CSCI 3907/6907

Fall 2019

Lecture 7

Machine Translation

Slides based on: Philipp Koehn's "Statistical Machine Translation" - Jurafsy & Martin's "Speech and Language Processing"

Machine Translation (MT)

- The use of computers to automate some of the process of translating from one language to another

- Fully-automated machine translation is still not possible
  - Rough translation can be helpful

- Current MT systems can be used to speed-up human translation, where MT output is fixed in **post-editing**

Machine Translation (MT)

- But possible in limited **sublanguage** domains
    - Limited vocabulary and few basic phrase types.
    - Ambiguity can be resolved using local context.
    - **Examples**:
        - Weather forecasting: *Cloudy with a chance of showers today and Thursday*
        - Air travel queries: book my flight to Washington DC
        - Restaurant recommendation: Washington DC has the best seafood restaurants

Challenges in MT

- Translation is a difficult and creative endeavor
  - Typically requires a deep understanding of the source and target languages

- Different languages vary across multiple dimensions
  - syntax and word order, morphology, lexical differences, etc.

  Example: wsyktbh (Arabic) → and he will write it

Typology

- Refers to the study of systematic cross-linguistic similarities and differences between languages

- Typological differences between languages can cause problems for translation

- Dimensions include morphology and syntax

Morphological Variation

- **Morpheme**: "Minimal meaningful unit of language"

  `Word = Morpheme + Morpheme + Morpheme +…`

- Number of morphemes per word:
  - **Isolating** languages: typically one morpheme per word (e.g. Vietnamese, Cantonese)
  - **Polysynthetic** languages: many morphemes per word, can correspond to whole English sentences (e.g. Siberian Yupik ("Eskimo"))

- The degree to which morphemes are segmentable (e.g. agglutinative and fusion languages)
  - Agglutinative (e.g. Turkish word evlerinizden "from your houses" consists of ev-ler-iniz-den "house-plural-your-from") (source: wikipedia)
  - -om in the word stol**om** in Russian language

Syntactic Variation: Basic Word Orders

- SVO (Subject-Verb-Object) languages: English, German, French, Mandarin

  I baked a pizza

- SOV Languages: Japanese, Hindi

  English: He adores listening to music

  Japanese: kare ha ongaku wo kiku no ga daisuki desu

  he music to listening adores

- VSO languages: Irish, Classical Arabic, Tagalog

  English: Jack is writing a story

  Arabic: yktb jAk Alqsh

  is writing Jack a story

Lexical Divergences

- Homonyms → words with the same spelling but different senses.
  - English: bass (fish) vs. bass (instrument) → *lubina* or *bajo* in Spanish
  - Closely related to word sense disambiguation

- Polysemous words → related meanings

  English: I **know** he just bought a book

  French: Je sais qu'il vient d'acheter un livre

  English: I **know** John

  French: Je connais Jean

Lexical Divergences

- Lexical gap:
  - Sometimes no word or phrase can express the meaning of a word in the other language.
  - Example: Japanese does not have a word for ‘privacy’

- Part of Speech divergences:
  - English `She likes to sing`
  - German `Sie singt gerne` ***[She sings likefully]***
  - English `I’m hungry`
  - Spanish `Tengo hambre` ***[I have hunger]***

Lexical Specificity

- Grammatical specificity
  - Spanish: plural pronouns have gender (`ellos/ellas`)
  - English: plural pronouns no gender (`they`)

- So translating "they" from English to Spanish, we need to figure out gender of the referent!

Semantic Specificity

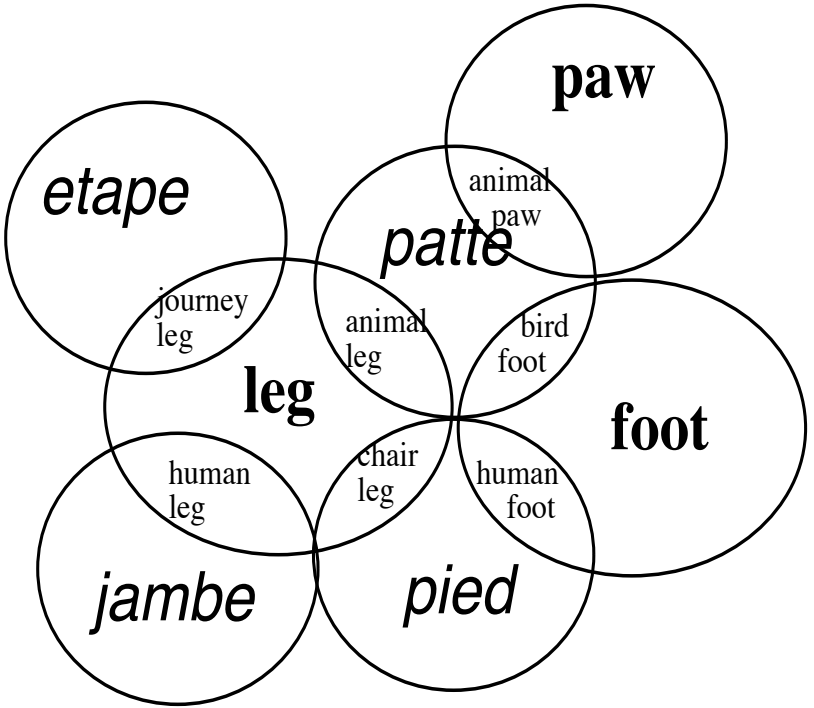
English `wall`
German `Wand` (inside) `Mauer` (outside)

English `fish`
Spanish `pez` (the creature) `pescado` (fish as food)

Cantonese `ngau`
English `cow` `beef`

Complex Lexical Overlap

- Lexical divergences can be more complex than one-to-many translations
  - example: many-to-many mapping between the English words **leg**, **food**, and **paw**, and their French translations.

