Machine Translation

Introduction to Statistical MT

Statistical MT

- The intuition for Statistical MT comes from the **impossibility** of perfect translation
- Why perfect translation is impossible
  - Goal: Translating Hebrew *`adonai roi`* ("the lord is my shepherd") for a culture without sheep or shepherds
- Two options:
  - Something **fluent** and understandable, but not faithful:

    `The Lord will look after me`
  - Something **faithful**, but not fluent or natural

    `The Lord is for me like somebody who looks after animals with cotton-like hair`

A good translation is:

- Faithful
  - Has the same meaning as the source
  - (Causes the reader to draw the same inferences as the source would have)
- Fluent
  - Is natural, fluent, grammatical in the target

- Real translations trade off these two factors

Dan Jurafsky

Statistical MT: Faithfulness and Fluency formalized!

Peter Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, Robert L. Mercer. 1993. The Mathematics of Statistical Machine Translation: Parameter Estimation. Computational Linguistics 19:2, 263-311. **"The IBM Models"**

Given a French (foreign) sentence F, find an English sentence

$$\hat{E} = \underset{E \in English}{\operatorname{argmax}} P(E \mid F)$$

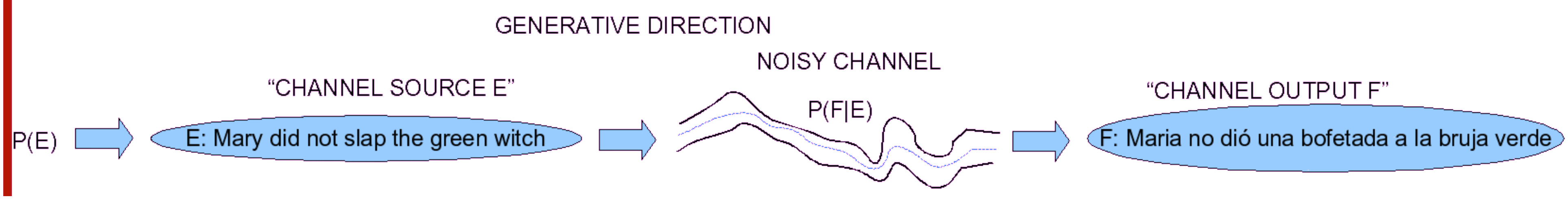
$$= \underset{E \in English}{\operatorname{argmax}} \frac{P(F \mid E) P(E)}{P(F)}$$

$$= \underset{E \in English}{\operatorname{argmax}} \underbrace{P(F \mid E)}_{\text{Translation Model}} \underbrace{P(E)}_{\text{Language Model}}$$

Convention in Statistical MT

- We always refer to translating
  - from input F, the foreign language (originally F = French)
  - to output E, English.
- Obviously statistical MT can translate from English into another language or between any pair of languages
- The convention helps avoid confusion about which way the probabilities are conditioned for a given example
- I will call the input F, or sometimes French, or Spanish.

The noisy channel model for MT



Fluency: P(E)

- We need a metric that ranks this sentence

  `That car almost crash to me`

as less fluent than this one:

  `That car almost hit me.`

- Answer: language models (N-grams!)

  P(me|hit) > P(to|crash)

  - And we can use any other more sophisticated model of grammar
- Advantage: this is monolingual knowledge!

Faithfulness: P(F|E)

- Spanish:
  - Maria no dió una bofetada a la bruja verde
- English candidate translations:
  - Mary didn't slap the green witch
  - Mary not give a slap to the witch green
  - The green witch didn't slap Mary
  - Mary slapped the green witch
- More faithful translations will be composed of phrases that are high probability translations
  - How often was "slapped" translated as "dió una bofetada" in a large **bitext** (parallel English-Spanish corpus)
  - We'll need to align phrases and words to each other in bitext

We treat Faithfulness and Fluency as independent factors

- P(F|E)'s job is to model "bag of words"; which words come from English to Spanish.
  - P(F|E) doesn't have to worry about internal facts about English word order.
- P(E)'s job is to do bag generation: put the following words in order:
  - a ground there in the hobbit hole lived a in

Three Problems for Statistical MT

- Language Model: given E, compute P(E)

  good English string → high P(E)
  random word sequence → low P(E)

- Translation Model: given (F,E) compute P(F | E)

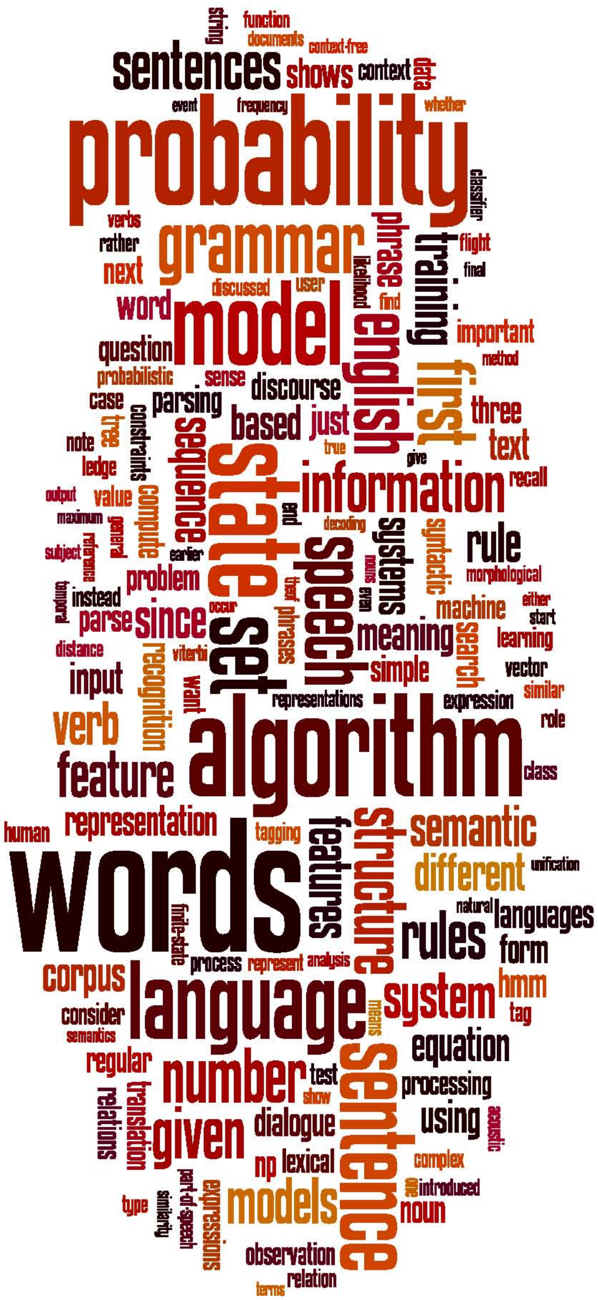
  (F,E) look like translations → high P(F | E)
  (F.E) don't look like translations → low P(F | E)

- Decoding algorithm: given LM, TM, F, find Ê

  Find translation E that maximizes P(E) * P(F | E)

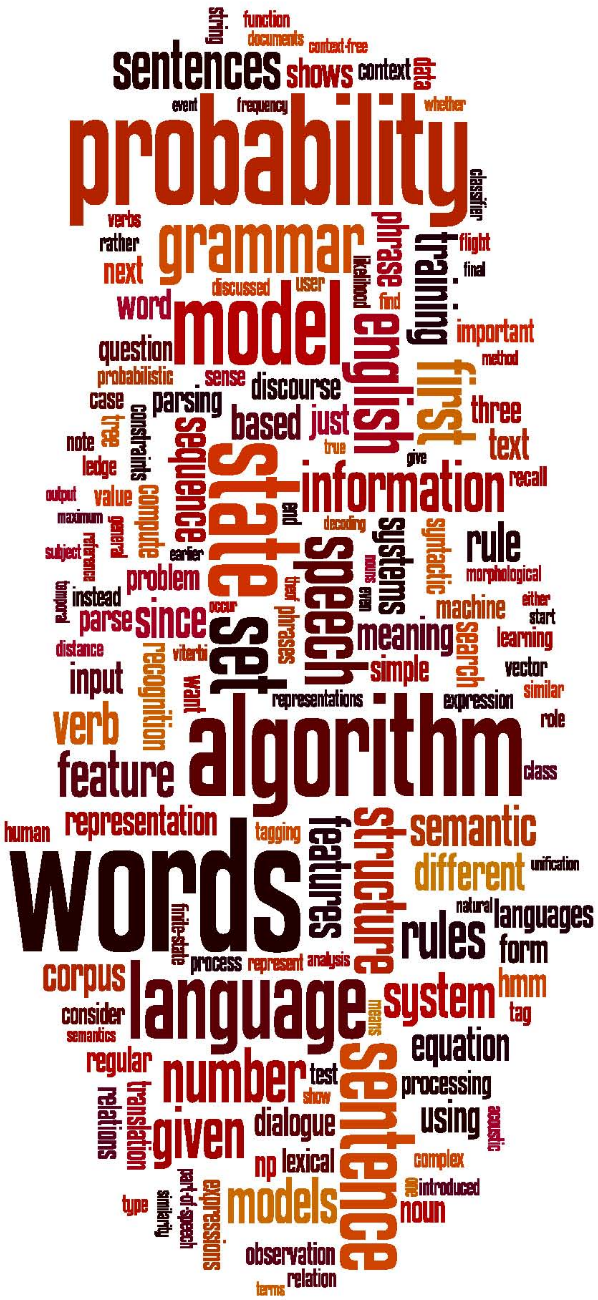
Language Model

- Use a standard *n*-gram language model for P(*E*).
- Can be trained on a large mono-lingual corpus
  - 5-gram grammar of English from terabytes of web data
  - More sophisticated parser-based language models can also help



Machine Translation

Introduction to Statistical MT

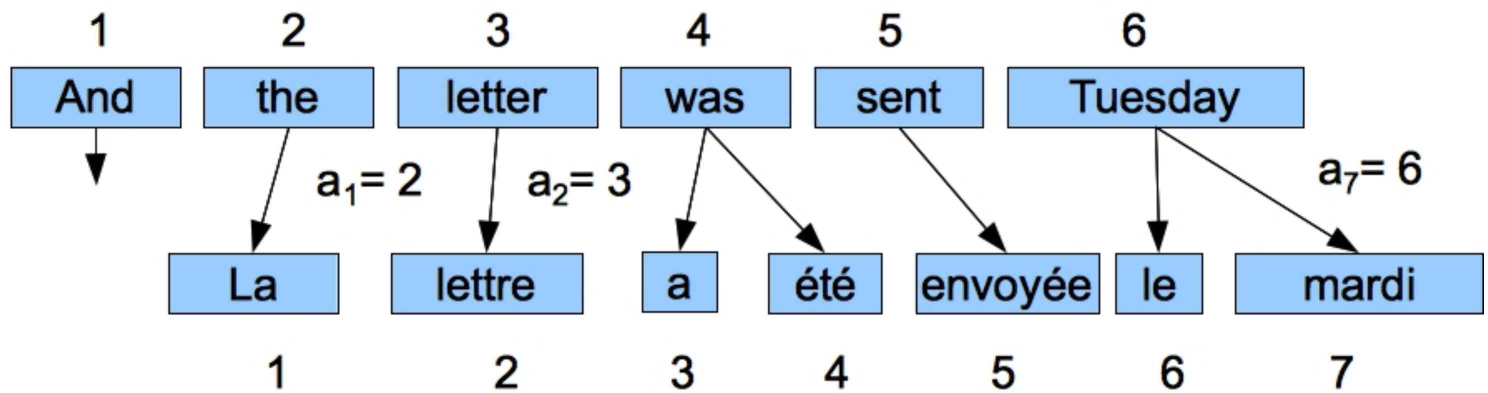


Machine Translation

Alignment and IBM Model 1

Word Alignment

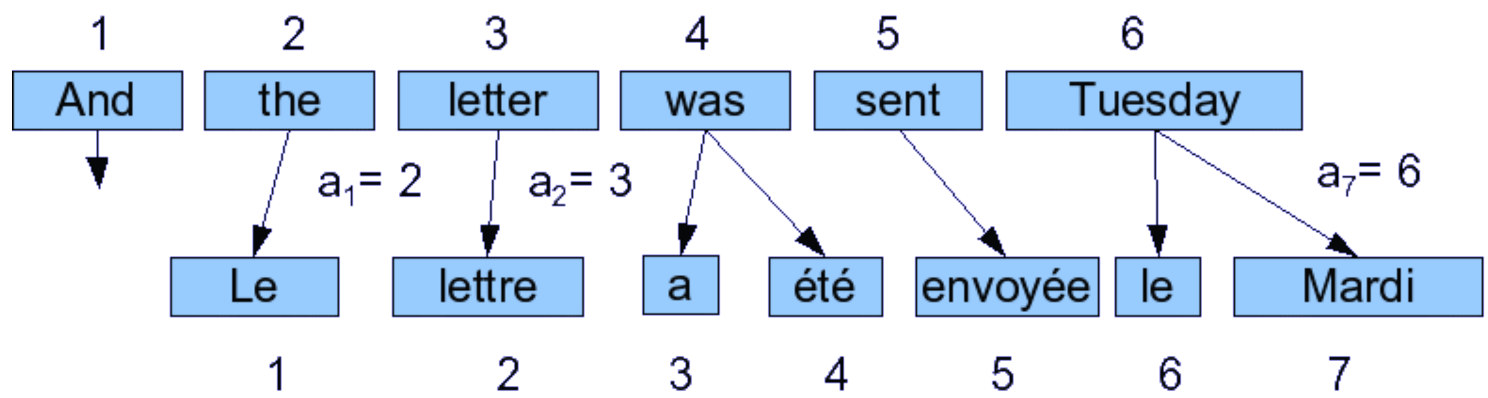
- A mapping between words in F and words in E



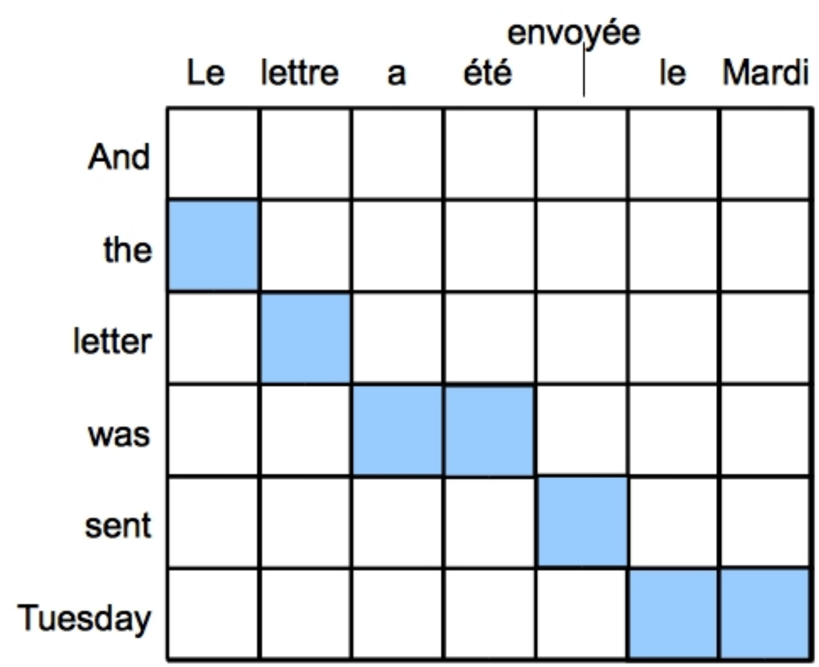
- Simplifying assumptions (for Model 1 and HMM alignments):
  - one-to-many (not many-to-one or many-to-many)
    - each French word comes from exactly one English word
  - An alignment is a vector of length J, one cell for each French word
    - The index of the English word that the French word comes from
- Alignment above is thus the vector $A = [2, 3, 4, 4, 5, 6, 6]$
- a_1=2, a_2=3, a_3=4, a_4=4…

Dan Jurafsky

Three representations of an alignment

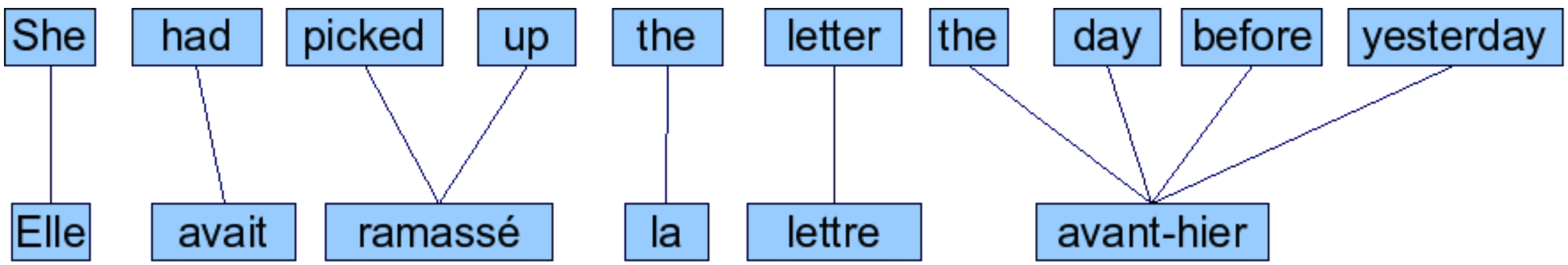


$$A = [2, 3, 4, 4, 5, 6, 6]$$

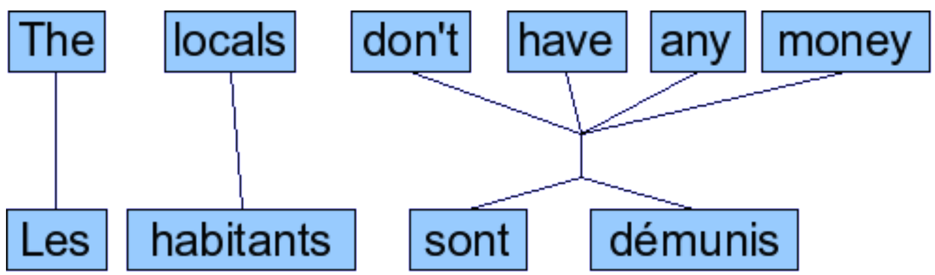


Alignments that don't obey one-to-many restriction

- Many to one:

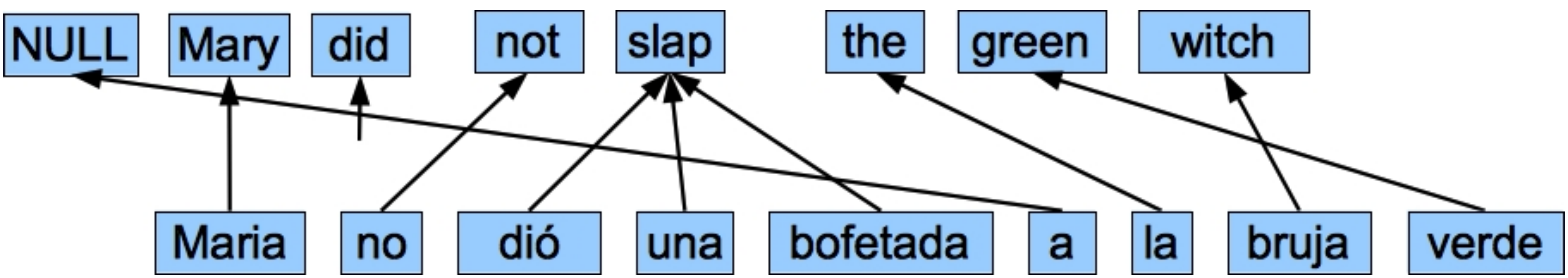


- Many to many:



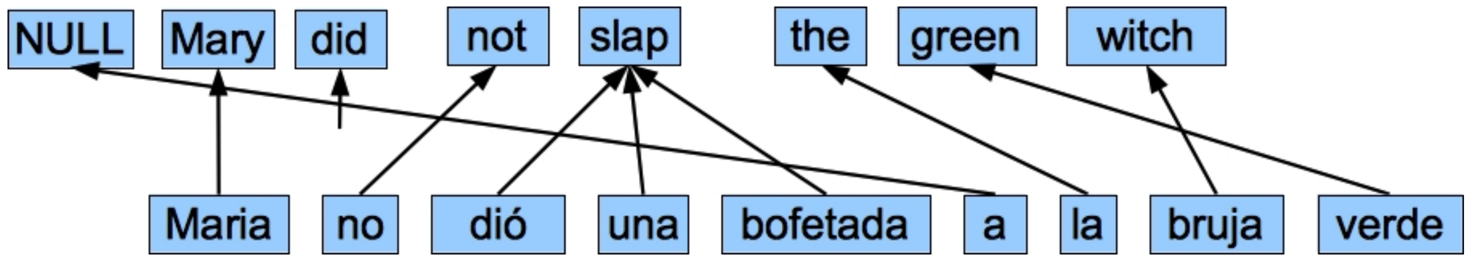
One addition: spurious words

- A word in the Spanish (French, foreign) sentence that doesn't align with any word in the English sentence is called a **spurious word**.
- We model these by pretending they are generated by a NULL English word e_0 :

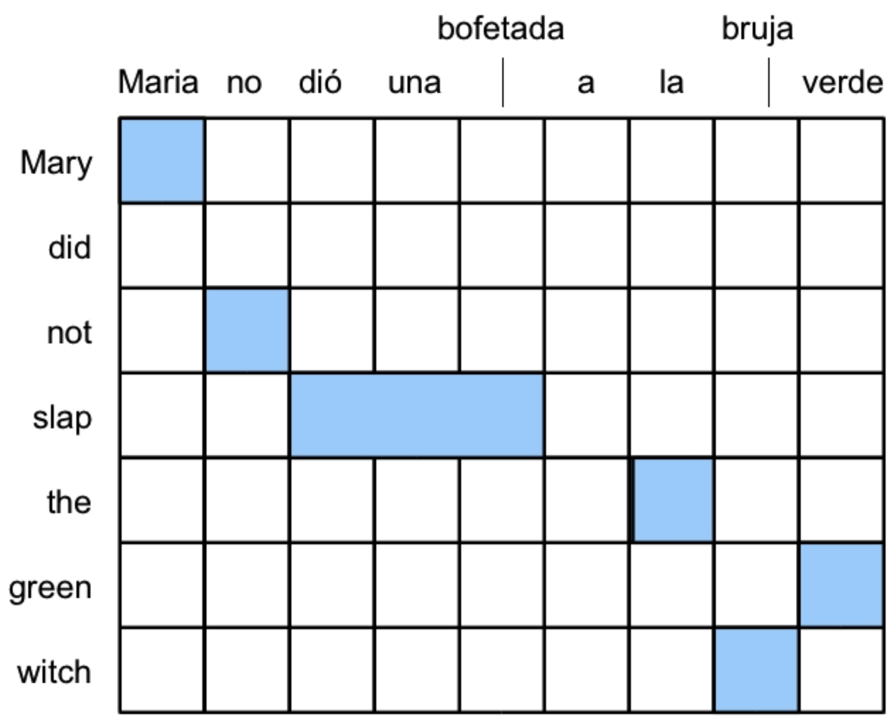


A = 1, 3, 4, 4, 4, 0, 5, 7, 6

Resulting alignment



A = [1, 3, 4, 4, 4, 0, 5, 7, 6]



Dan Jurafsky

Computing word alignments

- Word alignments are the basis for most translation algorithms
- Given two sentences F and E, find a good alignment
- But a word-alignment algorithm can also be part of a mini-translation model itself.

$$P(F \mid E) = \sum_{A} P(F, A \mid E)$$

- One of the most basic alignment models is also a simplistic translation model.

