Natural Language Processing
Machine Translation

Sudeshna Sarkar

5 Sep 2019

SMT: Basic Idea

- Search for the most probable translation T̂ for a given source sentence S:

$$\hat{T} = argmax_T P(S|T)P(T)$$

Translation Model

- The first translation models relied on translation probabilities for individual words
- Does not account well for idiosyncratic expressions.
- Better: use translation tables for entire phrases instead
- We can decompose the translation probability P(S|T) into I phrase pairs {(s1,t1),...(sI,tI)}:

Translation Model

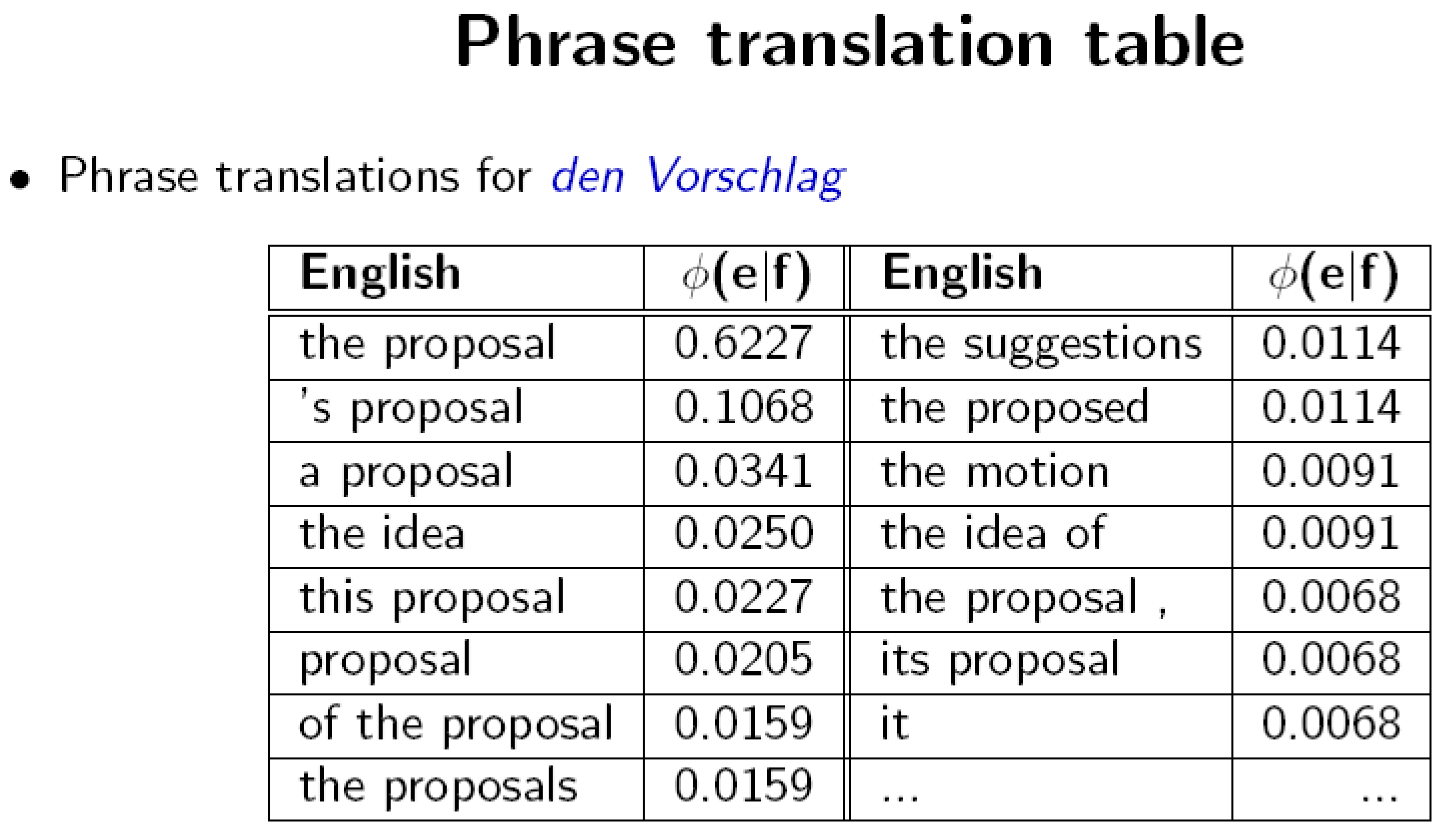
$$\hat{T} = argmax_T \boldsymbol{P(S|T)} P(T)$$

- The first translation models relied on translation probabilities for individual words
- Does not account well for idiosyncratic expressions.
- Better: use translation tables for entire phrases instead
- We can decompose the translation probability P(S|T) into I phrase pairs {(s1,t1),...(sI,tI)}

$$P(S|T) = \prod_{i=1}^{l} \phi(s_i|t_i) d(a_i - b_{i-1})$$

Phrase probability
(as given by the translation table)

Distortion probability
(relative distance between the phrase positions in the two languages)



Phrase translation table

- Phrase translations for *den Vorschlag*

English	ϕ(e|f)	English	ϕ(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	...	...

Slide from Koehn 2008

Language Model

- We also want the translated sentence T to be fluent in the target language
- A statistical language model is a probability distribution over sequences of words w1,w2,...wn
- Typically represented as N-grams and now neural language models

Decoding

IBM translation models

- A generative model based on noisy channel framework
  - Generate the translation sentence ***e*** with regard to the given sentence ***f*** by a stochastic process
    1. Generate the length of ***f***
    2. Generate the ***alignment*** of ***e*** to the target sentence ***f***
    3. Generate the words of ***f***
  -

$$Eng^* = argmax_{Eng} p(Fre|Eng)p(Eng)$$

Word Alignment

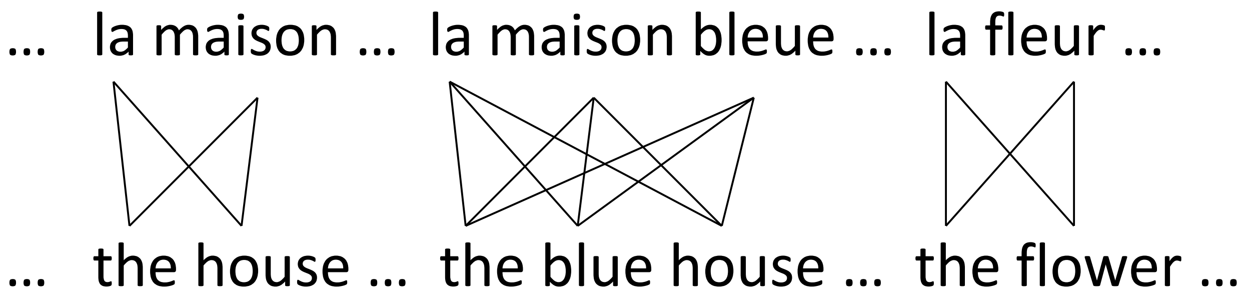
- Directly constructing phrase alignments is difficult, so rely on first constructing word alignments.
- If the alignments are known, the translation probabilities can be calculated simply by counting the aligned words.
- If translation probabilities were known then the alignments could be estimated.

We know neither!

- Use **Expectation-Maximization Algorithm**: an iterative method where the alignments and translation method are refined over time.

EM for training alignment probabilities

Kevin Knight’s example



… la maison … la maison bleue … la fleur …

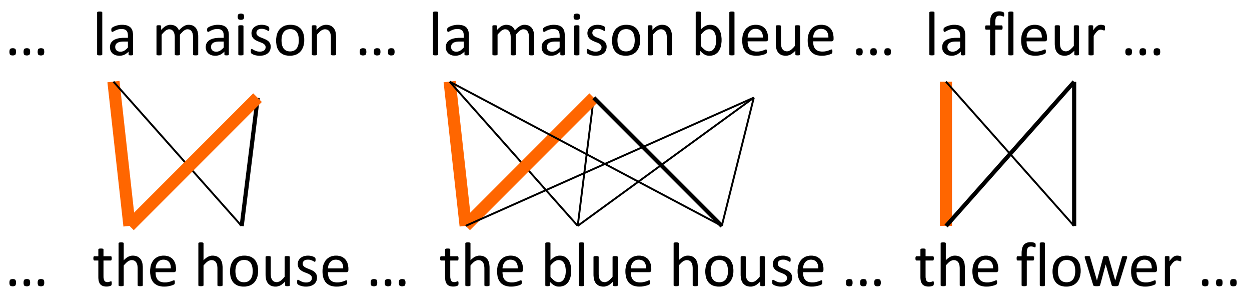
… the house … the blue house … the flower …

Initial stage:

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

EM for training alignment probabilities

Kevin Knight's example



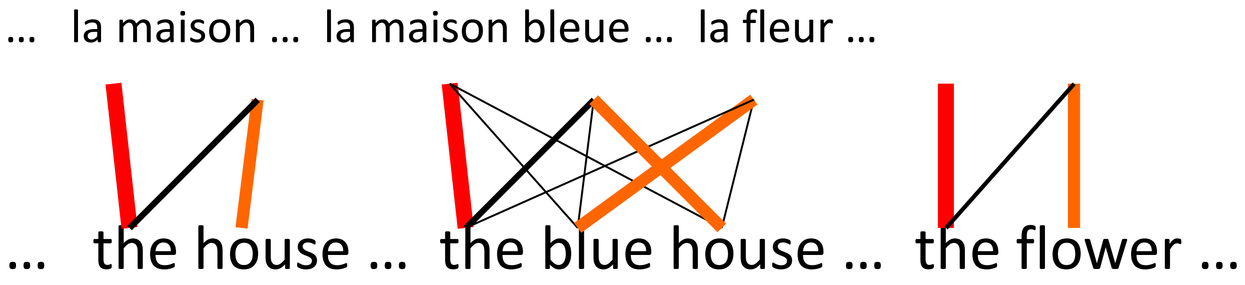
… la maison … la maison bleue … la fleur …

… the house … the blue house … the flower …

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

EM for training alignment probabilities

Kevin Knight’s example



… la maison … la maison bleue … la fleur …

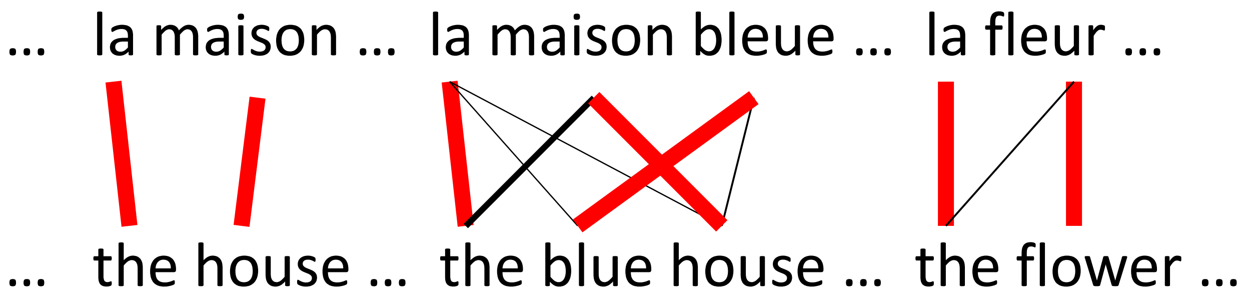
… the house … the blue house … the flower …

“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



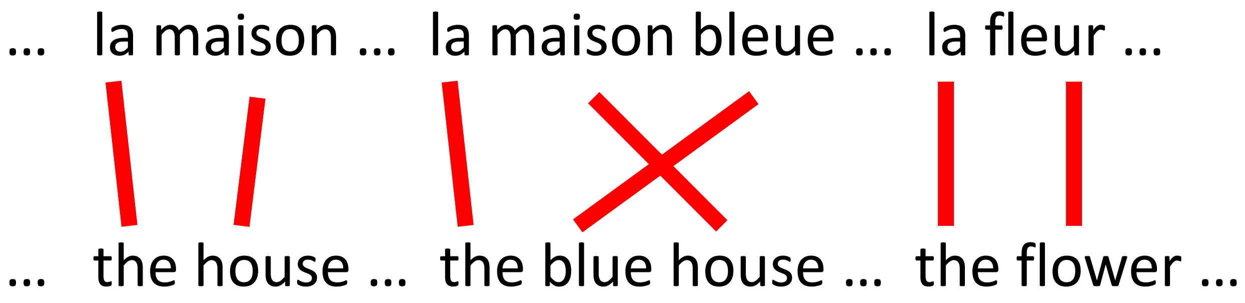
… la maison … la maison bleue … la fleur …

… the house … the blue house … the flower …

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

    Initialize all t(f|e) to any value in [0,1].

1. 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

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)

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

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

Training Corpus

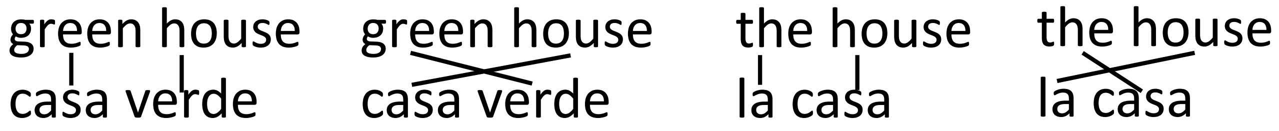
green house
casa verde

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

Compute Alignment Probabilities P(A, F | E)



green house
casa verde

green house
casa verde

the house
la casa

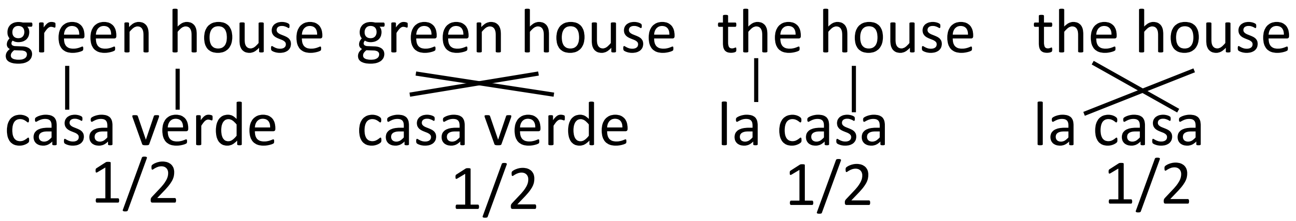
the house
la casa

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
casa verde
1/2

green house
casa verde
1/2

the house
la casa
1/2

the house
la casa
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

EM example continued

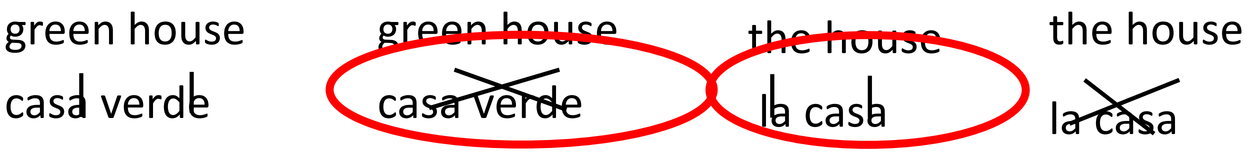
Translation Probabilities

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

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

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

Recompute Alignment Probabilities P(A, F | E)



½ × ¼ =⅛ ½ × ½ =¼ ½ × ½ =¼ ½ × ¼=⅛

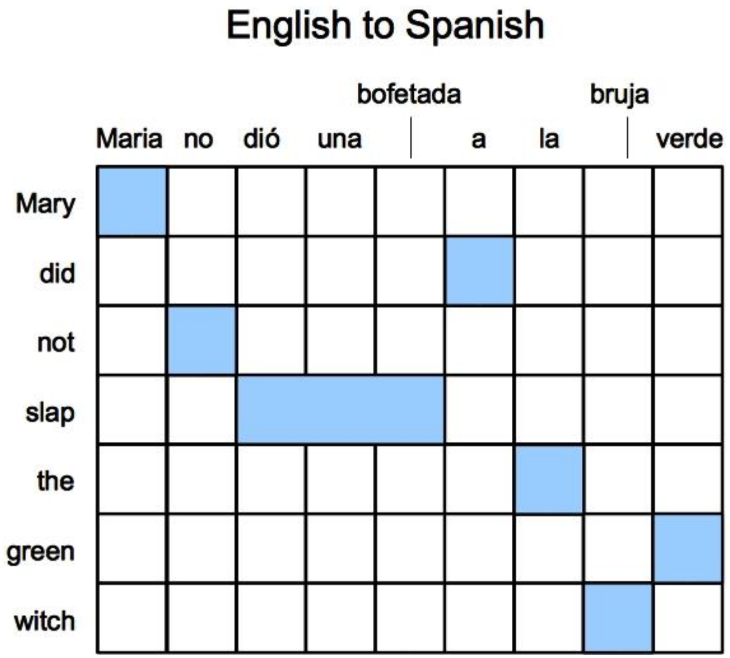
Normalize to get P(A | F, E)

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

Continue EM iterations until translation parameters converge

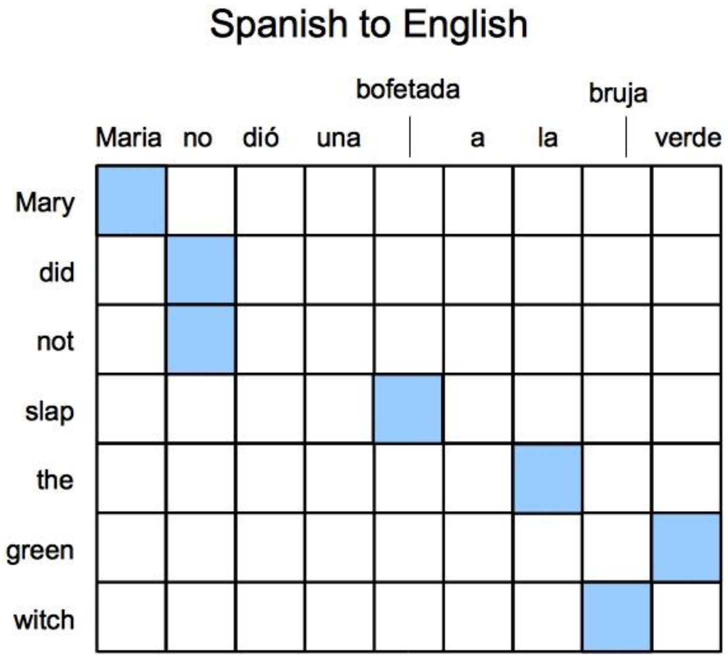
Creating Phrase Alignments from Word Alignments

- 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*→*F* and *F* →*E*
  2) Then we “symmetrize” the two word alignments into one set of phrase alignments



English to Spanish

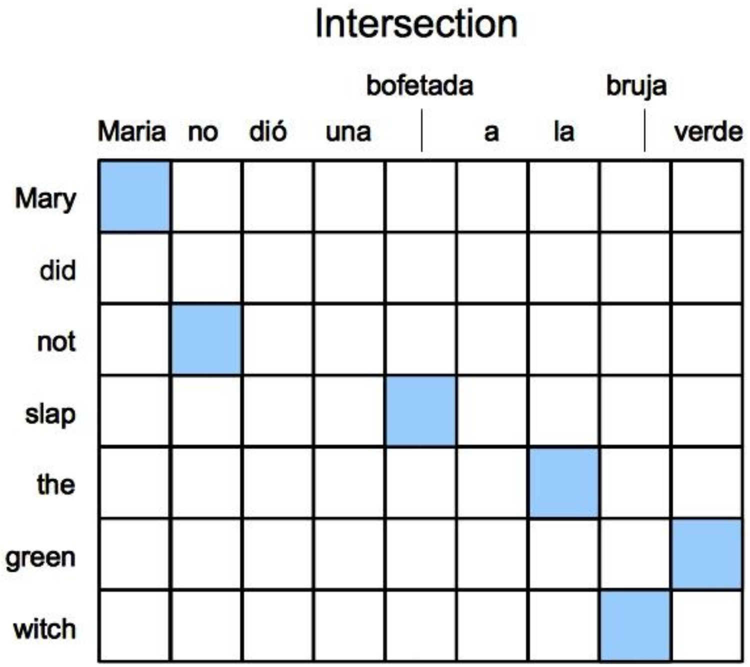
	Maria	no	dió	una	bofetada	a	la	bruja	verde
Mary	■								
did						■			
not		■							
slap			■	■	■				
the							■		
green									■
witch								■	



Spanish to English

	Maria	no	dió	una	bofetada	a	la	bruja	verde
Mary	■								
did		■							
not		■							
slap					■				
the							■		
green									■
witch								■	

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

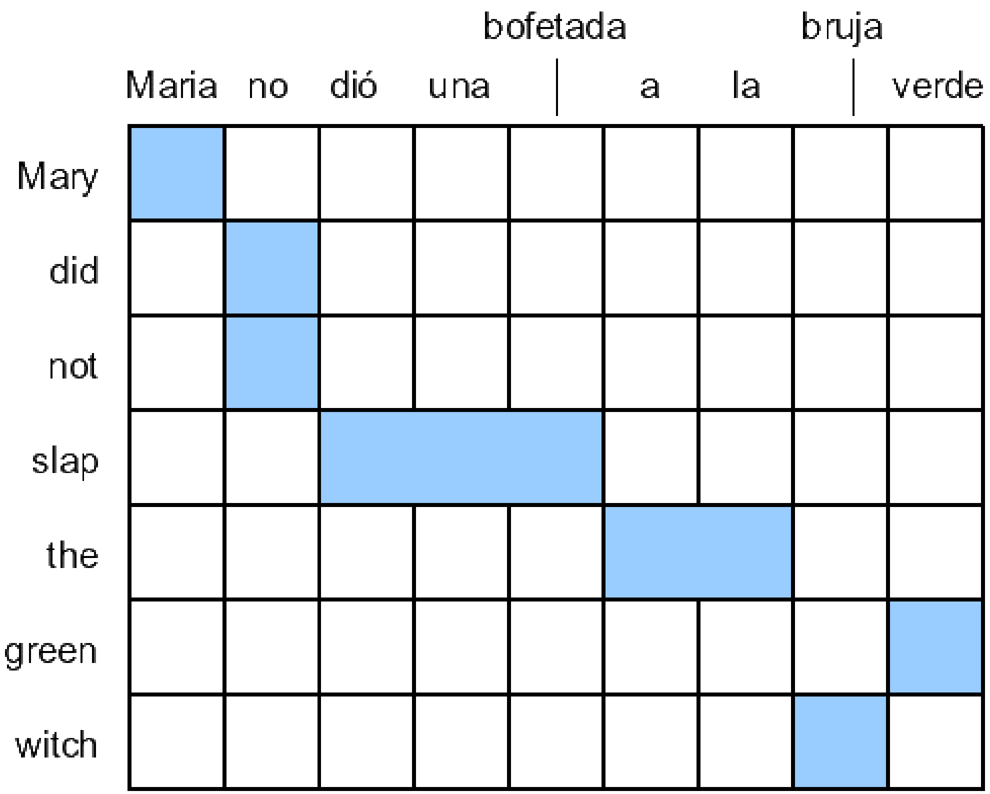


Intersection

	Maria	no	dió	una	bofetada	a	la	bruja	verde
Mary	■								
did									
not		■							
slap					■				
the							■		
green									■
witch								■	

Extracting phrases from the resulting phrase aligment

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



	Maria	no	dió	una	bofetada	a	la	bruja	verde
Mary	■								
did		■							
not		■							
slap			■	■	■				
the						■	■		
green									■
witch								■	

(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: $j\ (\bar{f}, \bar{e})$
- Algorithm
  - Given the phrase aligned bitext
  - And all extracted phrases
  - MLE estimate of ϕ: just count and divide

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

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)

The goal of the decoder

- The best English sentence

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

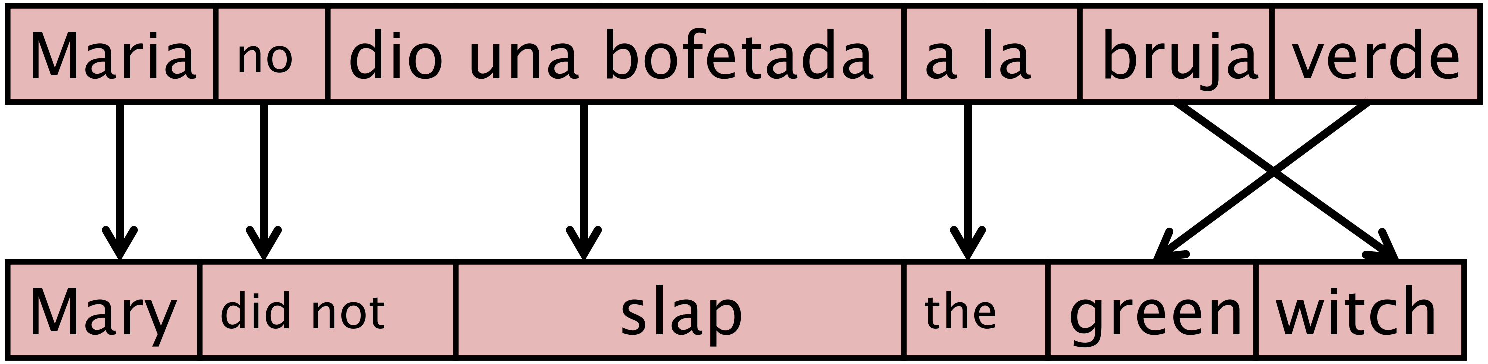
- The Viterbi approximation to the best English sentence:

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

- Search through the space of all English sentences

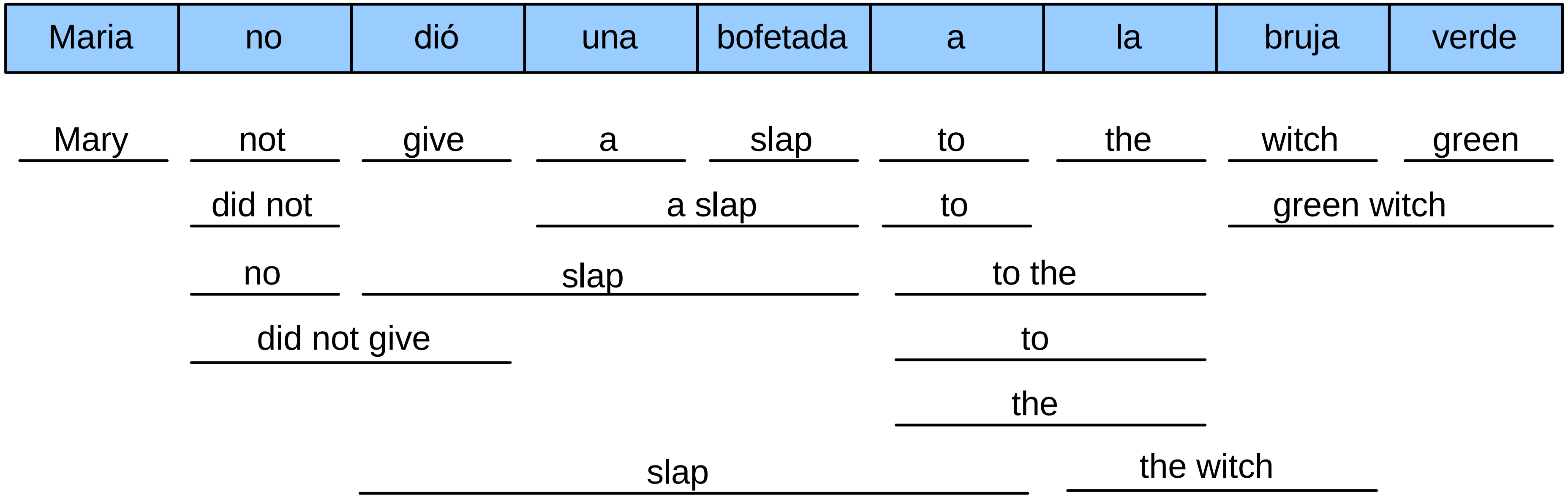
Phrase-based Decoding

Slide adapted from Philipp Koehn



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

Decoding: The lattice of possible English translations for phrases



Maria	no	dió	una	bofetada	a	la	bruja	verde

Mary not give a slap to the witch green

did not a slap to green witch

no slap to the

did not give to

the

slap the witch

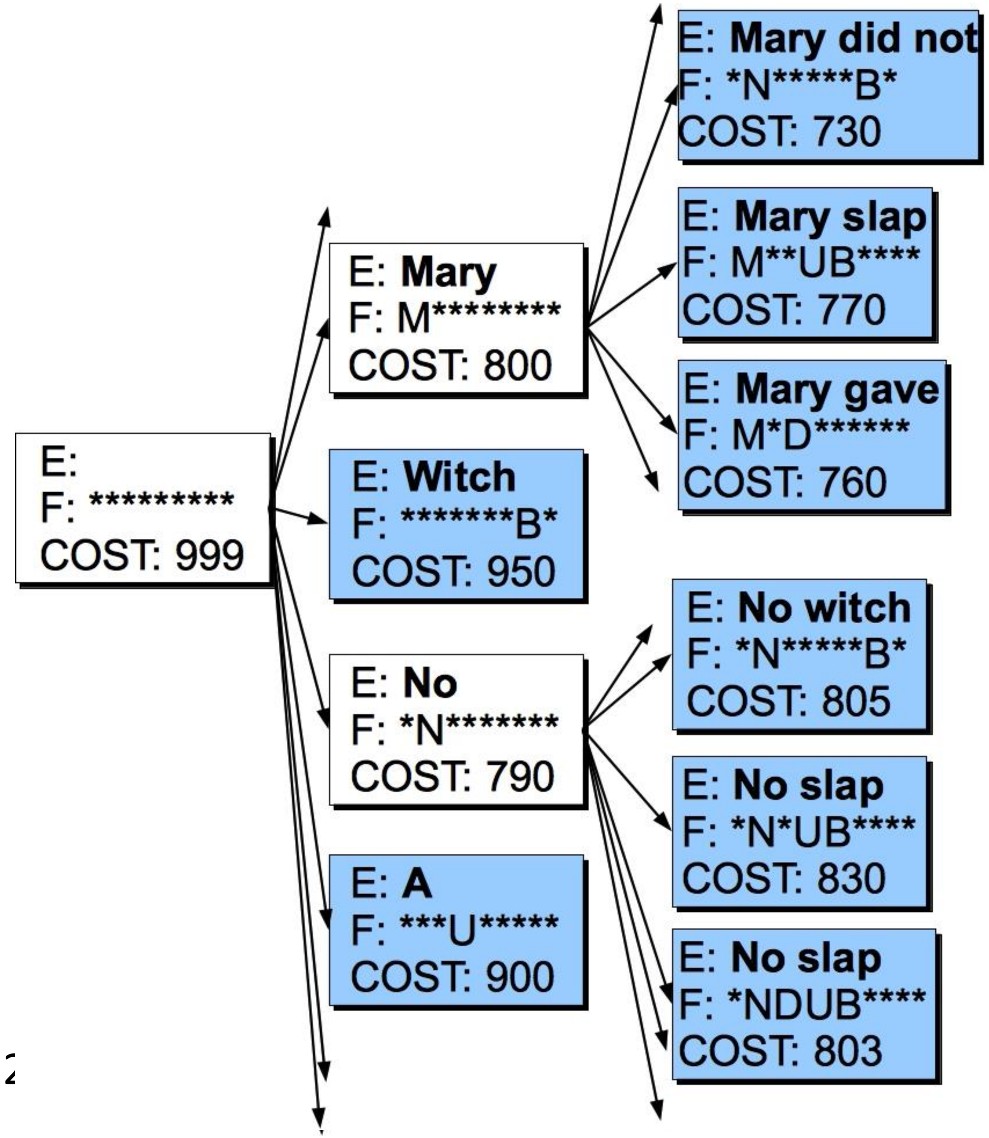
Decoding

- 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

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

  - Estimate of future costs

Decoding by hypothesis expansion



E:
F: *********
COST: 999

E: **Mary**
F: M********
COST: 800

E: **Witch**
F: *******B*
COST: 950

E: **No**
F: *N*******
COST: 790

E: **A**
F: ***U*****
COST: 900

E: **Mary did not**
F: *N*****B*
COST: 730

E: **Mary slap**
F: M**UB****
COST: 770

E: **Mary gave**
F: M*D******
COST: 760

E: **No witch**
F: *N*****B*
COST: 805

E: **No slap**
F: *N*UB****
COST: 830

E: **No slap**
F: *NDUB****
COST: 803

Maria no dio una bofetada…

- After expanding NULL
- After expanding No

E: No slap

- After expanding Mary

F: *N*UB****

COST: 803

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

Evaluating MT

- Human subjective evaluation is the best but is time-consuming and expensive.
- Automated evaluation comparing the output to multiple human reference translations is cheaper and correlates with human judgements.

Human Evaluation of MT

- Ask humans to estimate MT output on several dimensions.
  - **Fluency**: Is the result grammatical, understandable, and readable in the target language.
  - **Fidelity**: Does the result correctly convey the information in the original source language.
    - **Adequacy**: Human judgment on a fixed scale.
      - Bilingual judges given source and target language.
      - Monolingual judges given reference translation and MT result.
    - **Informativeness**: Monolingual judges must answer questions about the source sentence given only the MT translation (task-based evaluation).