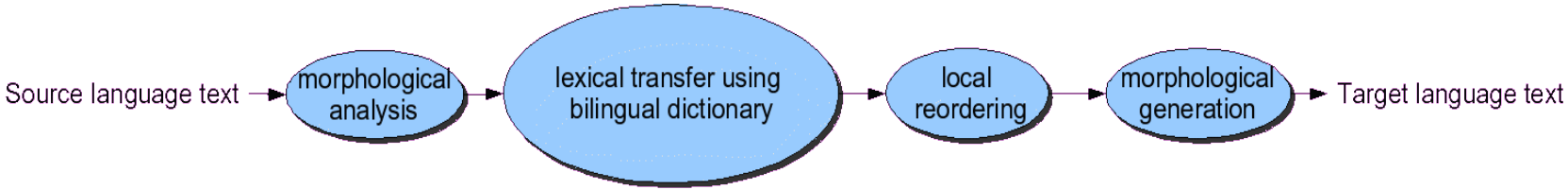


Approaches

- **Classic MT approaches:**
  - Direct Translation
  - Transfer Models
  - Interlingua

- **Statistical Machine Translation:**
  - Phrase-based Statistical MT
  - Neural MT **(later in this course)**

Direct Translation



- Proceed word-by-word through text, translating each word
- No intermediate structures except morphological analysis
- Based on a large bilingual dictionary
- After word translation, can do simple reordering
  - Adjective ordering English -> French/Spanish

Direct Translation – Example

English: Mary didn't slap the green witch

Spanish: Maria no dio una bofetada a la bruja verde

Mary not gave a slap to the witch green

Input → Mary didn't slap the green witch

(morphology) → Mary DO-PAST not **slap** the green witch

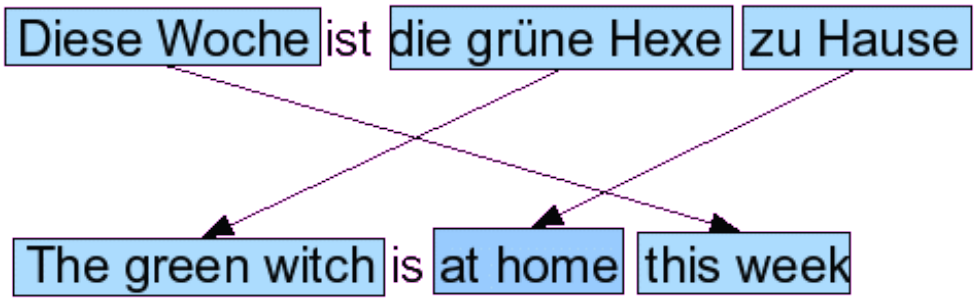
(Lexical Transfer) → Maria PAST no **dar una bofetada** a la **verde bruja**

(Local reordering) → Maria no dar PAST una bofetada a la **bruja verde**

(Morphology) → Maria no dio una bofetada a la bruja verde

Problems with direct MT

- Can only handle simple word reordering; does not have a parsing component
  - Cannot handle long-distance reordering
  - Example: German to English reordering often involves long-distance reordering of adverbs and subjects of tensed verbs

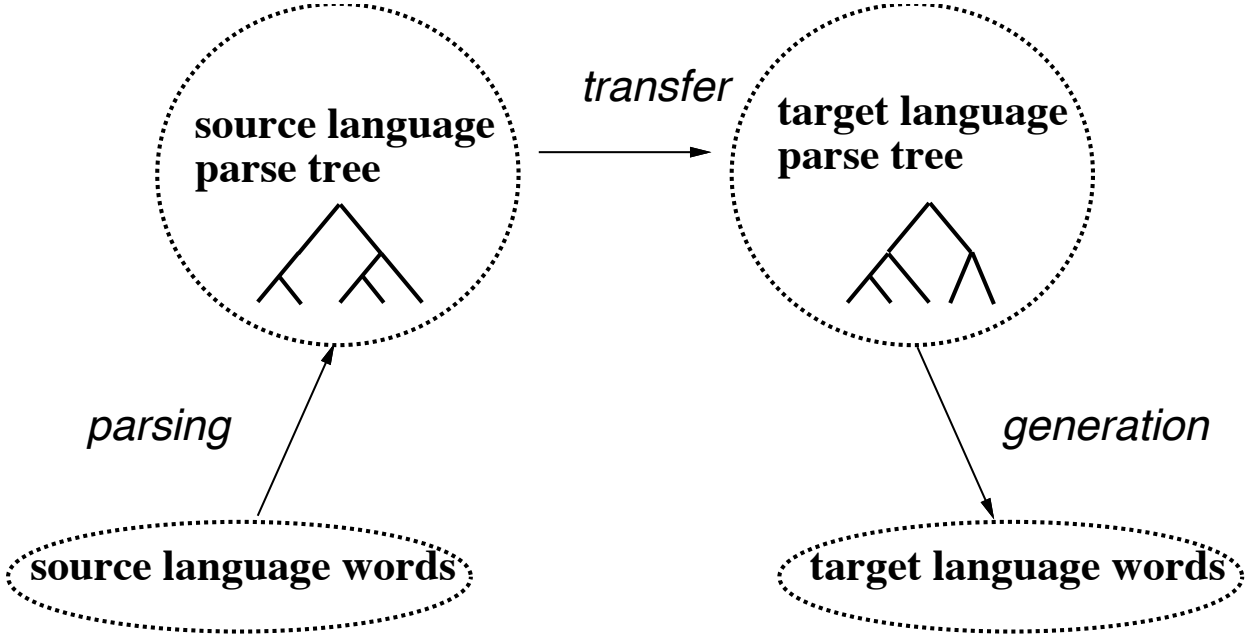


The Transfer Model

- Apply contrastive knowledge
  - knowledge about the difference between two languages

- Three Steps:
  - **Analysis**: Syntactically parse source language
  - **Transfer**: Rules to turn this parse into parse for target language
  - **Generation**: Generate target sentence from parse tree

The Transfer Model



Example: English to French

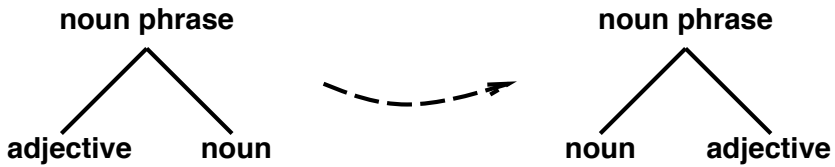
English: Adjective Noun

French: Noun Adjective

- This is not always true

  Route mauvaise ‘bad road, badly-paved road’

  Mauvaise route ‘wrong road’
- But is a reasonable first approximation
- Rule:



The Transfer Model

- Simple transfer rules are not sufficient; we need complex rules that combine lexical knowledge with syntactic and semantic features.
- **Analysis**:
  - Morphological analysis and parts-of-speech tagging
  - Shallow constituency and dependency parsing
- **Transfer**:
  - Translation of idioms
  - Word sense disambiguation
- **Generation**:
  - Lexical translation using a rich bilingual dictionary
  - Reordering
  - Morphological generation

Problems with Transfer Models

- Requires a distinct set of transfer rules for each pair of languages.
  - Expensive and inefficient