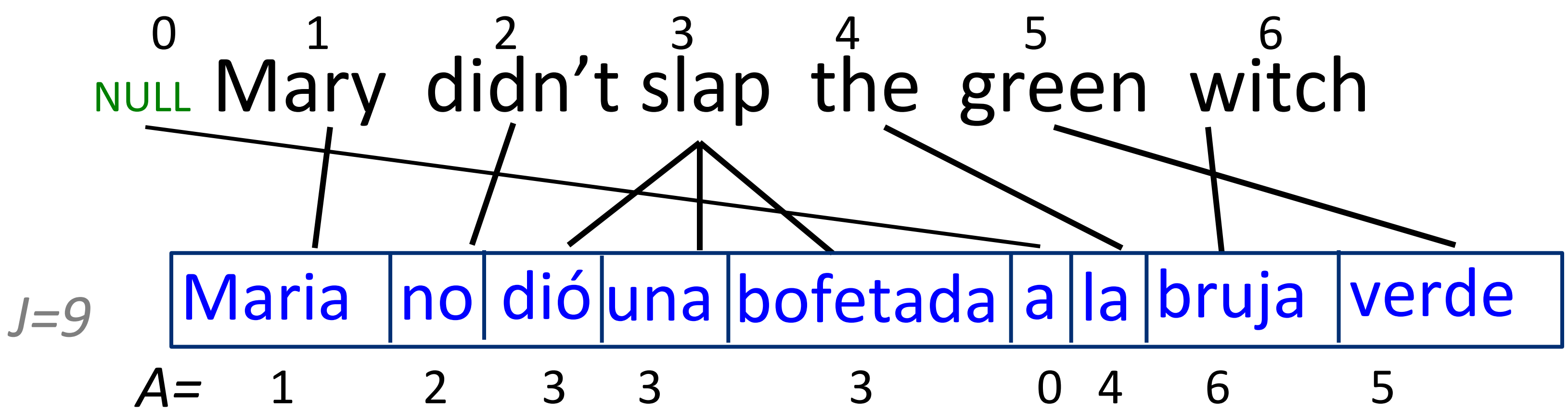
IBM Model 1

Peter Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, Robert L. Mercer. 1993. The Mathematics of Statistical Machine Translation: Parameter Estimation. Computational Linguistics 19:2, 263-311

- First of 5 "IBM models"
  - CANDIDE, the first complete SMT system, developed at IBM

- Simple generative model to produce F given $E=e_1, e_2, \ldots e_I$
  - Choose J, the number of words in F: $F=f_1, f_2, \ldots f_J$
  - Choose a 1-to-many alignment $A=a_1, a_2, \ldots a_J$
  - For each position in F, generate a word f_j from the aligned word in E: e_{a_j}

IBM Model 1: Generative Process



1. Choose J, the number of words in F: $F=f_1, f_2, \ldots f_J$
2. Choose a 1-to-many alignment $A=a_1, a_2, \ldots a_J$
3. For each position in F, generate a word f_j from the aligned word in E: e_{a_j}

Computing P(*F* | *E*) in IBM Model 1: P(F|E,A)

- *Let*

  e_{a_j} : the English word assigned to Spanish word f_j

  $t(f_x,e_y)$: probability of translating e_y as f_x

- If we knew E, the alignment A, and J, then:

$$P(F \mid E,A) = \prod_{j=1}^{J} t(f_j, e_{a_j})$$

- The probability of the Spanish sentence if we knew the English source, the alignment, and J

Computing P(F | E) in IBM Model 1: P(A|E)

- A normalization factor, since there are $(I+1)^J$ possible alignments:

$$P(A \mid E) = \frac{\varepsilon}{(I+1)^J}$$

- The probability of an alignment given the English sentence.

Computing P(*F* | *E*) in IBM Model 1: P(F,A|E) and then P(F|E)

$$P(A \mid E) = \frac{\varepsilon}{(I+1)^J} \qquad P(F \mid E, A) = \prod_{j=1}^{J} t(f_j, e_{a_j})$$

The probability of generating F through a particular alignment:

$$P(F, A \mid E) = \frac{\varepsilon}{(I+1)^J} \prod_{j=1}^{J} t(f_j, e_{a_j})$$

To get P(*F* | *E*), we sum over all alignments:

$$P(F \mid E) = \sum_A P(F, A \mid E) = \sum_A \frac{\varepsilon}{(I+1)^J} \prod_{j=1}^{J} t(f_j, e_{a_j})$$

Decoding for IBM Model 1

- Goal is to find the most probable alignment given a parameterized model.

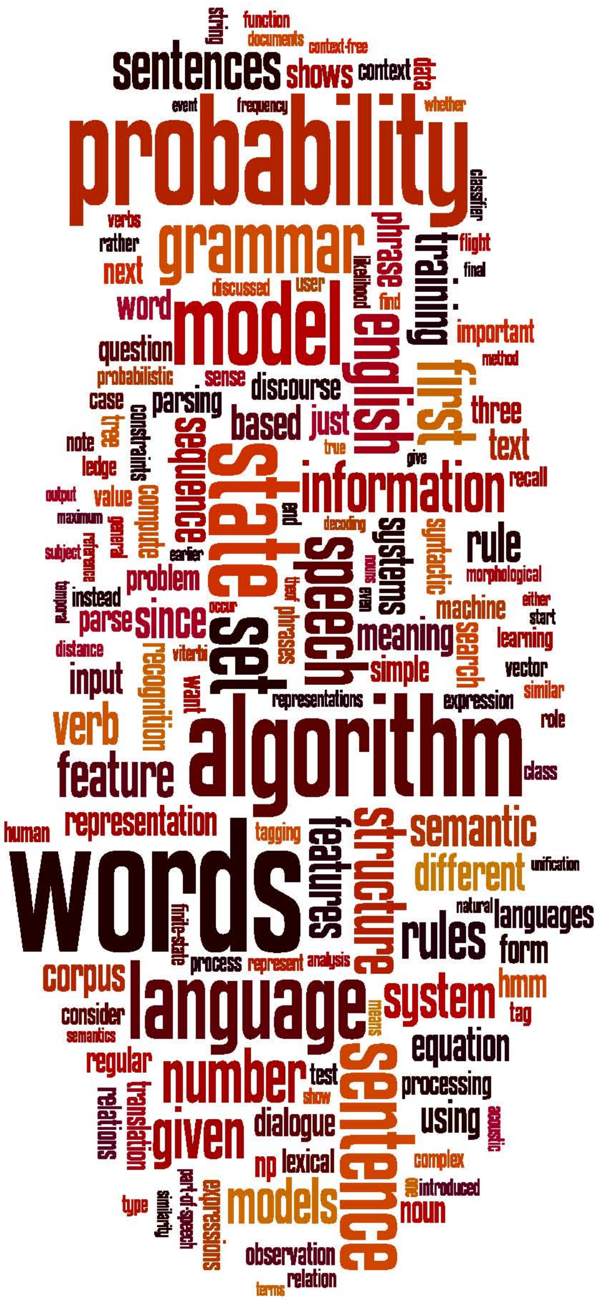
$$\hat{A} = \underset{A}{\operatorname{argmax}}\, P(F, A \mid E)$$

$$= \underset{A}{\operatorname{argmax}} \frac{P(J \mid E)}{(I+1)^J} \prod_{j=1}^{J} t(f_j, e_{a_j})$$

$$= \underset{A}{\operatorname{argmax}} \prod_{j=1}^{J} t(f_j, e_{a_j})$$

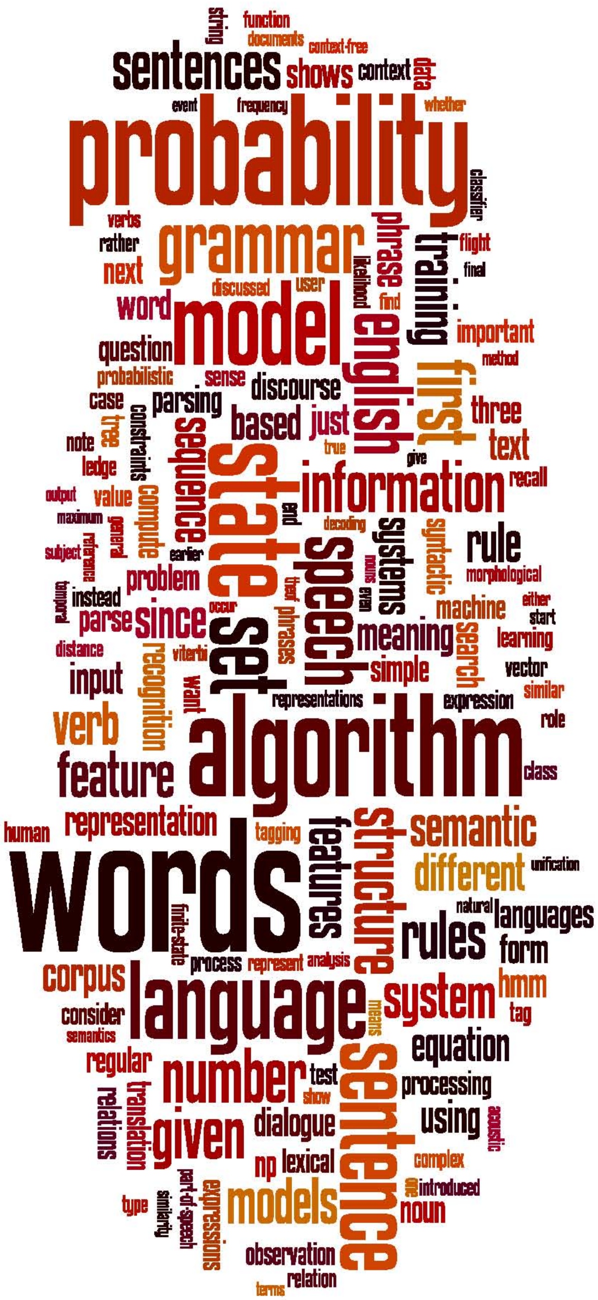
Since translation choice for each position *j* is independent, the product is maximized by maximizing each term:

$$a_j = \underset{0 \le i \le I}{\operatorname{argmax}}\, t(f_j, e_i) \qquad 1 \le j \le J$$



Machine Translation

Alignment and IBM Model 1



Machine Translation

Learning Word Alignments in IBM Model 1

Dan Jurafsky

Word Alignment

- Given a pair of sentences (one English, one French)
- Learn which English words align to which French words

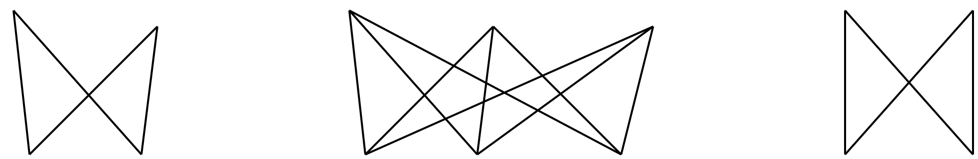
- Method: IBM Model 1
  - An iterative unsupervised algorithm
  - The EM (Expectation-Maximization) algorithm

Dan Jurafsky

EM for training alignment probabilities

Kevin Knight's example

… la maison … la maison bleue … la fleur …



… the house … the blue house … the flower …

Initial stage:

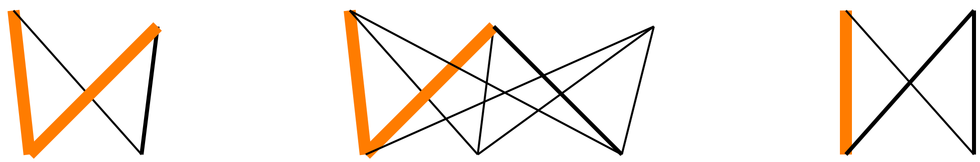
- All word alignments equally likely
- All P(french-word | english-word) equally likely

Dan Jurafsky

EM for training alignment probabilities

Kevin Knight's example

… la maison … la maison bleue … la fleur …



… the house … the blue house … the flower …

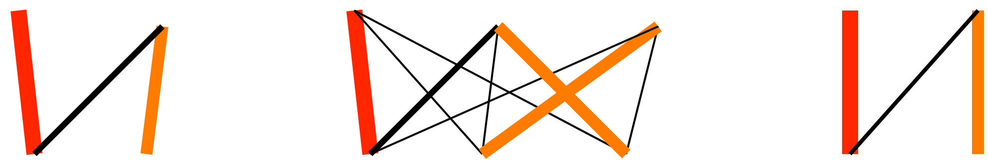
"la" and "the" observed to co-occur frequently,
so P(la | the) is increased.

