

Natural Language Processing Machine Translation

Sudeshna Sarkar

30 Aug 2019

History of Machine Translation

(Based on work by John Hutchins, mt-archive.info)

- Before the computer: In the mid 1930s, a French-Armenian Georges Artsrouni and a Russian Petr Troyanskii applied for patents for 'translating machines'.
- The pioneers (1947-1954): the first public MT demo was given in 1954 (by IBM and Georgetown University).
- Machine translation was one of the first applications envisioned for computers

History of MT (2)



One naturally wonders if the problem of translation could conceivably be treated as a problem in cryptography. When I look at an article in Russian, I say: "This is really written in English, but it has been coded in some strange symbols. I will now proceed to decode."

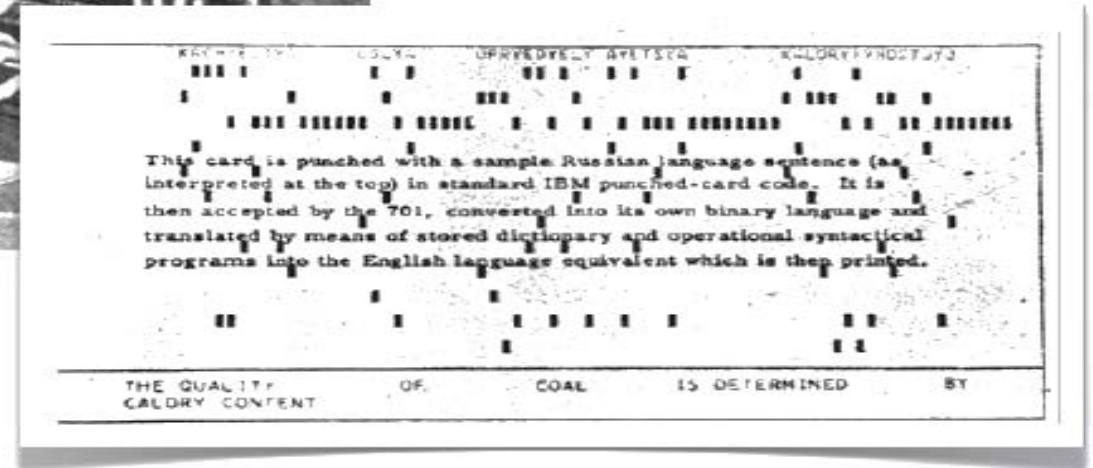
Warren Weaver, 1947

Warren Weaver, PhD was an American scientist, mathematician, and science administrator. He is widely recognized as one of the pioneers of machine translation, and as an important figure in creating support for science in the United States.

History of MT (3)



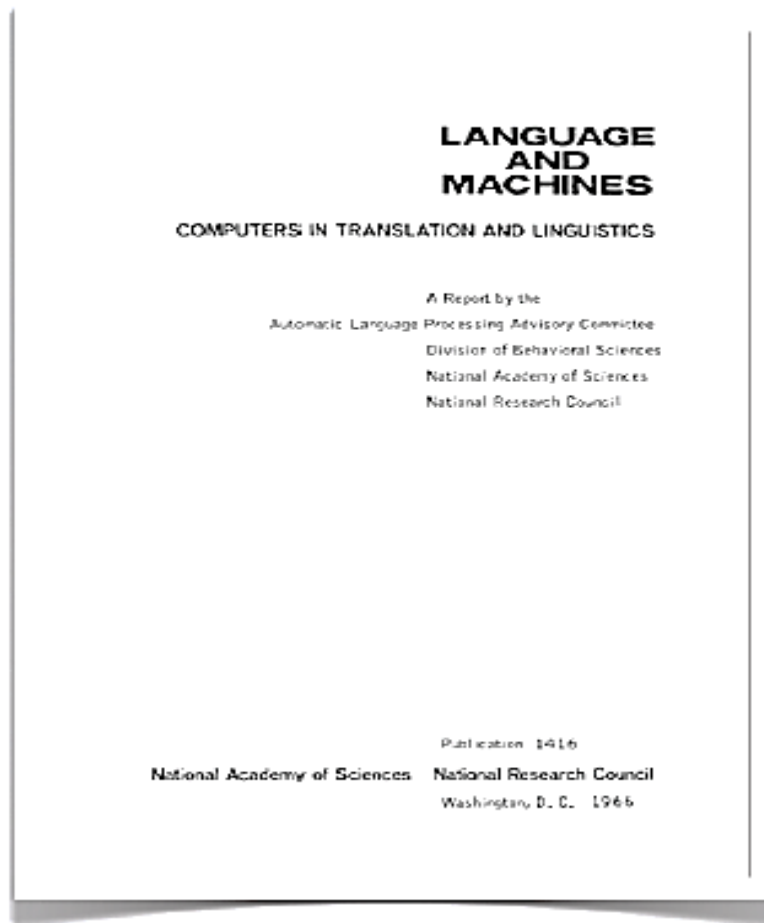
First demonstrated by IBM in 1954 with a basic word-for-word translation system



History of MT (4)

- The decade of optimism (1954-1966) ended with the...
- ALPAC (Automatic Language Processing Advisory Committee) report in 1966: "There is no immediate or predictable prospect of useful machine translation."

History of MT (5)



The ALPAC Report

The ALPAC (Automatic Language Processing Advisory Committee) was a govt. committee of seven scientists.

Their 1966 report was very skeptical of the progress in computational linguistics and machine translation.

History of MT (6)

- The aftermath of the ALPAC report...
- Research on machine translation virtually stopped from 1966 to 1980

History of MT (7)

- Then, a rebirth...
- The 1980s: Interlingua, example-based Machine Translation
- The 1990s: Statistical MT
- The 2000s: Hybrid MT
- The 2010s: Google, real-time, mobile, Crowdsourcing, more hybrid approaches

What is Machine Translation?

Automatic conversion of text/speech from one natural language to another

e.g.

- Be the change you want to see in the world
- वह परिवर्तन बनो जो संसार में देखना चाहते हो
- Google (Hindi): वह बदलाव बनें जो आप दुनिया में देखना चाहते हैं
- Google (Bengali): পরিবর্তন আপনি বিশ্বের দেখতে চান
- Google (Tamil): நீங்கள் உலகத்தில் பார்க்க விரும்பும் மாற்றமாக இருங்கள்
- Google (Telugu): మీరు ప్రపంచంలో చూడాలనుకుంటున్న మార్పుగా ఉండండి

Why study machine translation?

- One of the most challenging problems in Natural Language Processing
- Pushes the boundaries of NLP
- Involves analysis as well as synthesis
- Involves all layers of NLP: morphology, syntax, semantics, pragmatics, discourse
- Theory and techniques in MT are applicable to a wide range of other problems like transliteration, speech recognition and synthesis

Why is machine translation difficult?

Language Divergence: the great diversity among languages of the world

- Word order: SOV (Hindi), SVO (English), VSO, OSV,
- Free (Sanskrit) vs rigid (English) word order
- Analytic (Chinese) vs Polysynthetic (Finnish) languages
- Different ways of expressing same concept
- Case marking systems
- Language registers
- