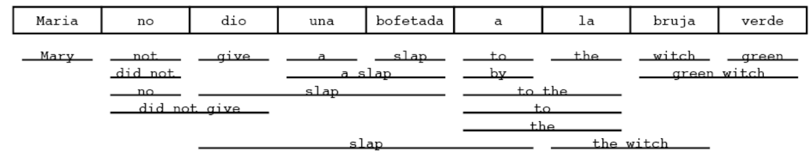
Monotonic Translation



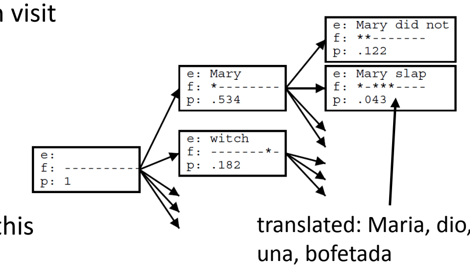


- Several paths can get us to this state, max over them (like Viterbi)
- Variable-length translation pieces = semi-HMM

Non-Monotonic Translation



- Non-monotonic translation: 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

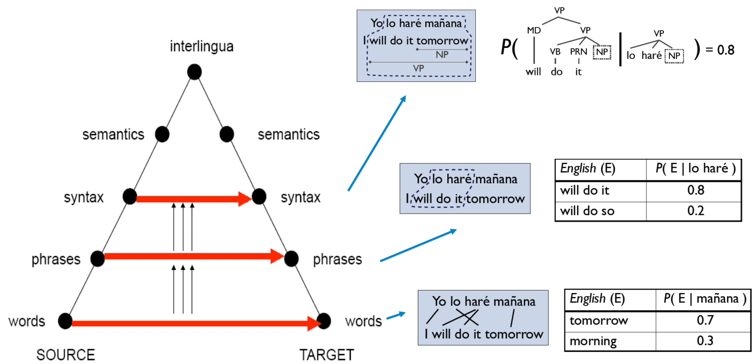


Moses

- 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 training regimes and more
  - Highly optimized and heavily engineered, could more or less build SOTA translation systems with this from 2007-2015
- Next time: results on these and comparisons to neural methods

Syntax

Levels of Transfer: Vauquois Triangle



- Is syntax a "better" abstraction than phrases?

Slide credit: Dan Klein

Syntactic MT

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

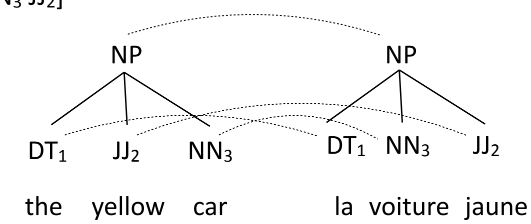
NP → [DT_1 JJ_2 NN_3; DT_1 NN_3 JJ_2]

DT → [the, la]

DT → [the, le]

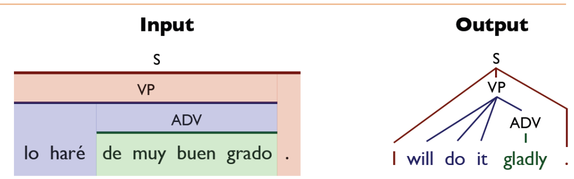
NN → [car, voiture]

JJ → [yellow, jaune]



- Translation = parse the input with "half" of the grammar, read off the other half
- Assumes parallel syntax up to 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

Takeaways

- Phrase-based systems consist of 3 pieces: aligner, language model, decoder
  - HMMs work well for alignment
  - N-gram language models are scalable and historically worked well
  - Decoder requires searching through a complex state space
- Lots of system variants incorporating syntax
- Next time: neural MT