Computer-Aided Translation Evaluation

- **Edit cost**: Measure the number of changes that a human translator must make to correct the MT output.
  - Number of words changed
  - Amount of time taken to edit
  - Number of keystrokes needed to edit

Automatic Evaluation of MT

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

BLEU

- Determine number of *n*-grams of various sizes that the MT output shares with the reference translations.
- Compute a modified precision measure of the *n*-grams in MT result.

BLEU Example

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.

Cand 1 Unigram Precision: 5/6

BLEU Example

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.

Cand 1 Bigram Precision: 1/5

BLEU Example

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

Cand 2 Unigram Precision: 7/10

BLEU Example

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.

Cand 2 Bigram Precision: 4/9

Modified *N*-Gram Precision

- Average *n*-gram precision over all *n*-grams up to size *N* (typically 4) using geometric mean.

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

$$p = \sqrt[N]{\prod_{n=1}^{N} p_n}$$

Cand 1: $p = \sqrt[2]{\frac{5}{6}\frac{1}{5}} = 0.408$

Cand 2: $p = \sqrt[2]{\frac{7}{10}\frac{4}{9}} = 0.558$

Brevity Penalty

- Not easy to compute recall to complement precision since there are multiple alternative gold-standard references and don't need to match all of them.
- Instead, use a penalty for translations that are shorter than the reference translations.
- Define effective reference length, *r*, for each sentence as the length of the reference sentence with the largest number of *n*-gram matches. Let *c* be the candidate sentence length.

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

BLEU Score

- Final BLEU Score: BLEU = $BP \times p$

Cand 1: Mary no slap the witch green.

Best Ref: Mary did not slap the green witch.

$$c = 6, \quad r = 7, \qquad BP = e^{(1-7/6)} = 0.846$$

$$BLEU = 0.846 \times 0.408 = 0.345$$

Cand 2: Mary did not give a smack to a green witch.

Best Ref: Mary did not smack the green witch.

$$c = 10, \quad r = 7, \qquad BP = 1$$

$$BLEU = 1 \times 0.558 = 0.558$$

BLEU Score Issues

- BLEU has been shown to correlate with human evaluation when comparing outputs from different SMT systems.
- However, it is does not correlate with human judgments when comparing SMT systems with manually developed MT (Systran) or MT with human translations.
- Other MT evaluation metrics have been proposed that claim to overcome some of the limitations of BLEU.

Syntax-Based Statistical Machine Translation

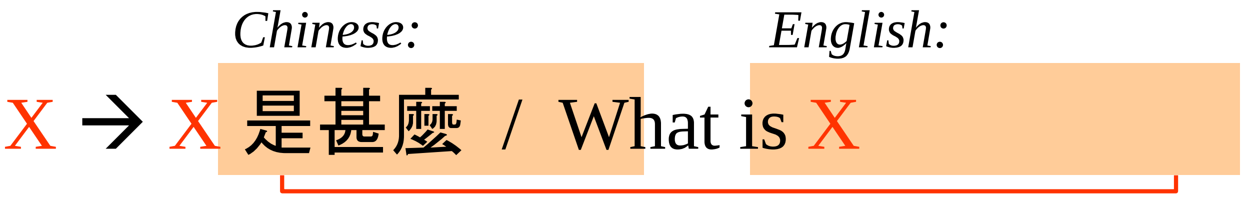
- Recent SMT methods have adopted a syntactic transfer approach.
- Improved results demonstrated for translating between more distant language pairs, e.g. Chinese/English.

Synchronous Grammar

- Multiple parse trees in a single derivation.
- Used by (Chiang, 2005; Galley et al., 2006).
- Describes the hierarchical structures of a sentence and its translation, and also the correspondence between their sub-parts.

Synchronous Productions

- Has two RHSs, one for each language.