Dan Jurafsky

EM for training alignment probabilities

Kevin Knight's example

… la maison … la maison bleue … la fleur …



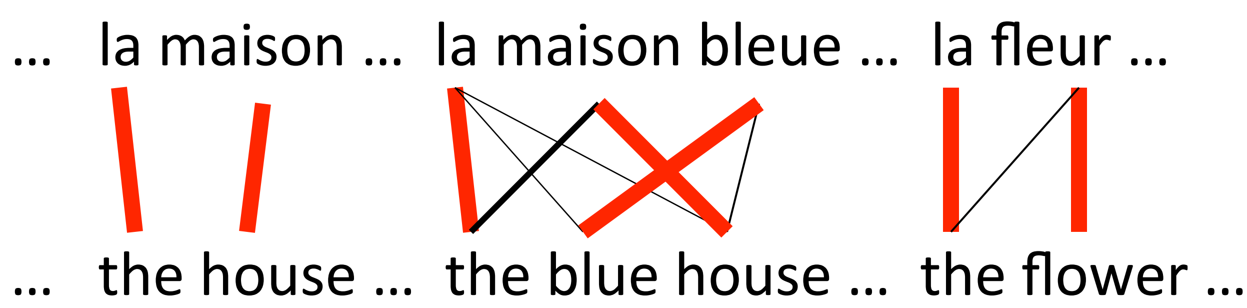
… the house … the blue house … the flower …

"house" co-occurs with both "la" and "maison",

- but P(maison | house) can be raised without limit, to 1.0
- while P(la | house) is limited because of "the"
- (pigeonhole principle)

EM for training alignment probabilities

Kevin Knight's example

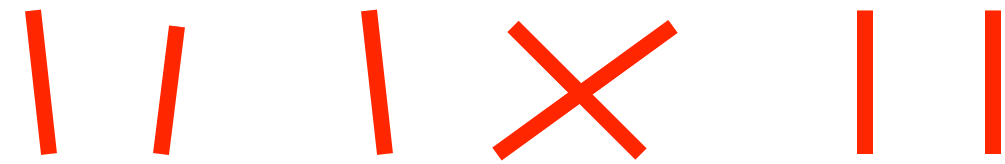


settling down after another iteration

EM for training alignment probabilities

Kevin Knight's example

… la maison … la maison bleue … la fleur …



… the house … the blue house … the flower …

- EM reveals inherent hidden structure!
- We can now estimate parameters from aligned corpus:

$$p(la|the) = 0.453$$
$$p(le|the) = 0.334$$
$$p(maison|house) = 0.876$$
$$p(bleu|blue) = 0.563$$

The EM Algorithm for Word Alignment

1. Initialize the model, typically with uniform distributions
2. Repeat

   E Step: Use the current model to compute the probability of all possible alignments of the training data

   M Step: Use these alignment probability estimates to re-estimate values for all of the parameters.

until converge (i.e., parameters no longer change)

Dan Jurafsky

Example EM Trace for Model 1 Alignment

- Simplified version of Model 1

  (No NULL word, and subset of alignments: ignore alignments for which English word aligns with no foreign word)

- E-step

(ignoring a constant here)

$$P(A,F \mid E) = \prod_{j=1}^{J} t(f_j \mid e_{a_j})$$

- Normalize to get probability of an alignment:

$$P(A \mid E,F) = \frac{P(A,F \mid E)}{\sum_A P(A,F \mid E)} = \frac{\prod_{j=1}^{J} t(f_j \mid e_{a_j})}{\sum_A \prod_{j=1}^{J} t(f_j \mid e_{a_j})}$$

Sample EM Trace for Alignment: E step

(IBM Model 1 with no NULL Generation)

$$P(A,F \mid E) = \prod_{j=1}^{J} t(f_j | e_{a_j})$$

Training Corpus

green house	the house
casa verde	la casa

$$P(A|E,F) = \frac{P(A,F|E)}{\sum_A P(A,F|E)}$$

Translation Probabilities

	verde	casa	la
green	1/3	1/3	1/3
house	1/3	1/3	1/3
the	1/3	1/3	1/3

Assume uniform initial probabilities

Compute Alignment Probabilities P(A, F | E)

green house / casa verde	green house / casa verde	the house / la casa	the house / la casa
1/3 X 1/3 = 1/9	1/3 X 1/3 = 1/9	1/3 X 1/3 = 1/9	1/3 X 1/3 = 1/9

Normalize to get P(A | F, E)

$$\frac{1/9}{2/9} = \frac{1}{2} \qquad \frac{1/9}{2/9} = \frac{1}{2} \qquad \frac{1/9}{2/9} = \frac{1}{2} \qquad \frac{1/9}{2/9} = \frac{1}{2}$$

EM example continued: M step

green house	green house	the house	the house
casa verde	casa verde	la casa	la casa
1/2	1/2	1/2	1/2

Compute weighted translation counts
$C(f_j, e_{a(j)})$ += $P(a|e,f)$

	verde	casa	la
green	1/2	1/2	0
house	1/2	1/2 + 1/2	1/2
the	0	1/2	1/2

Normalize rows to sum to one to estimate P(f | e)

	verde	casa	la
green	1/2	1/2	0
house	1/4	1/2	1/4
the	0	1/2	1/2

Dan Jurafsky

EM example continued

$$P(A,F \mid E) = \prod_{j=1}^{J} t(f_j \mid e_{a_j})$$

$$P(A \mid E,F) = \frac{P(A,F \mid E)}{\sum_A P(A,F \mid E)}$$

Translation Probabilities

	verde	casa	la
green	1/2	1/2	0
house	1/4	1/2	1/4
the	0	1/2	1/2

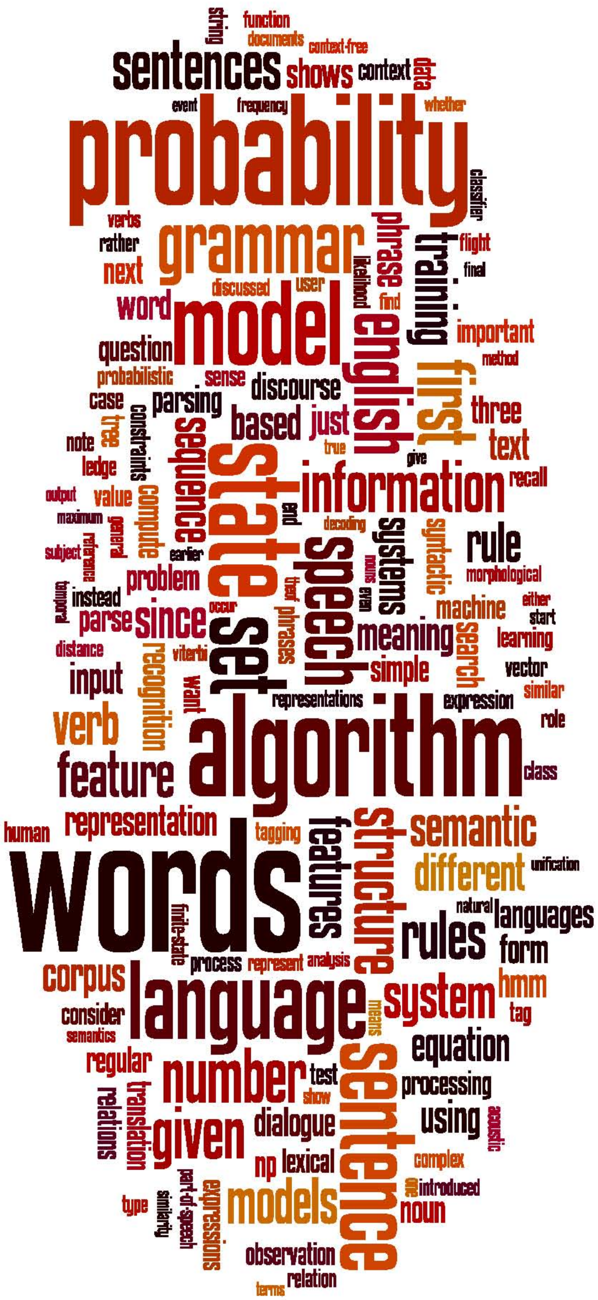
Recompute Alignment Probabilities P(A, F | E)

green house / casa verde	green house / casa verde	the house / la casa	the house / la casa
½ × ¼ =⅛	½ × ½ =¼	½ × ½ =¼	½ × ¼=⅛

Normalize to get P(A | F, E)

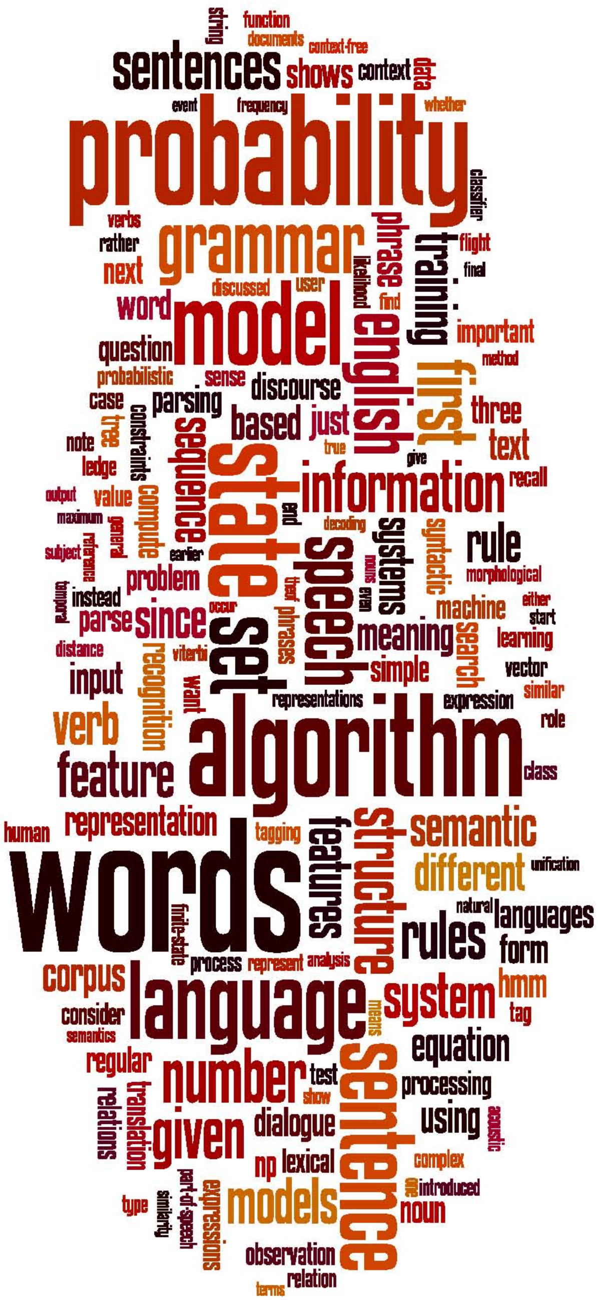
$\frac{1/8}{3/8} = \frac{1}{3}$ $\frac{1/4}{3/8} = \frac{2}{3}$ $\frac{1/4}{3/8} = \frac{2}{3}$ $\frac{1/8}{3/8} = \frac{1}{3}$

Continue EM iterations until translation parameters converge



Machine Translation

Learning Word Alignments in IBM Model 1



Machine Translation

Phrase Alignments and the Phrase Table

Dan Jurafsky

The Translation Phrase Table

Philipp Koehn's phrase translations for den Vorschlag
Learned from the Europarl corpus (this table is $\varphi(\bar{e}|f)$; normally we want $\varphi(f|\bar{e})$):

English	$\varphi(\bar{e}\|f)$	English	$\varphi(\bar{e}\|f)$
the proposal	0.6227	the suggestions	0.0114
's proposal	0.1068	the proposed	0.0114
a proposal	0.0341	the motion	0.0091
the idea	0.0250	the idea of	0.0091
this proposal	0.0227	the proposal	0.0068
proposal	0.0205	its proposal	0.0068
of the proposal	0.0159	it	0.0068
the proposals	0.0159	…	…

Learning the Translation Phrase Table

1. Get a **bitext** (a parallel corpus)
2. Align the sentences → E-F sentence pairs
3. Use IBM Model 1 to learn word alignments
   E→*F* and *F* →*E*
4. Symmetrize the alignments
5. Extract phrases
6. Assign scores

Step 1: Parallel corpora

- EuroParl: http://www.statmt.org/europarl/
  - A parallel corpus extracted from proceedings of the European Parliament.
  - Philipp Koehn. 2005. Europarl: A Parallel Corpus for Statistical Machine Translation. MT Summit
  - Around 50 million words per language for earlier members:
    - Danish, Dutch, English, Finnish, French, German, Greek, Italian, Portuguese, Spanish, Swedish
  - 5-12 million words per language for recent member languages
    - Bulgarian, Czech, Estonian, Hungarian, Latvian, Lithuanian, Polish, Romanian, Slovak, and Slovene
- LDC: http://www.ldc.upenn.edu/
  - Large amounts of parallel English-Chinese and English-Arabic text

Step 2: Sentence Alignment

Kevin Knight example

E-F Bitext

The old man is happy. He has fished many times. His wife talks to him. The fish are jumping. The sharks await.

El viejo está feliz porque ha pescado muchos veces. Su mujer habla con él. Los tiburones esperan.

Sentence Alignment Algorithm:

- Segment each text into sentences
- Extract a feature for each sentence
  - length in words or chars
  - number of overlapping words from some simpler MT model
- Use dynamic programming to find which sentences align

Dan Jurafsky

Sentence Alignment: Segment sentences

1. The old man is happy.
2. He has fished many times.
3. His wife talks to him.
4. The fish are jumping.
5. The sharks await.

1. El viejo está feliz porque ha pescado muchos veces.
2. Su mujer habla con él.
3. Los tiburones esperan.

Sentence Alignment: Features per sentence plus dynamic programming

1. The old man is happy.
2. He has fished many times.
3. His wife talks to him.
4. The fish are jumping.
5. The sharks await.