Interlingua

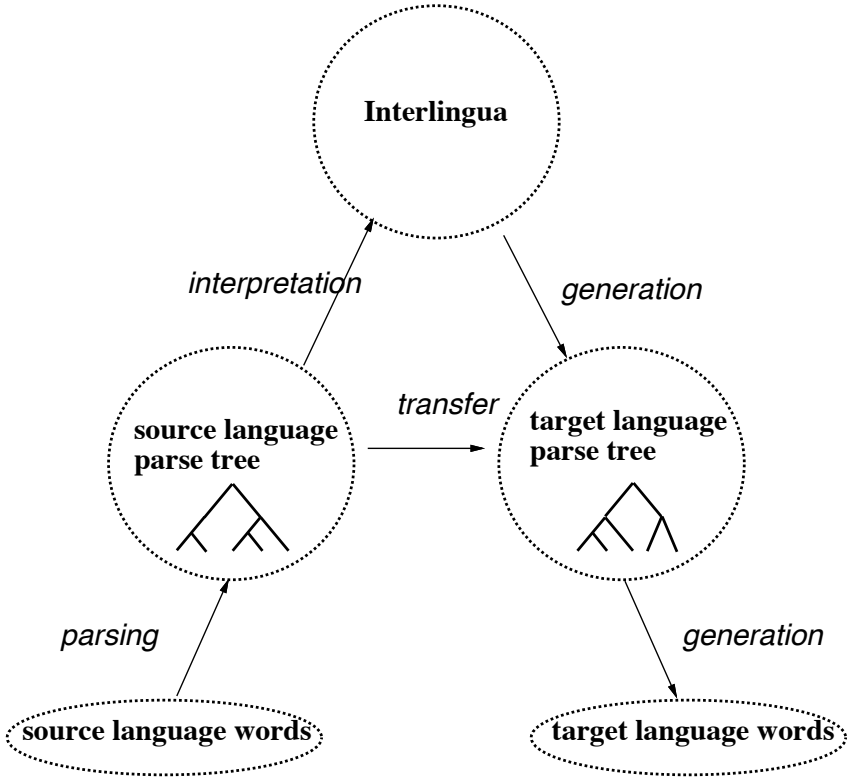
- Use meaning as a representation language
  (a meaning representation → interlingua)
  1. Parse source sentence into meaning representation
  2. Generate target sentence from meaning

- Represent all sentences that mean the same thing in the same way regardless of the language they happen to be in

Interlingua

- Use meaning as a representation language
  (a meaning representation → interlingua)
  1. Parse source sentence into meaning representation
  2. Generate target sentence from meaning

Interlingua



Problems with Interlingual Models

- Requires exhaustive semantic analysis and disambiguation

- Generally not possible except in relatively simple domains based on a database model (sublanguage domains)

Statistical “Data Driven” MT

- The intuition for Statistical MT comes from the **impossibility** of perfect translation
- Language translation is burdened with so many decisions that are hard to formalize
- It may be better to learn how to translate from past translation examples
- ***Given a text in the source language, what is the most probable translation in the target language?***

Data for Statistical MT

- **Parallel** text (or bitext) → collections of human translated text that is aligned at the sentence level

- Examples: Bible, UN and European Parliament Proceedings, Translated literature

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 and grammatical in the target

- Real translations trade off these two factors

Statistical Machine Translation

- Find the most likely target language sentence (e) given a foreign source sentence (f)

$$p(e|f)$$

$$\hat{e} = \arg\max_{e} \; p(e|f)$$

$$p(e|f) = \frac{p(e)p(f|e)}{p(f)}$$

$$\hat{e} = \arg\max_{e} \; p(e)p(f|e)$$

F → Maria no dio una bofetada a la bruja verde

E → English sentence 1
English sentence 2
.
.
.
etc

Statistical Machine Translation

$$\hat{e} = \arg\max_{e} \; p(e)p(f|e)$$

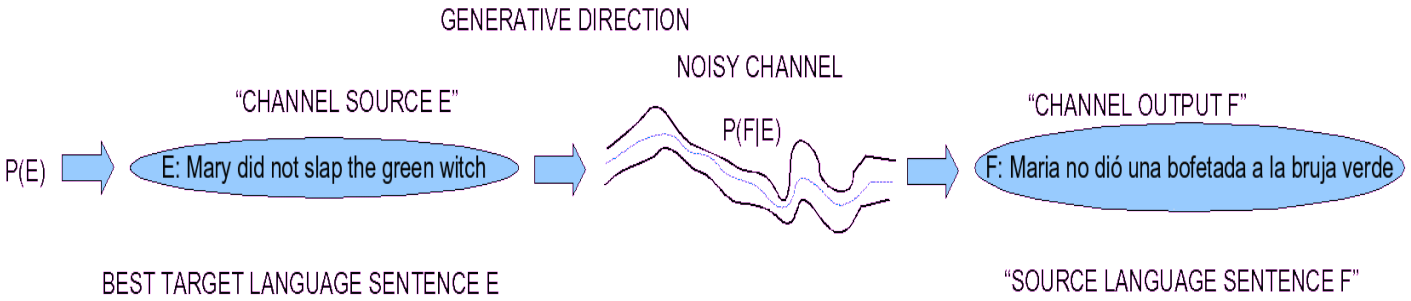
- p(e) is a language model
  - Estimates the **fluency** of target sentences

- P(f|e) is a translation model
  - Estimates the **adequacy or faithfulness** of the translation

Alternative View: The Noisy Channel Model

- Consider the foreign sentence as a *corrupted version* of some English sentence, and the objective of MT is to recover this hidden sentence

  – Thus the translation model is p*(f|e)*, the probability that the English sentence generated the foreign sentence



Fluency: 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

Faithfulness: Translation Model

- How to estimate *p(f|e)*
  - Maximum Likelihood estimate:

$$p(f|e) = \frac{count(f, e)}{count(e)}$$

  - **Impossible** to calculate for new sentences, so we need to introduce simplifying assumption

Phrase-Based Translation Model

- Use phrases (sequence of words) as well as single words as the fundamental units of translation

- The probability model of phrase based translation relies on a **translation probability** and a **distortion probability**
  - Distortion refers to the position of a phrase in a translated sentence

Phrase-Based Translation Model

- A generative model of translation
  1. Group *E* into phrases $\bar{e}_1, \bar{e}_2, \ldots, \bar{e}_I$
  2. Translate each phrase $\bar{e}_i$, into f_i, based on ***translation probability*** $p(f_i \mid \bar{e}_i)$
  3. Reorder each Foreign phrase f_i based on its ***distortion probability*** *d*.