Chinese: *English:*

X → X 是甚麼 / What is X

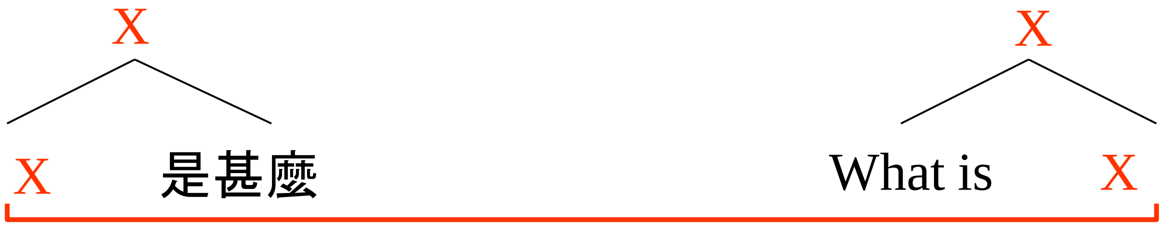
Syntax-Based MT Example

Input: 俄亥俄州的首府是甚麼？

Syntax-Based MT Example

Input: 俄亥俄州的首府是甚麼？

Syntax-Based MT Example



X

X 是甚麼

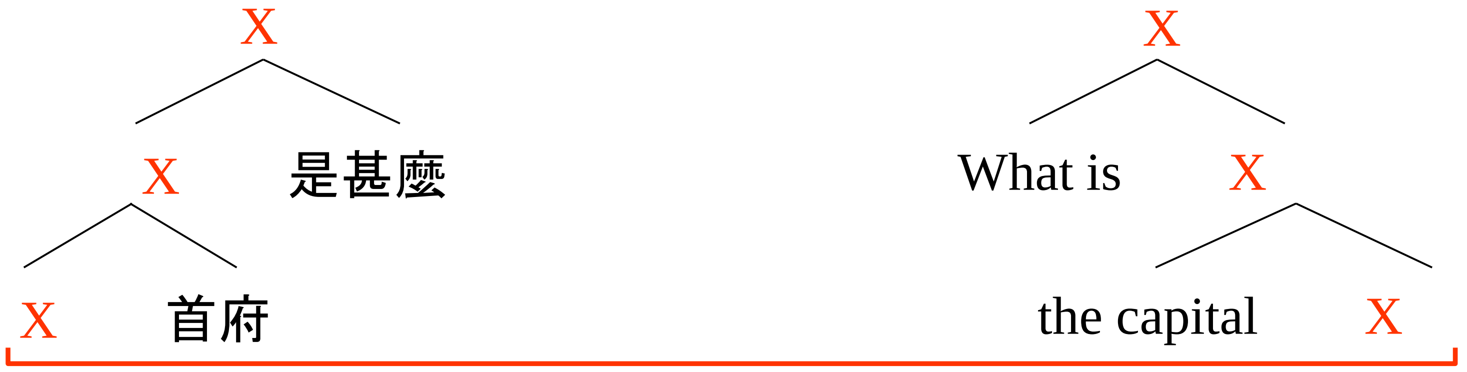
X

What is X

Input: 俄亥俄州的首府是甚麼？

X → X 是甚麼 / What is X

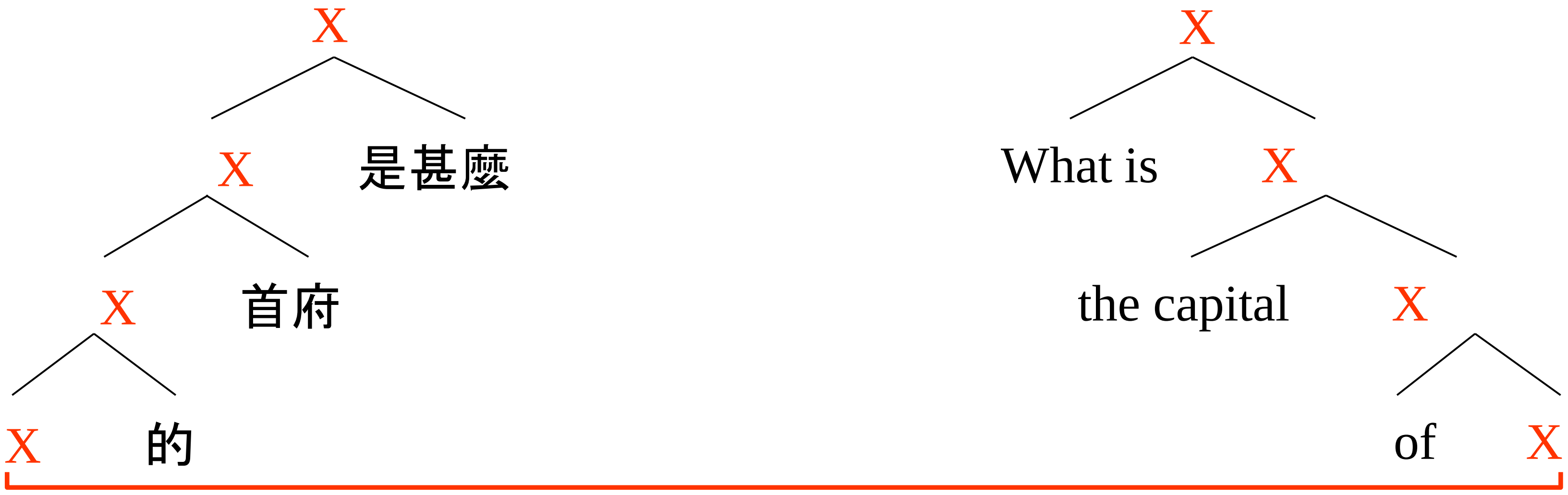
Syntax-Based MT Example



Input: 俄亥俄州的首府是甚麼？

X → X 首府 / the capital X

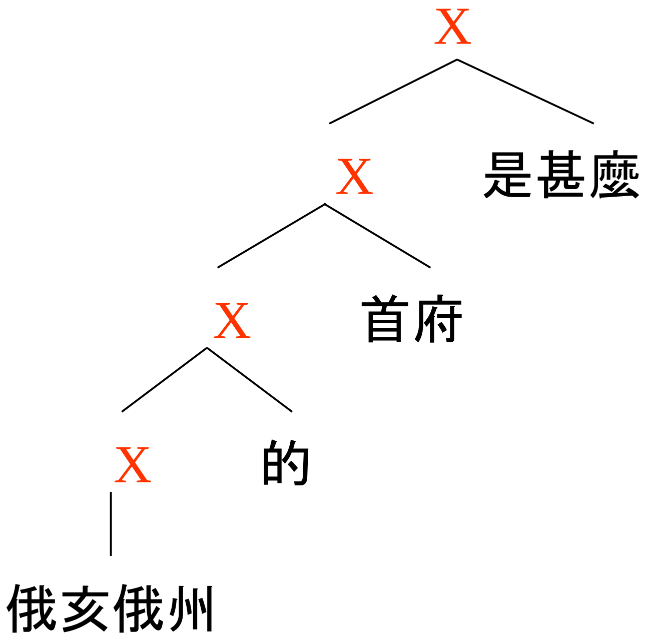
Syntax-Based MT Example



Input: 俄亥俄州的首府是甚麼？

X → X 的 / of X

Syntax-Based MT Example



X

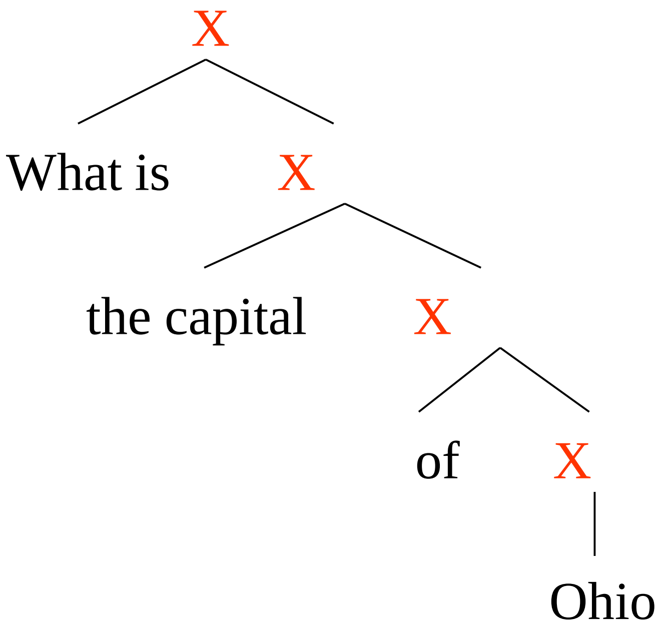
X 是甚麼

X 首府

X 的

俄亥俄州

Input: 俄亥俄州的首府是甚麼？



X

What is X

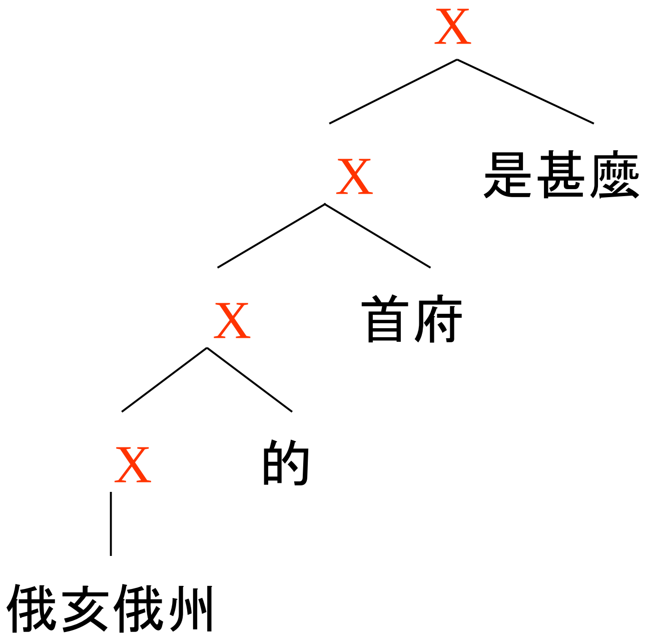
the capital X

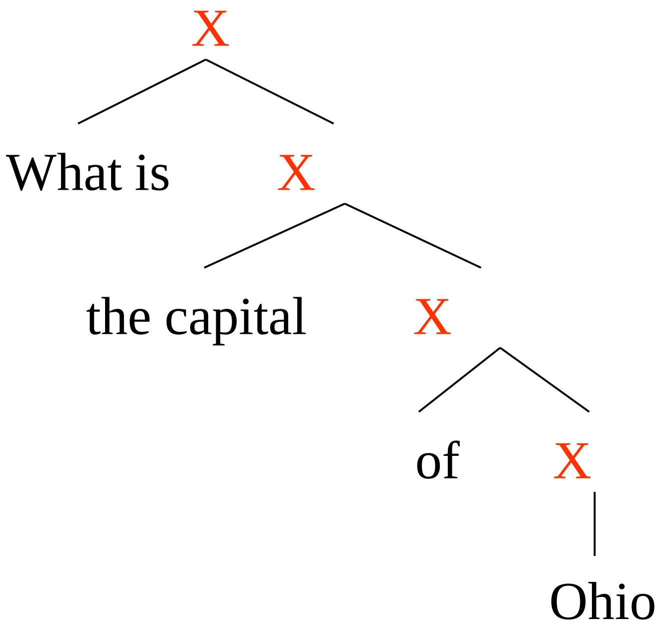
of X

Ohio

X → 俄亥俄州 / Ohio

Syntax-Based MT Example





Input: 俄亥俄州的首府是甚麼？

Output: What is the capital of Ohio?

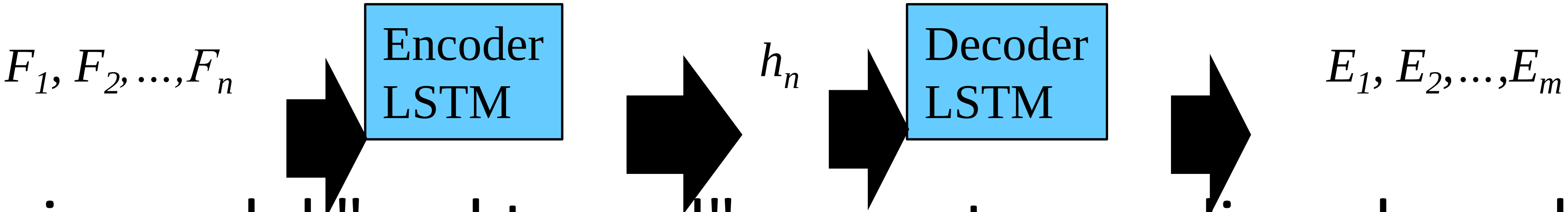
Synchronous Derivations and Translation Model

- Need to make a probabilistic version of synchronous grammars to create a translation model for P(*F* | *E*).
- Each synchronous production rule is given a weight λ_i that is used in a maximum-entropy (log linear) model.