1. El viejo está feliz porque ha pescado muchos veces.
2. Su mujer habla con él.
3. Los tiburones esperan.

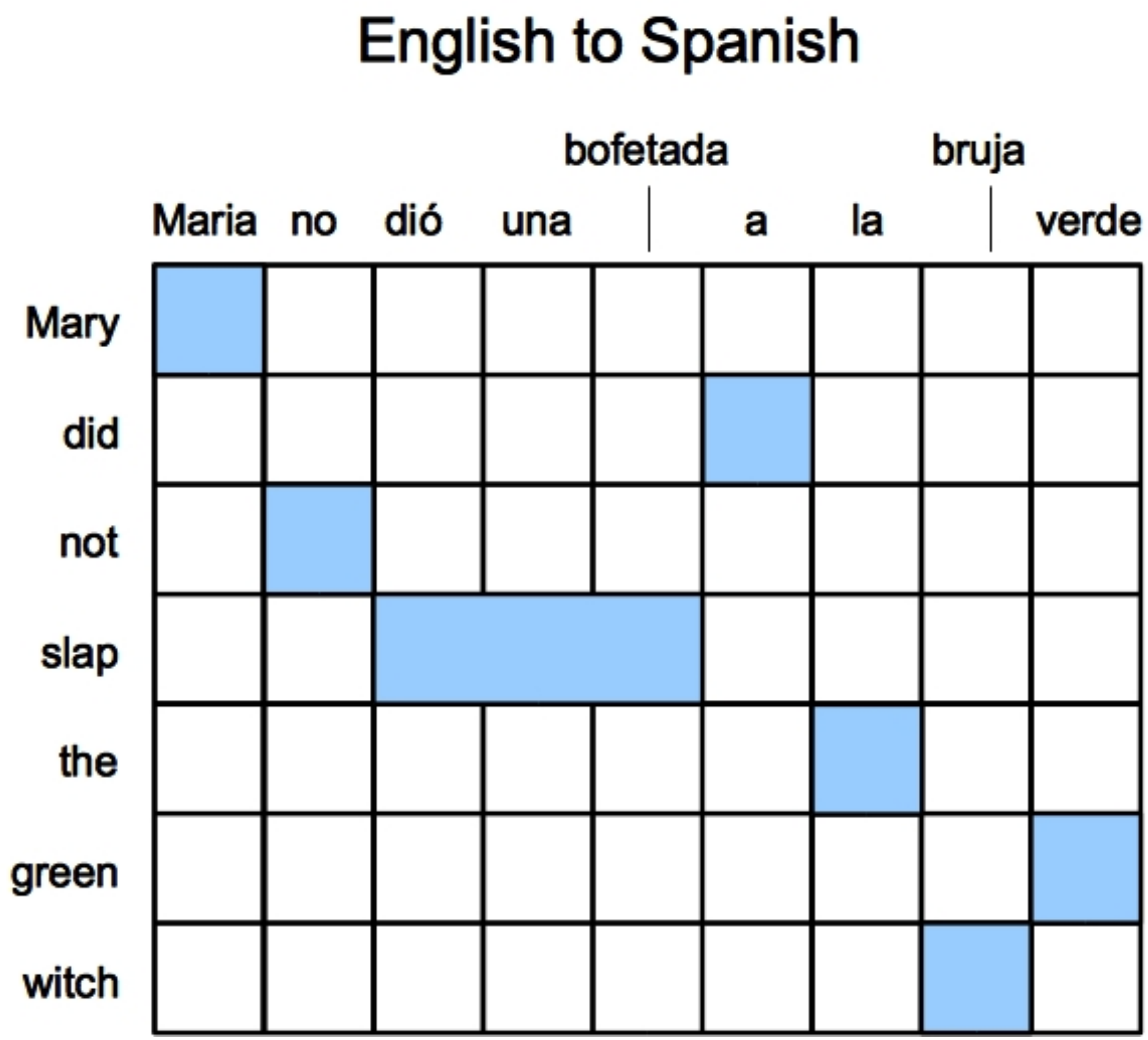
Sentence Alignment: Remove unaligned sentences

1. The old man is happy. He has fished many times.	El viejo está feliz porque ha pescado muchos veces.
2. His wife talks to him.	Su mujer habla con él.
3. The sharks await.	Los tiburones esperan.

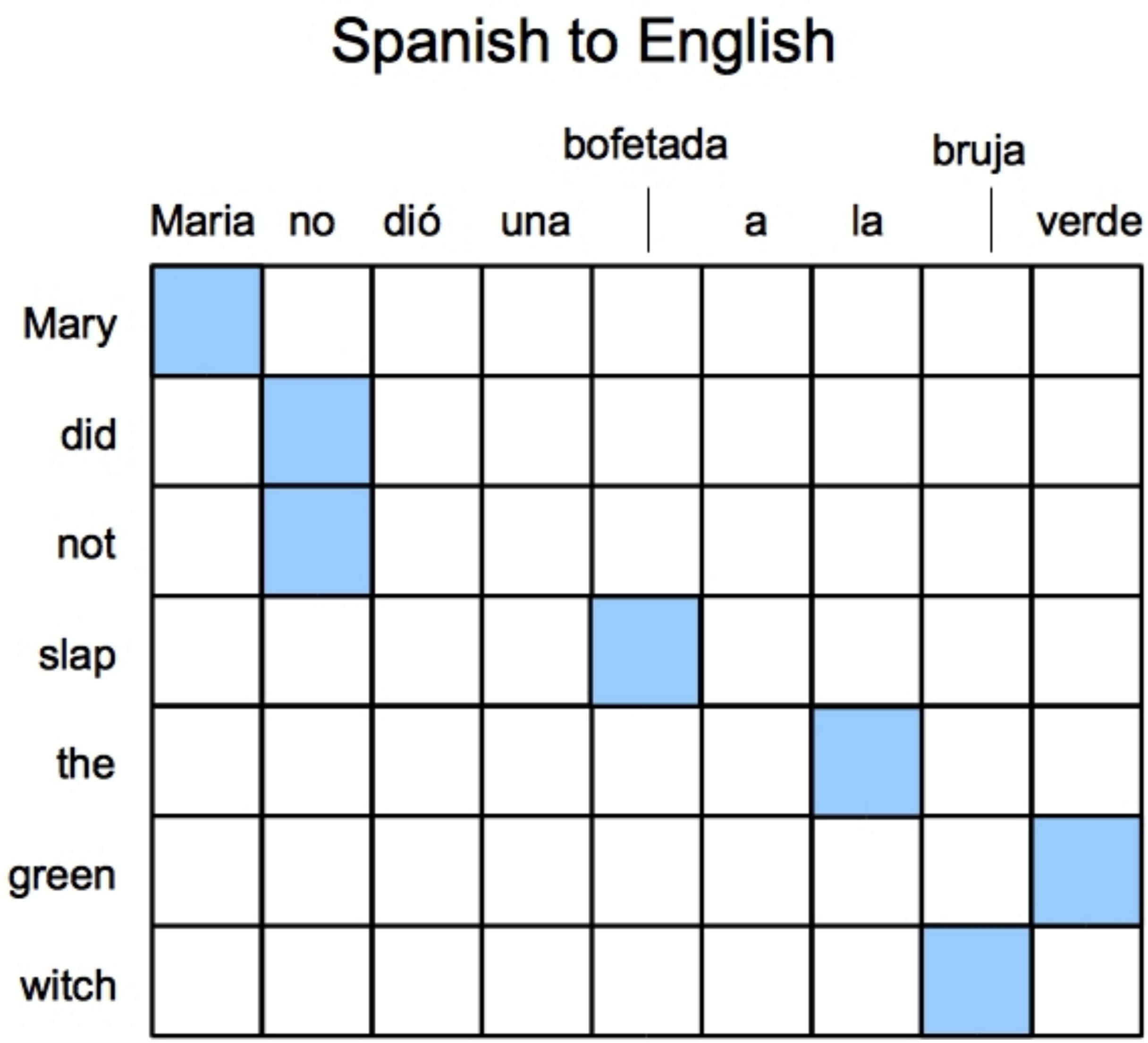
Steps 3 and 4: Creating Phrase Alignments from Word Alignments

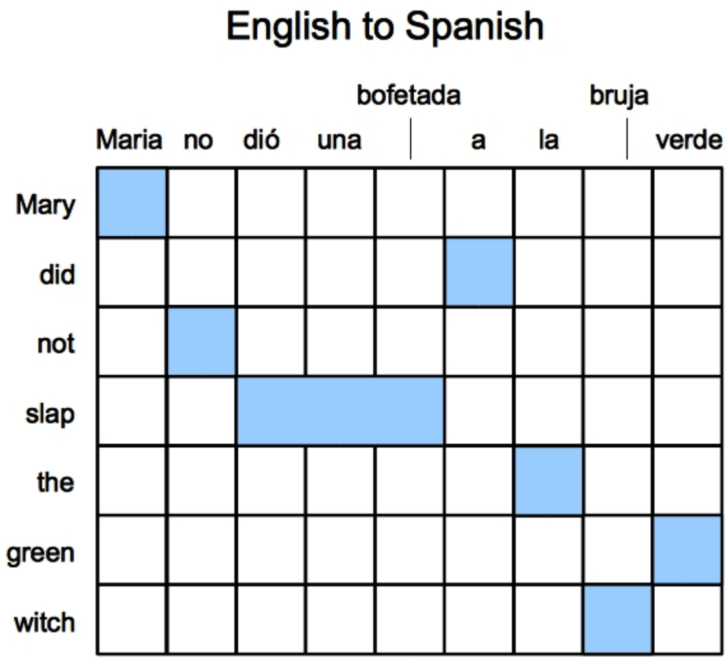
- Word alignments are one-to-many
- We need phrase alignment (many-to-many)
- To get phrase alignments:
  1) We first get word alignments for both $E \rightarrow F$ and $F \rightarrow E$
  2) Then we "symmetrize" the two word alignments into one set of phrase alignments

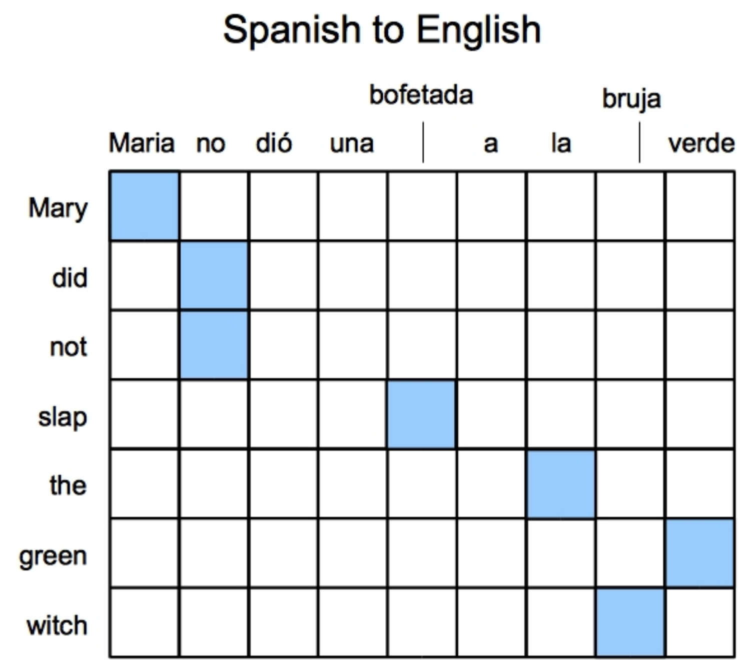
Regular 1-to-many alignment



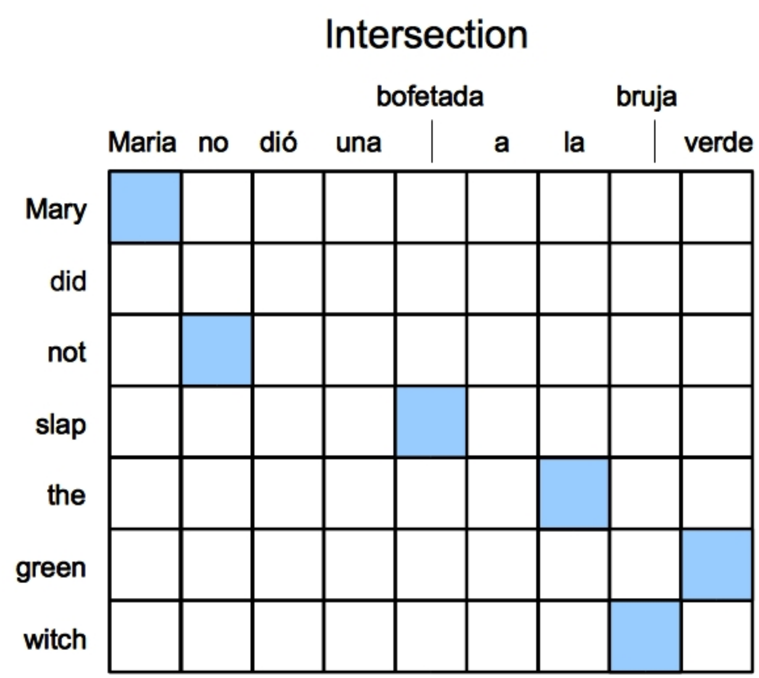
Alignment from IBM Model 1 run on reverse pairing