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

  - We can use a simple distortion function that penalizes large distortions
  - The only parameters we need to estimate are the phrase translation probabilities p(f,e)

Example

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

P(F|E)=P(Maria|Mary)***d(1)***P(no| did not)***d(1)***p(dio una bofetada| slap)***d(1)***P(a la|the)***d(1)***p(bruja verde|green witch)***d(1)**

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

$start_i - end_{i-1} - 1$	0	0	0	0	1	-2

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

Parameter Estimation

- Using translation data, we can use **MLE estimates** of phrase translation probabilities:

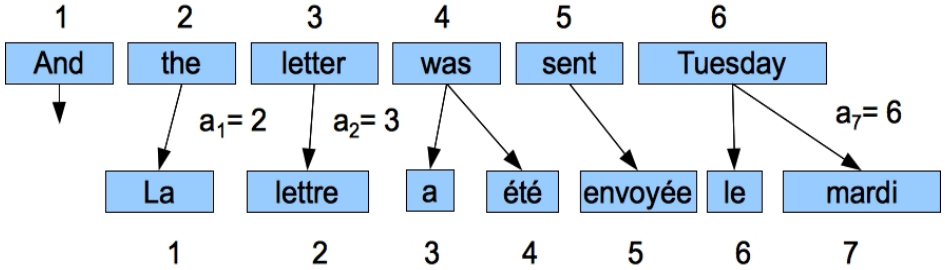
$$p(f|e) = \frac{count(f, e)}{count(e)}$$

  - But we don't have phrase-aligned training data; the parallel corpora are aligned at the sentence level

  - We can extract phrase alignments automatically by first aligning the words in the parallel sentences

Word Alignment

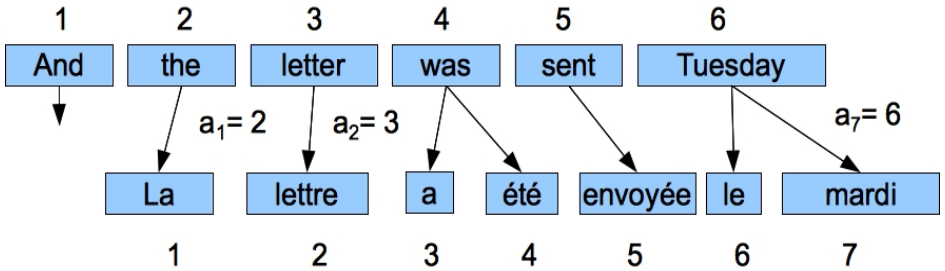
- A mapping between the source and target words in a set of parallel sentences



- Simplifying assumptions (for IBM Model 1):
  - **one-to-many** (not many-to-one or many-to-many)
    - each French word comes from exactly one English word

Word Alignment

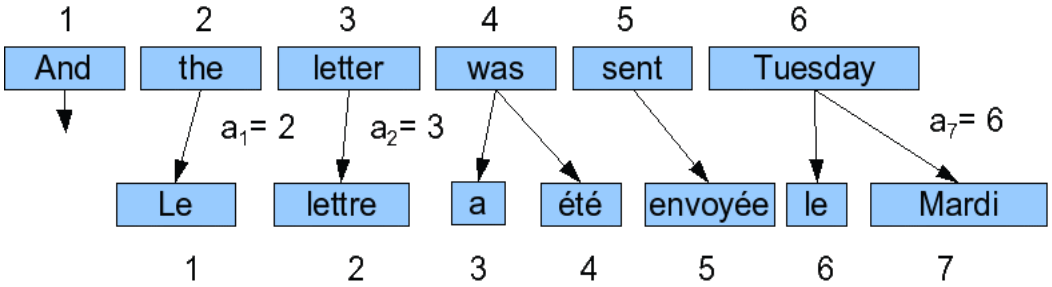
- An alignment is a vector of length J, one cell for each French word
  - The index of the English word that the French word comes from



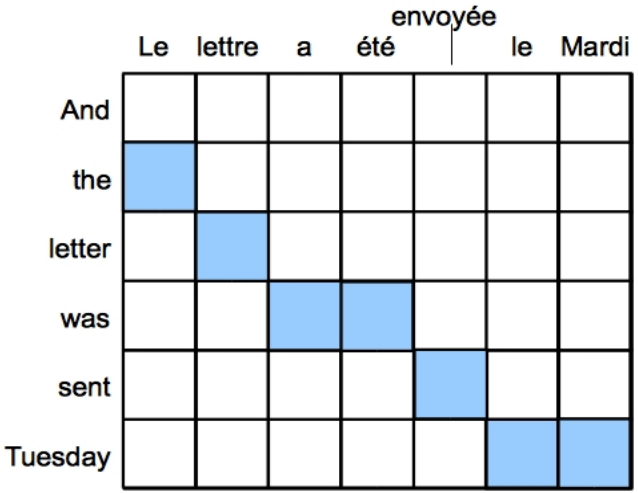
Alignment above is thus the vector A = [2, 3, 4, 4, 5, 6, 6]

- $a_1=2$, $a_2=3$, $a_3=4$, $a_4=4$…
- a_1→ the index in English that corresponds to the first word in the foreign language

Three representations of an alignment

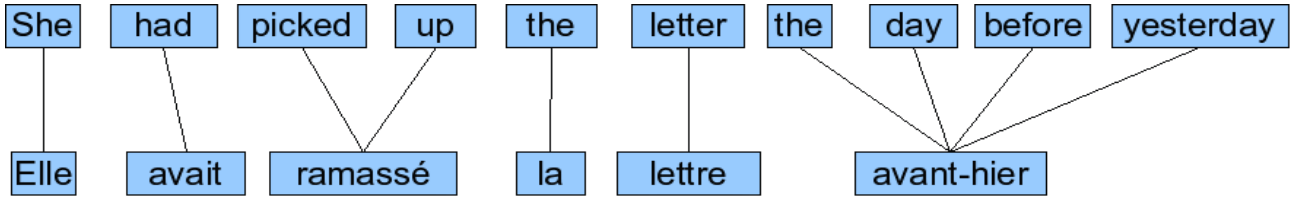


A = [2, 3, 4, 4, 5, 6, 6]

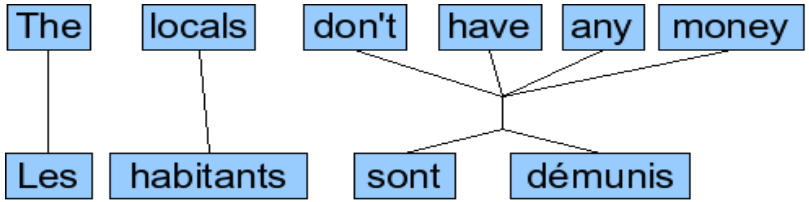


Alignments that don’t obey one-to-many restriction

- Many to one:



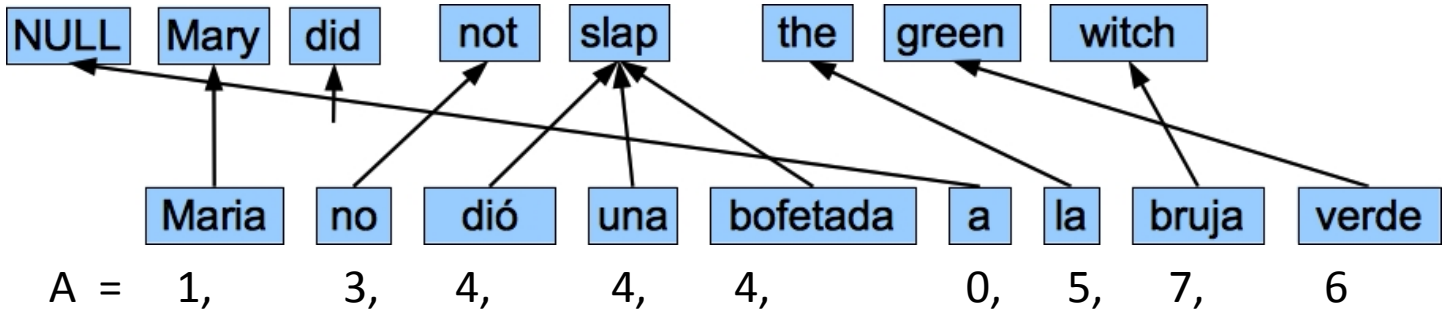
- Many to many:



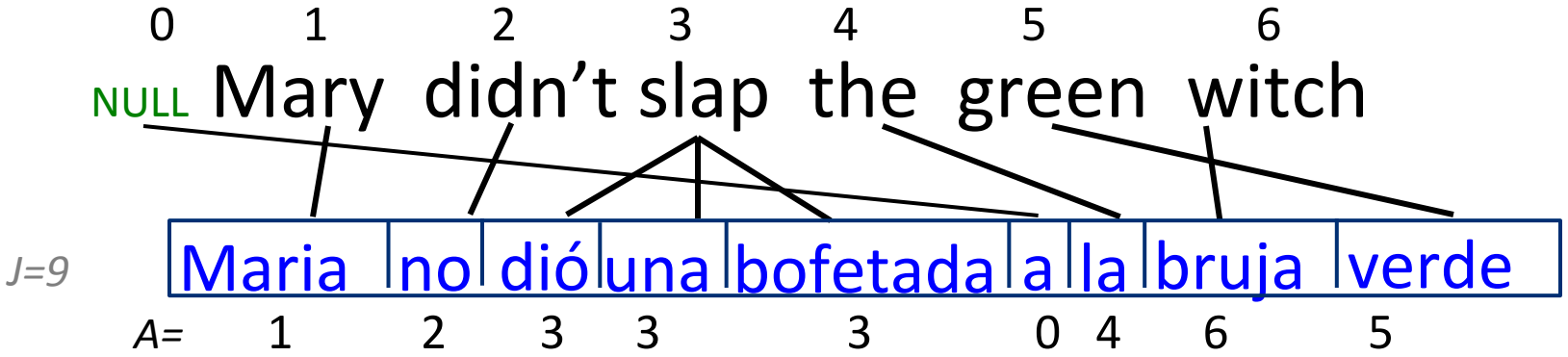
One addition: spurious words

- A word in the 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 :



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 $\mathrm{f_j}$

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

- If we knew the English source (E), the alignment A, and J (word length in the foreign source), then:

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

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

- IBM model 1 assumes each alignment is equally likely
  - each with probability (1/number of all possible alignments)
  - $(I+1)^J$ possible alignments

- With a normalization factor:

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

  - The probability of choosing a length J and then an alignment given the English sentence
    - epsilon → probability of choosing length J

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

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

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

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

$$= \underset{A}{\text{argmax}} \frac{\varepsilon}{(I+1)^J} \prod_{j=1}^{J} t(f_j, e_{a_j})$$

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

F → foreign sentence
E → English sentence
A → alignment
J → the length of the source sentence
I → the length of the target sentence

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

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

$t(f_j|e_{aj})$ are the word translation probabilities, which can be estimated using MLE.

IBM Model 1: Parameter Estimation

- How to estimate the parameters:
  - If we have word alignments, we can use MLE and just count
  - If we have the model parameters, we can calculate the alignment probabilities.

$$p(a, f|e) = \prod_{j=1}^{m} t(f_j|e_i)$$

  - Obvious solution: Expectation maximization (EM)
    - We use EM algorithm when we have a variable that we can't optimize directly because it is hidden

The EM Algorithm for Word Alignment

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

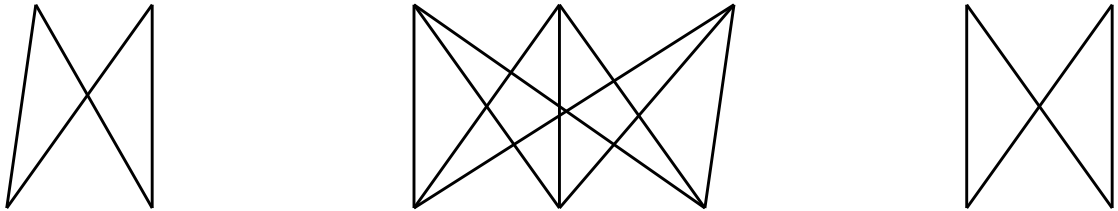
   E Step: Use the current model to compute the probability of all possible alignments of the training data

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

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

EM

... la maison ... la maison blue ... la fleur ...

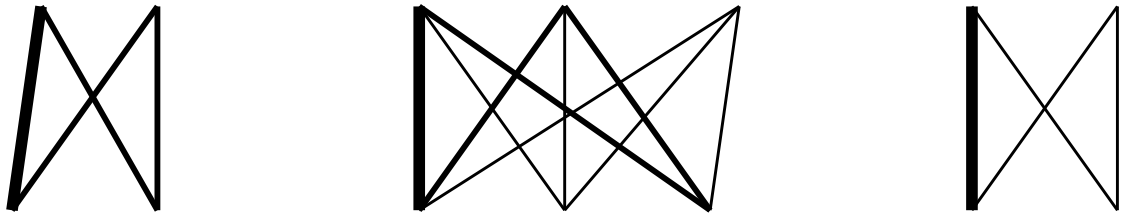


... the house ... the blue house ... the flower ...

- Initial step: all alignments equally likely
- Model learns that, e.g., la is often aligned with the

EM

... la maison ... la maison blue ... la fleur ...

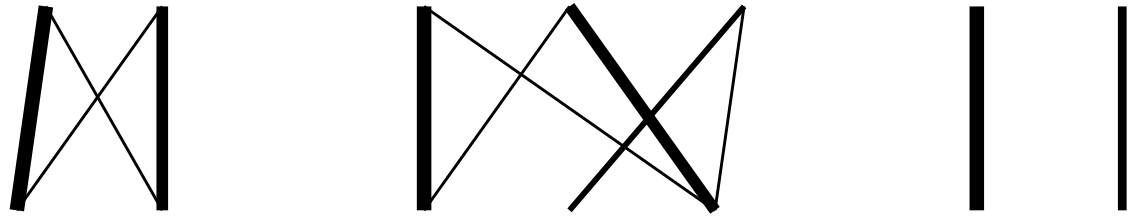


... the house ... the blue house ... the flower ...