- Parameters are learned to maximize the conditional log-likelihood of the training data.

$$\lambda^{\star} = \arg\max_{\lambda} \sum_{j} \log \Pr_{\lambda}(\mathbf{f}_j | \mathbf{e}_j)$$

Neural Machine Translation (NMT)

- Encoder/Decoder framework maps sentence in source language to a "deep vector" then another LSTM maps this vector to a sentence in the target language

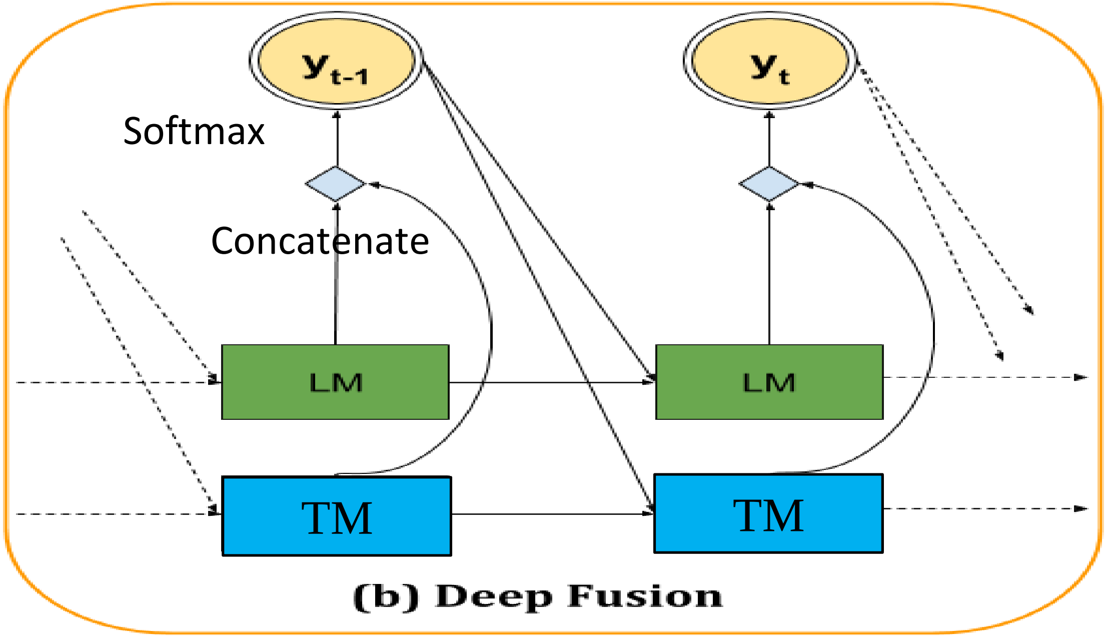


- Train model "end to end" on sentence-aligned parallel corpus.

NMT with Language Model

- Vanilla LSTM approach does not use a language model so does not exploit monolingual data for the target language.
- Can integrate an LSTM language model using “deep fusion.”

- Decoder predicts the next word from a concatenation of the hidden states of both the translation and language LSTM models.



Conclusions

- MT methods can usefully exploit various amounts of syntactic and semantic processing along the Vauquois triangle.
- Statistical MT methods can automatically learn a translation system from a parallel corpus.
- Typically use a noisy-channel model to exploit both a bilingual translation model and a monolingual language model.
- Neural LSTM methods are currently the state-of-the-art.