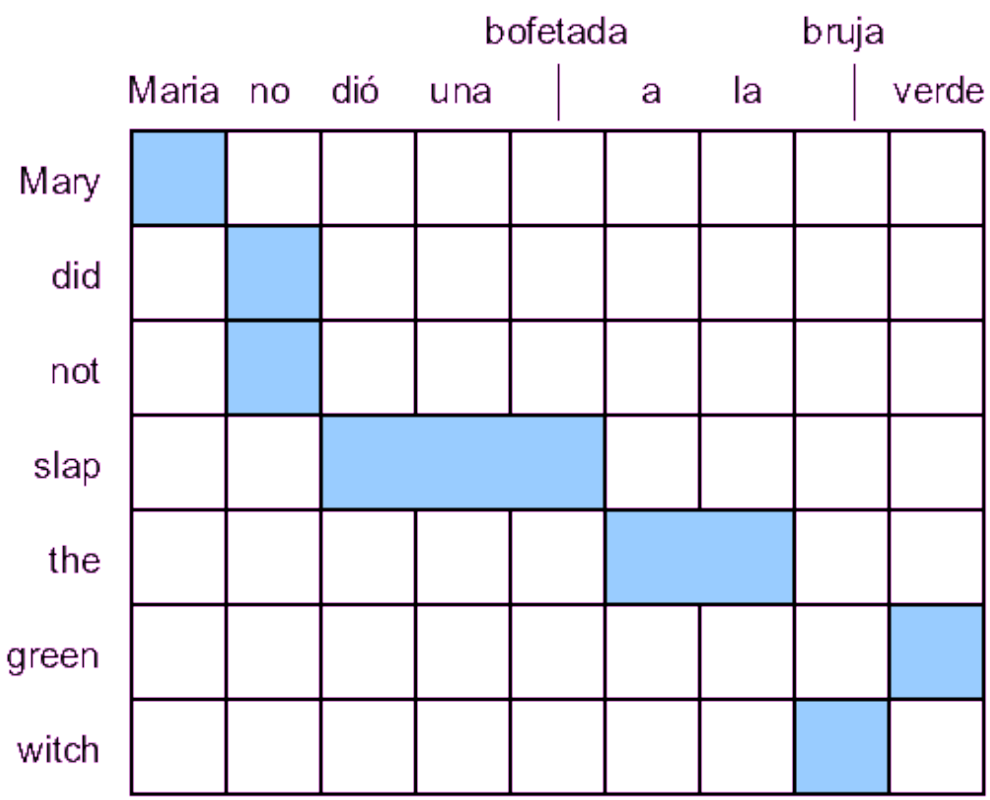


- Compute intersection
- Then use heuristics to add points from the union
- Philipp Koehn 2003. Noun Phrase Translation



Step 5. Extracting phrases from the resulting phrase aligment

Extract all phrases that are **consistent** with the word alignment



(Maria, Mary),

(no, did not),

(slap, dió una bofetada),

(verde, green),

(a la, the)

(Maria no, Mary did not),

(no dió una bofetada, did not slap),

(dió una bofetada a la, slap the),

(bruja verde, green witch),

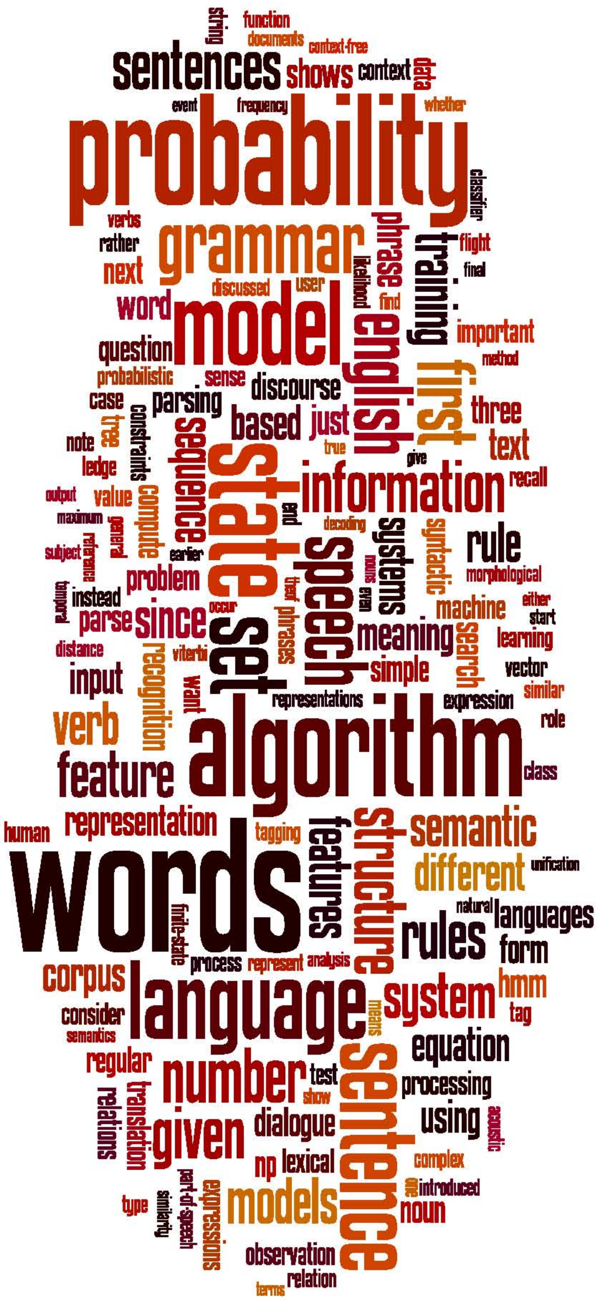
(a la bruja verde, the green witch)

…

Final step: The Translation Phrase Table

- Goal: A phrase table:
  - A set of phrases: $\bar{f}, \bar{e}$
  - With a weight for each: $\varphi(\bar{f}, \bar{e})$
- Algorithm
  - Given the phrase aligned bitext
  - And all extracted phrases
  - MLE estimate of ϕ: just count and divide

$$\phi(\bar{f}, \bar{e}) = \frac{\text{count}(\bar{f}, \bar{e})}{\sum_{f} \text{count}(\bar{f}, \bar{e})}$$



Machine Translation

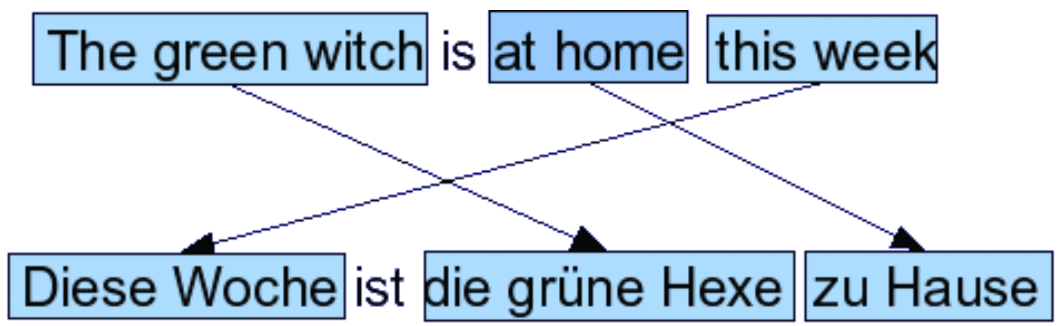
Phrase Alignments and the Phrase Table

Machine Translation

Phrase-Based Translation

Phrase-Based Translation

(Koehn et al. 2003)



- Remember the noisy channel model is backwards:
  - We translate German to English by pretending an English sentence generated a German sentence
    - Generative model gives us our probability P(F|E)
  - Given a German sentence, find the English sentence that generated it.

Dan Jurafsky

Three Components of Phrase-based MT

- P(F|E) Translation model
- P(E): Language model
- Decoder: finding the sentence E
  that maximizes P(F|E)P(E)

Dan Jurafsky

The Translation Model P(F|E)
Generative Model for Phrase-Based Translation

P(*F* | *E*): Given English phrases in *E*, generate Spanish phrases in *F*.

1. Group *E* into phrases $\bar{e}_1, \bar{e}_2, \ldots, \bar{e}_I$
2. Translate each phrase $\bar{e}_i$, into $\bar{f}_i$, based on ***translation probability*** $\phi(\bar{f}_i \mid \bar{e}_i)$
3. Reorder each Spanish phrase $\bar{f}_i$ based on its ***distortion probability*** *d*.

$$P(F \mid E) = \prod_{i=1}^{I} \varphi(\bar{f}_i, \bar{e}_i) d(start_i - end_{i-1} - 1)$$

Dan Jurafsky

Distortion Probability: Distance-based reordering

- Reordering distance: how many words were skipped (either forward or backward) when generating the next foreign word.
  - $start_i$: the word index of the first word of the foreign phrase that translates the *i*th English phrase
  - end_i: the word index of the last word of the foreign phrase that translates the *i*th English phrase
  - Reordering distance = $start_i - end_{i-1} - 1$
- What is the probability that a phrase in the English sentence skips over x Spanish words in the Spanish sentence?
- Two words in sequence: $start_i = end_{i-1} + 1$, so distance=0.
- How are d probabilities computed?
  - Exponentially decaying cost function:

$$d(x) = \alpha^{|x|}$$

  - Where $\alpha \in [0,1]$

Dan Jurafsky

Sample Translation Model

Position	1	2	3	4	5	6
English	Mary	did not	slap	the	green	witch
Spanish	Maria	no	dió una bofetada a	la	bruja	verde

$\text{start}_i - \text{end}_{i-1}$-1	0	0	0	0	1	-2

$$p(F \mid E) = \varphi(\text{Maria,Mary})\alpha^0 \varphi(\text{no, did not})\alpha^0 \varphi(\text{dio una bofetada a, slap})\alpha^0$$
$$\varphi(\text{la,the})\alpha^0 \varphi(\text{verde,green})\alpha^1 \varphi(\text{bruja,witch})\alpha^2$$

The goal of the decoder

- The best English sentence

$$\hat{E} = \mathrm{argmax}_E\, P(E \mid F)$$

- The Viterbi approximation to the best English sentence:

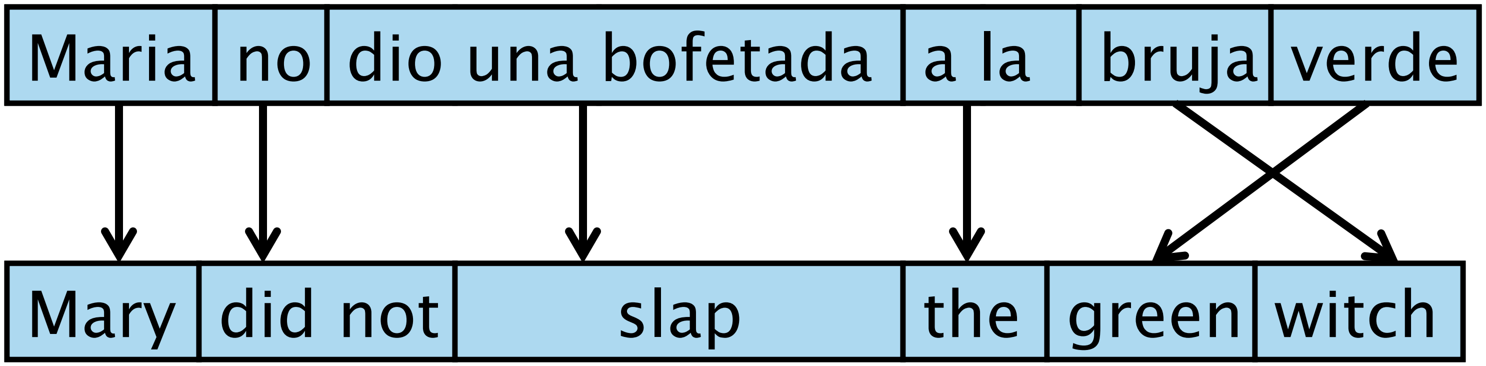
$$(A, E) = \mathrm{argmax}_{(A,E)}\, P(A, E \mid F)$$