- After one iteration
- Alignments, e.g., between la and the are more likely

EM

... la maison ... la maison bleu ... la fleur ...

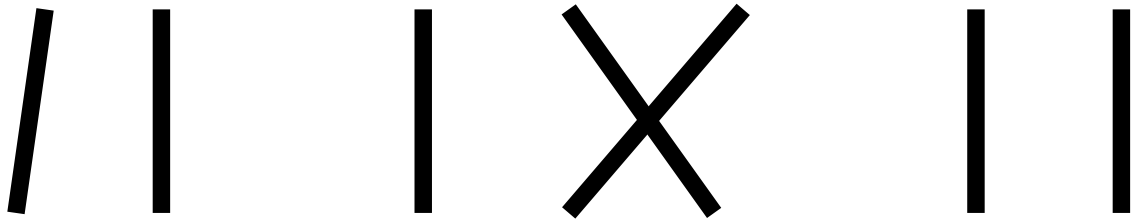


... the house ... the blue house ... the flower ...

- After another iteration
- It becomes apparent that alignments, e.g., between fleur and flower are more likely (pigeon hole principle)

EM

... la maison ... la maison bleu ... la fleur ...

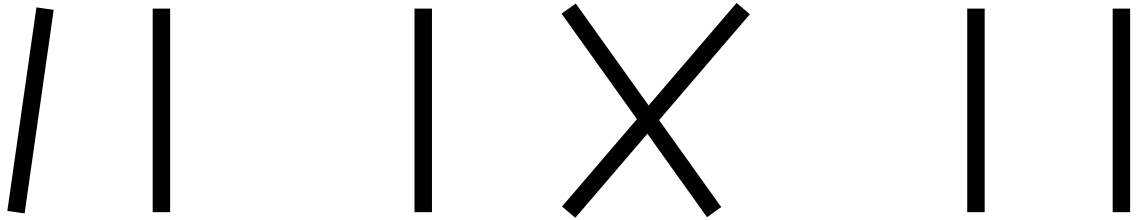


... the house ... the blue house ... the flower ...

- Convergence
- Inherent hidden structure revealed by EM

EM

... la maison ... la maison bleu ... la fleur ...



... the house ... the blue house ... the flower ...

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

- Parameter estimation from the aligned corpus

Example EM Trace for Model 1

- 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:

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

Example EM Trace for Model 1: E step

Training Corpus

green house
casa verde

the house
la casa

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

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

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

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

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

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

Normalize

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

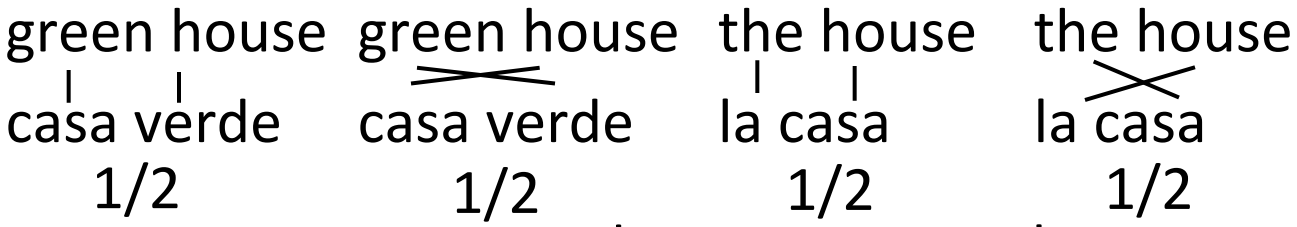
$$\frac{1/9}{2/9} = \frac{1}{2}$$

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

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

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

Example EM Trace for Model 1: M step



Compute weighted Translation counts

$C(f_j, e_{a(j)}) \mathrel{+}= P(a|e,f)$

	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

Example EM Trace for Model 1

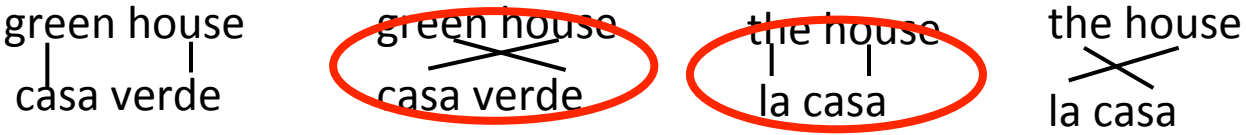
Translation Probabilities

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

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

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

Re-compute Alignment Probabilities P(A, F | E)



Normalize to get P(A | F, E)

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

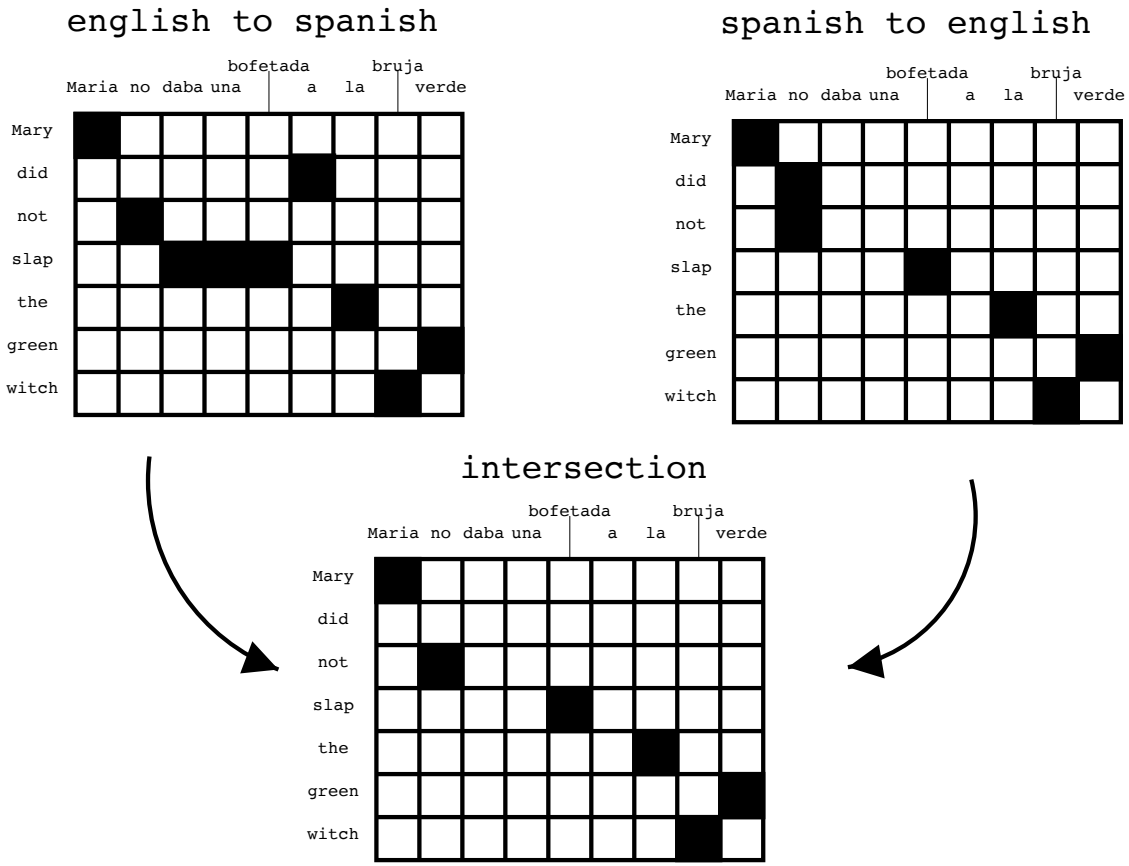
Continue EM iterations until translation parameters converge

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 to get a many-to-many mapping
5. Extract phrases
6. Calculate MLE probabilities for the aligned phrases

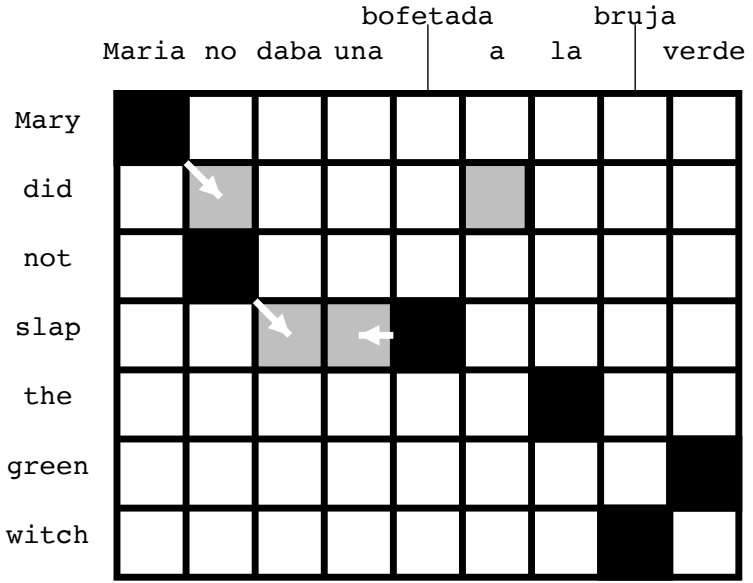
Symmetrization

- The alignments so far are not great since we are assuming each foreign word can only be aligned with a single English word
  - Real alignments are many-to-many
- We can fix this by aligning in both directions, then merge



Growing Heuristics

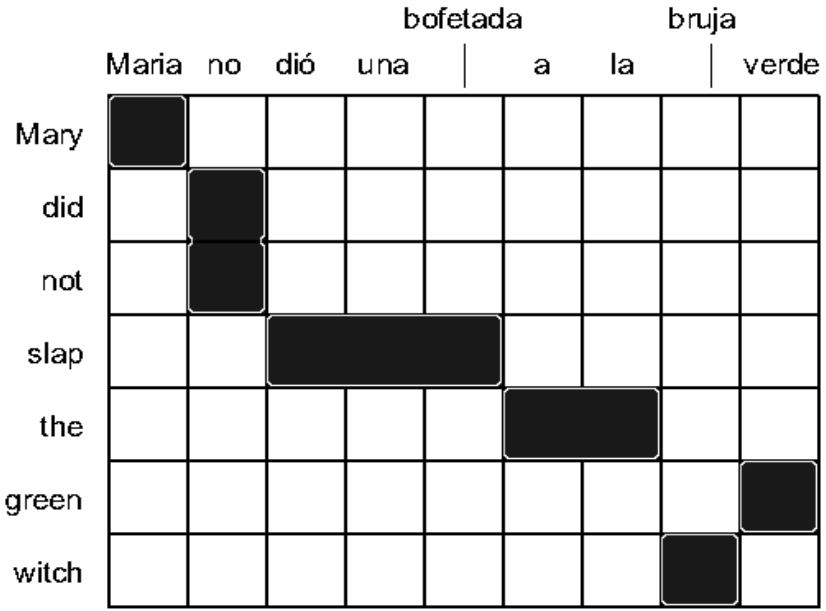
- Add alignment points from union with:
  - directly/diagonally neighboring points
  - finally, add alignments that connect unaligned words in source and/or target



black: intersection **grey**: additional points in union

Extracting phrases from the resulting word alignment

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)
…

The Translation Phrase Table

Philipp Koehn's phrase translations for den Vorschlag

English	ϕ(ē|f)	English	ϕ(ē|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	…	…

Decoding in Phrase-based MT

- Find the best English sentence

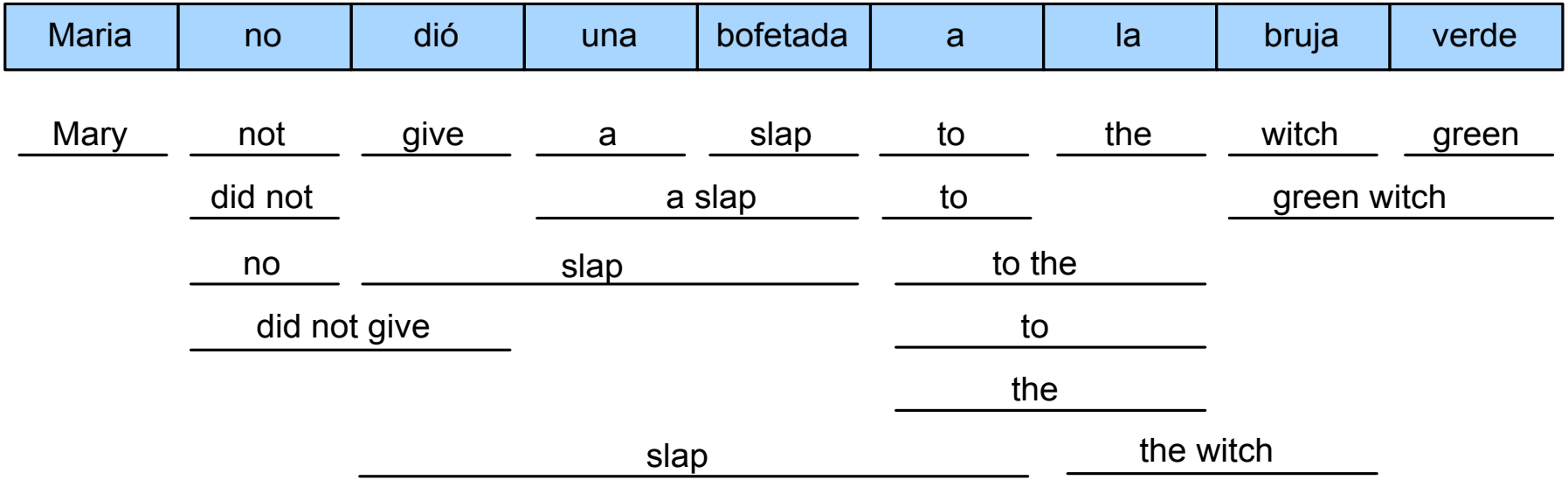
$$\hat{E} = \text{argmax}_E P(E \mid F) = \text{argmax}_E P(F \mid E) P(E)$$