- Search through the space of all English sentences

Dan Jurafsky

Phrase-based Decoding

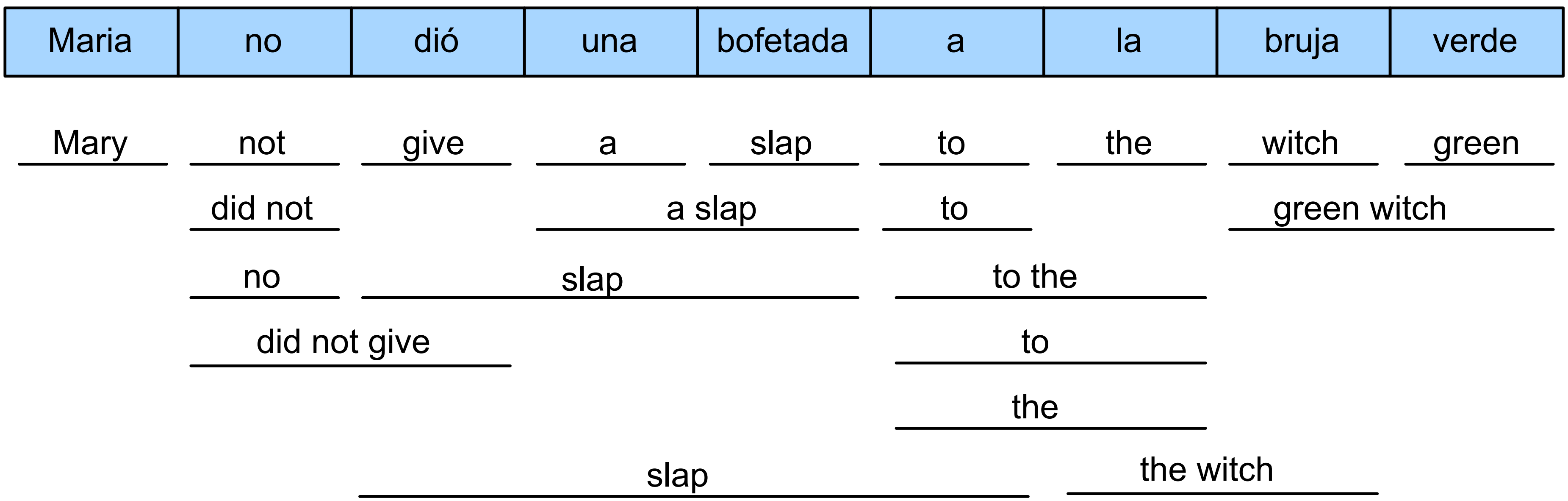
Slide adapted from Philipp Koehn



- Build translation left to right
  - *Select foreign word* to be translated
  - *Find English phrase* translation
  - *Add English phrase* to end of partial translation
  - *Mark foreign words* as translated

Dan Jurafsky

Decoding: The lattice of possible English translations for phrases



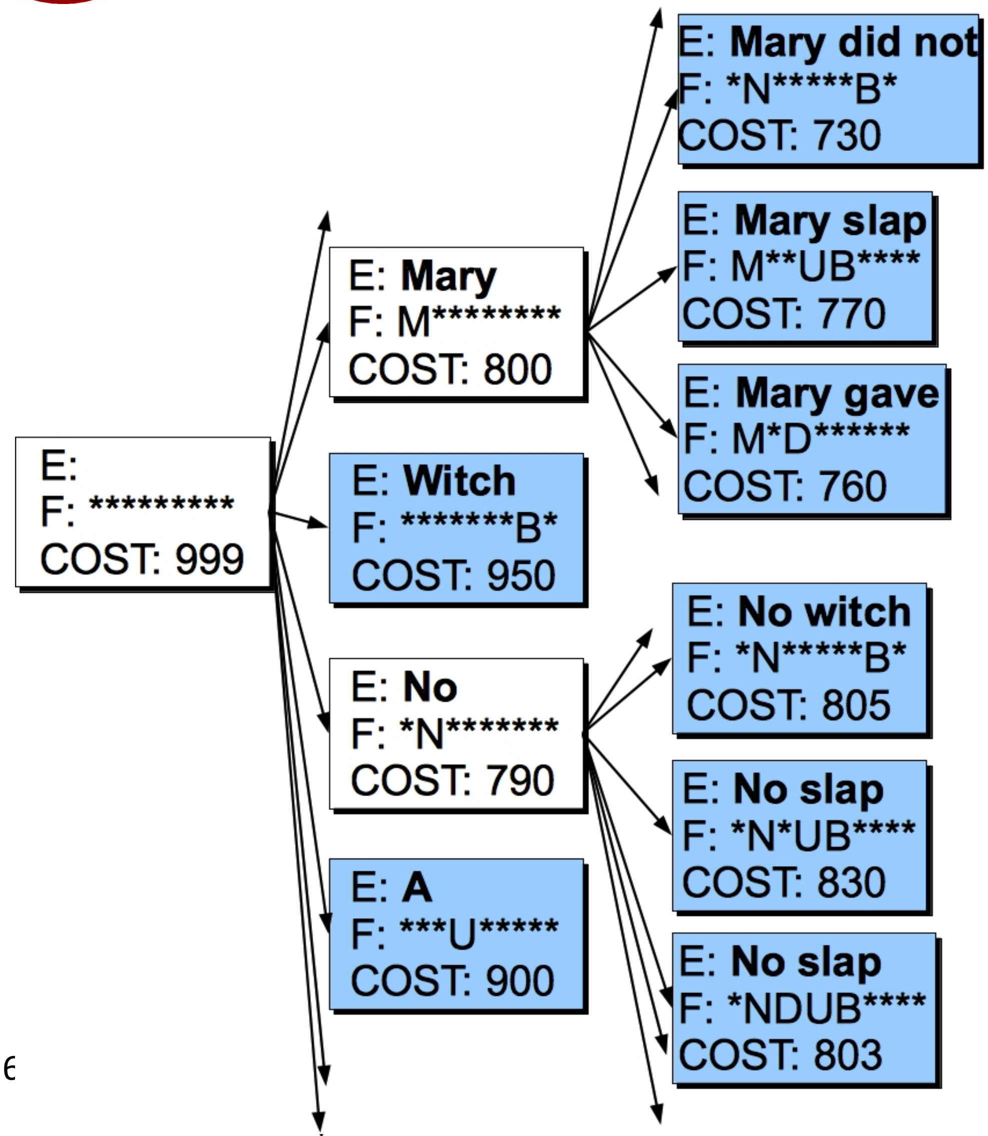
Decoding

- Stack decoding
- Maintaining a stack of hypotheses.
  - Actually a priority queue
  - Actually a set of different priority queues
- Iteratively pop off the best-scoring hypothesis, expand it, put back on stack.
- The score for each hypothesis:
  - Score so far

$$COST(hyp(S(E,F))) = \prod_{i \in S} \varphi(\bar{f}_i, \bar{e}_i) d(start_i - end_{i-1} - 1) P(E)$$

  - Estimate of future costs

Decoding by hypothesis expansion



Maria no dio una bofetada…

- After expanding NULL
- After expanding No

    E: No slap

    F: *N*UB***

    COST: 803

- After expanding Mary

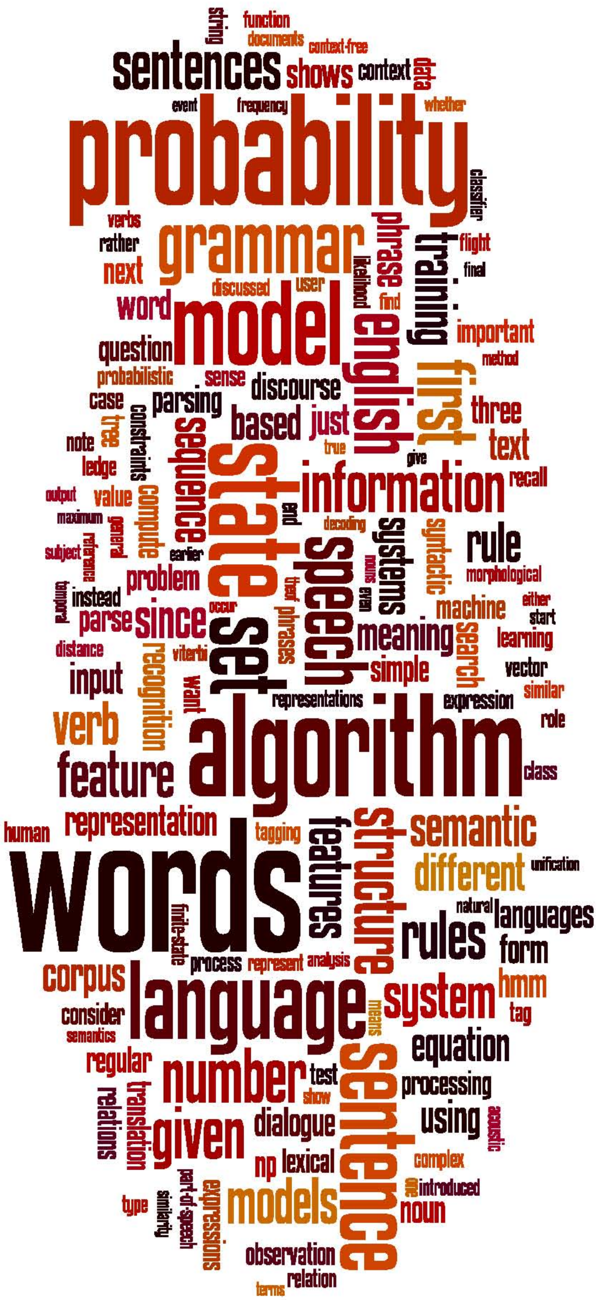
Efficiency

- The space of possible translations is huge!
- Even if we have the right n words, there are n! permutations
- We need to find the best scoring permutation
    - Finding the argmax with an n-gram language model is NP-complete [Knight 1999]

- Two standard ways to make the search more efficient
    - Pruning the search space
    - Recombining similar hypotheses

Machine Translation

Phrase-Based Translation



Machine Translation

Evaluating MT: BLEU

Evaluating MT: Using human evaluators

- **Fluency**: How intelligible, clear, readable, or natural in the target language is the translation?
- **Fidelity**: Does the translation have the same meaning as the source?
  - **Adequacy**: Does the translation convey the same information as source?
    - Bilingual judges given source and target language, assign a score
      - Monolingual judges given reference translation and MT result.
  - **Informativeness**: Does the translation convey enough information as the source to perform a task?
    - What % of questions can monolingual judges answer correctly about the source sentence given only the translation.

Automatic Evaluation of MT

George A. Miller and J. G. Beebe-Center. 1958. Some Psychological Methods for Evaluating the Quality of Translations. Mechanical Translation 3:73-80.

- Human evaluation is expensive and very slow
- Need an evaluation metric that takes seconds, not months
- Intuition: MT is good if it looks like a human translation

1. Collect one or more human ***reference translations*** of the source.
2. Score MT output based on its similarity to the reference translations.
   - BLEU
   - NIST
   - TER
   - METEOR

BLEU (Bilingual Evaluation Understudy)

Kishore Papineni, Salim Roukos, Todd Ward and Wei-Jing Zhu. 2002. BLEU: A method for automatic evaluation of machine translation. Proceedings of ACL 2002.

- "n-gram precision"
- Ratio of correct n-grams to the total number of output n-grams
  - Correct: Number of *n*-grams (unigram, bigram, etc.) the MT output shares with the reference translations.
  - Total: Number of *n*-grams in the MT result.
- The higher the precision, the better the translation
- Recall is ignored

Multiple Reference Translations

Slide from Bonnie Dorr

Reference translation 1:
The U.S. island of Guam is maintaining a high state of alert after the Guam airport and its offices both received an e-mail from someone calling himself the Saudi Arabian Osama bin Laden and threatening a biological/chemical attack against public places such as the airport.

Reference translation 2:
Guam International Airport and its offices are maintaining a high state of alert after receiving an e-mail that was from a person claiming to be the wealthy Saudi Arabian businessman Bin Laden and that threatened to launch a biological and chemical attack on the airport and other public places .

Machine translation:
The American [?] international airport and its the office all receives one calls self the sand Arab rich business [?] and so on electronic mail , which sends out ; The threat will be able after public place and so on the airport to start the biochemistry attack , [?] highly alerts after the maintenance.

Reference translation 3:
The US International Airport of Guam and its office has received an email from a self-claimed Arabian millionaire named Laden , which threatens to launch a biochemical attack on such public places as airport . Guam authority has been on alert .

Reference translation 4:
US Guam International Airport and its office received an email from Mr. Bin Laden and other rich businessman from Saudi Arabia . They said there would be biochemistry air raid to Guam Airport and other public places . Guam needs to be in high precaution about this matter .

Computing BLEU: Unigram precision

Slides from Ray Mooney

Cand 1: Mary no slap the witch green
Cand 2: Mary did not give a smack to a green witch.

Ref 1: Mary did not slap the green witch.
Ref 2: Mary did not smack the green witch.
Ref 3: Mary did not hit a green sorceress.

Candidate 1 Unigram Precision: 5/6

Computing BLEU: Bigram Precision

Cand 1: Mary no slap the witch green.
Cand 2: Mary did not give a smack to a green witch.

Ref 1: Mary did not slap the green witch.
Ref 2: Mary did not smack the green witch.
Ref 3: Mary did not hit a green sorceress.

Candidate 1 Bigram Precision: 1/5

Computing BLEU: Unigram Precision

Cand 1: Mary no slap the witch green.
Cand 2: Mary did not give a smack to a green witch.

Ref 1: Mary did not slap the green witch.
Ref 2: Mary did not smack the green witch.
Ref 3: Mary did not hit a green sorceress.

Clip the count of each *n*-gram
to the maximum count of the *n*-gram in any single reference

Candidate 2 Unigram Precision: 7/10

Dan Jurafsky

Computing BLEU: Bigram Precision

Cand 1: Mary no slap the witch green.
Cand 2: Mary did not give a smack to a green witch.

Ref 1: Mary did not slap the green witch.
Ref 2: Mary did not smack the green witch.
Ref 3: Mary did not hit a green sorceress.

Candidate 2 Bigram Precision: 4/9

Brevity Penalty

- BLEU is precision-based: no penalty for dropping words
- Instead, we use a brevity penalty for translations that are shorter than the reference translations.

$$\text{brevity-penalty} = \min\left(1\ ,\ \frac{\text{output-length}}{\text{reference-length}}\right)$$

Computing BLEU

- Precision$_1$, precision$_2$, etc., are computed over all candidate sentences C in the test set

$$\text{precision}_n = \frac{\sum_{C \in corpus} \sum_{\text{n-gram} \in C} \text{count-in-reference}_{\text{clip}}(\text{n-gram})}{\sum_{C \in corpus} \sum_{\text{n-gram} \in C} \text{count}(\text{n-gram})}$$

$$\text{BLEU-4} = \min\left(1, \frac{\text{output-length}}{\text{reference-length}}\right) \prod_{i=1}^{4} \text{precision}_i$$

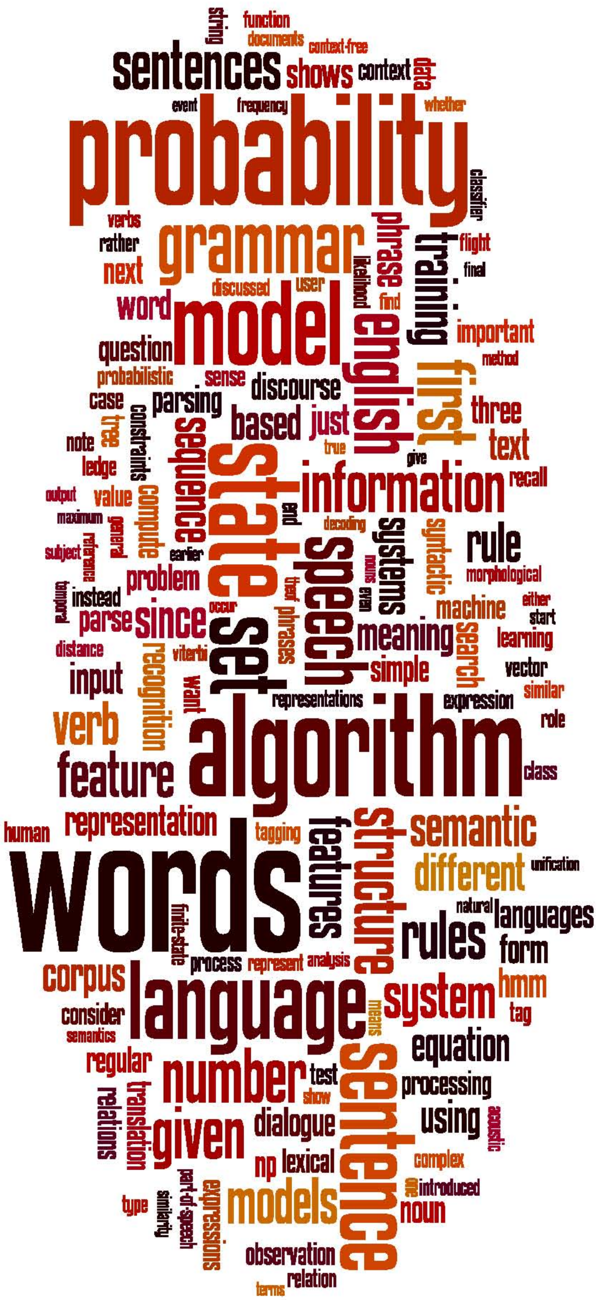
BLEU-2:

Candidate 1: Mary no slap the witch green.
Best Reference: Mary did not slap the green witch.

$$\frac{6}{7} \times \frac{5}{6} \times \frac{1}{5} = .14$$

Candidate 2: Mary did not give a smack to a green witch.
Best Reference: Mary did not smack the green witch.

$$\frac{7}{10} \times \frac{4}{9} = .31$$



Machine Translation

Evaluating MT: BLEU

Machine Translation

Loglinear Models for MT

The noisy channel is a special case

- The noisy channel model:

$$P_{LM} \times P_{TM}$$

- Adding distortion:

$$P_{LM} \times P_{TM} \times P_D$$

- Adding weights:

$$P_{LM}^{\lambda_1} \times P_{TM}^{\lambda_2} \times P_D^{\lambda_3}$$

- Many factors:

$$\prod_i P_i^{\lambda_i}$$

- In log space:

$$\log \prod_i P_i^{\lambda_i} = \sum_i \lambda_i \log P_i$$

It's a log-linear model!!

Knowledge sources for log-linear models

- language model
- distortion model
- P(F|E) translation model
- P(E|F) reverse translation model
- word translation model
- word penalty
- phrase penalty

Dan Jurafsky

Learning feature weights for MT

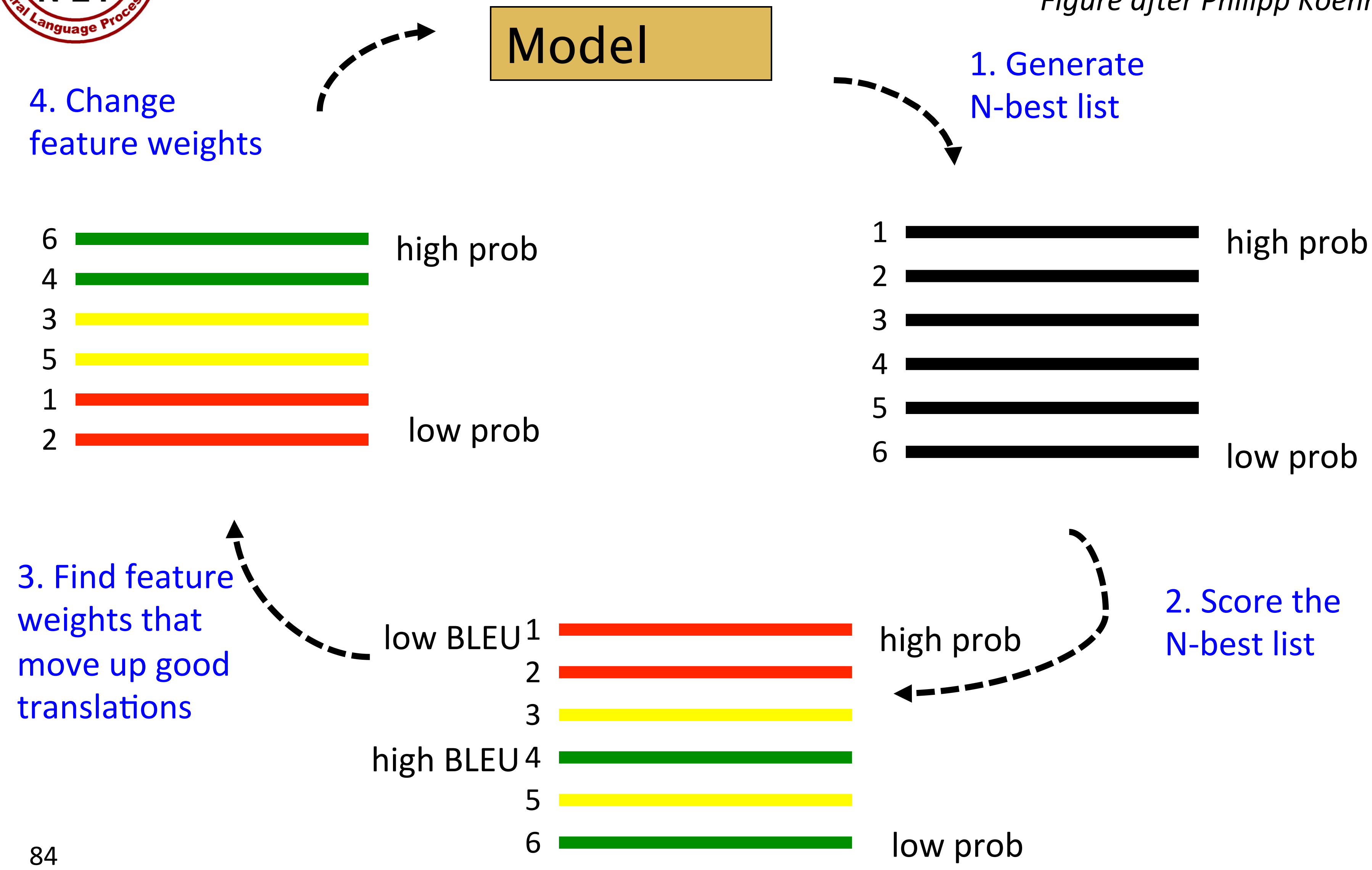
- Goal: choose weights that give optimal performance on a dev set.

Discriminative Training Algorithm:

- Translate the dev set, giving an N-best list of translations
  - Each translation has a model score (a probability)
  - Each translation has a BLEU score (how good is it)
- Adjust feature weights so the translation with the best BLEU gets better model score

Discriminative Training for MT

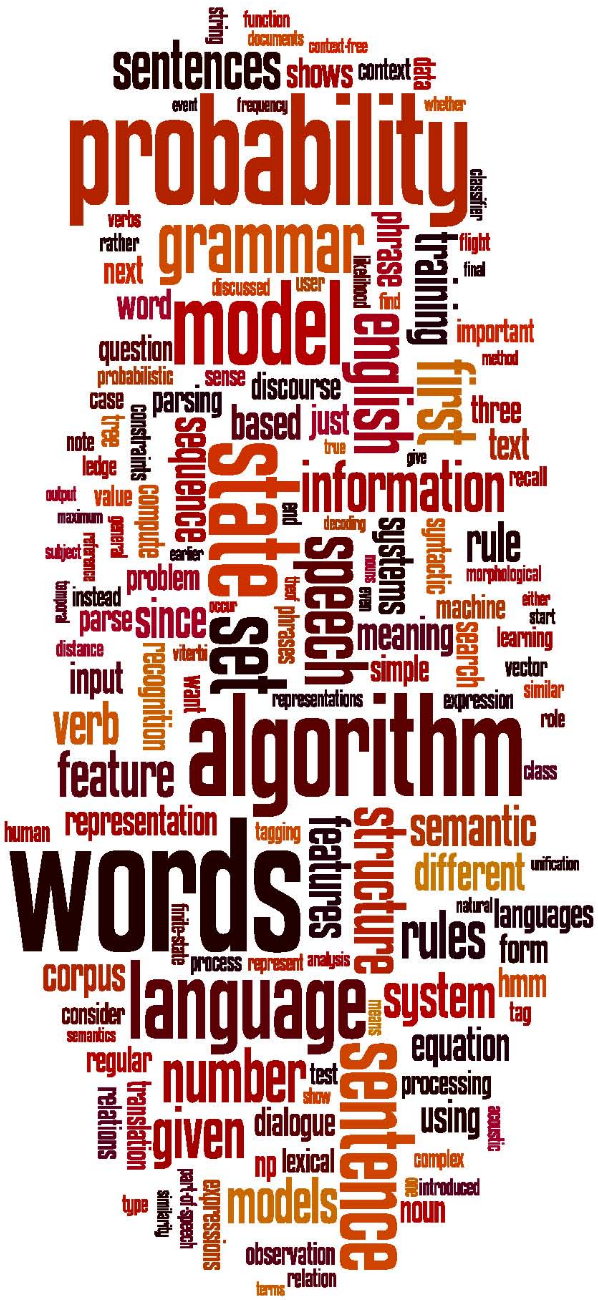
Figure after Philipp Koehn



How to find the optimal feature weights

Franz Joseph Och. 2003. Minimum error rate training in statistical machine translation. ACL 2003

- MERT (“Minimum Error Rate Training)
- MIRA
- Log-linear models
- etc.



Machine Translation

Loglinear Models for MT