- The Viterbi approximation to the best English sentence:

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

  - Search through the space of all possible English sentences (that include the words/phrases that are possible translations of the source words/ phrases)

Decoding: The lattice of possible English translations for phrases



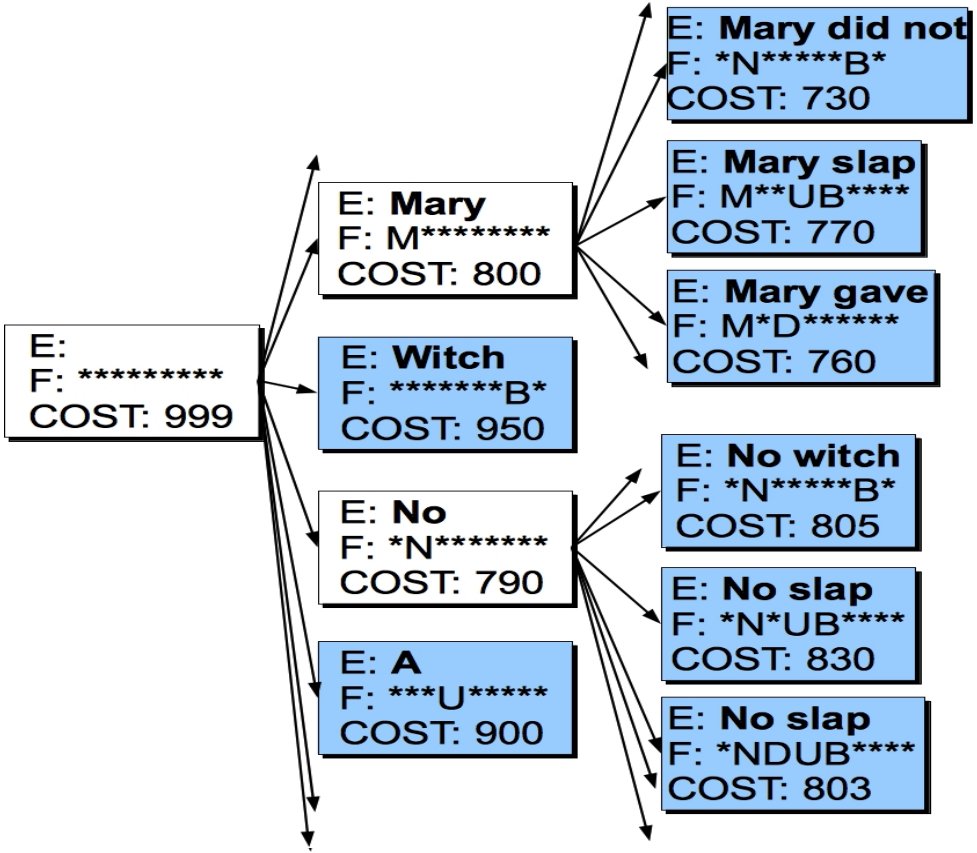
Decoding

- Decoding is a search problem

- In phrase-based MT, decoders are based on best-first search
  - **Stack decoding**; uses priority queues to store all possible states (partial translations) with increasing length.

- For efficiency, **beam search** is used to expand the most likely states.

Stack Decoding



Maria no dio una bofetada…

- After expanding NULL
- After expanding No
- After expanding Mary

Evaluation

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

Automatic Evaluation of MT

- 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)

- “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

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

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 Bigram Precision: 1/5

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 2 Unigram Precision: 7/10

Clip the count of each *n*-gram
to the maximum count of the *n*-gram in any single reference
(avoids having high precision for the candidate: the the the the the)

Computing BLEU: Bigram 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 2 Bigram Precision: 4/9

Brevity Penalty

- BLEU is precision-based: no penalty for dropping words
  - Example → ***of the*** can have 2/2 for its modified unigram precision

- Instead, we use a brevity penalty for translations that are shorter than the reference translations.

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

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

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

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

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.

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

Evaluation Metrics

- Have been useful to quickly evaluate potential system improvements

- Many of them focus on local information – Example:

  it is a guide to action which ensures that the military always obeys the commands of the party

  ensures that the military it is a guide to action which always obeys the commands of the party

Questions?

Midterm

- True or false questions
- Short answer questions
- Exercises: similar to examples in the class
- Bonus

Notes

- Focus on the slides and only read the corresponding sections in the assigned readings
  - All slides including the guest lecturers' are included except *supplement_NLP regression and classification*

- Definitions, goals, evaluation, differences between different techniques or algorithms
  - Ambiguity in NLP
  - Compute precision on the following corpus
  - The difference between different algorithms (e.g. stochastic vs. mini-batch gradient descent)

Notes

- All worked examples in the class are important
  - Whenever you learn something, apply it on an example
- If you don’t understand a topic, you can email me or post your question on Piazza (so other students can also help answering the questions)
- I will post a practice exam the coming few days

Notes

- The exam is closed book and closed notes
- For short questions, you don’t need to got into details. You can just answer briefly
- For exercises, try to write your steps clearly to get at least partial credits

Main Topics (not limited to these topics)

- Regular expressions and finite state automata
  - Substitution and memory
- Words and Morphology
  - Examples: Inflectional vs derivational morphology, lemma, morphemes, stems, tokens… etc
  - Finite State Transducers
- Language modeling
  - Examples: Markov Assumption, Smoothing techniques (Laplace, back-off, interpolation, … etc)

Main Topics (not limited to these topics)

- Text Classification
  - Examples: Non sequential classification, feature representation, naïve bayes algorithm, logistic regression and gradient descent algorithm
- Parsing:
  - Example: CFG, PCFG, CNF, CKY parsing algorithm
- Statistical machine translation
  - Example: classical vs. statistical approaches, Phrase Extraction, and BLEU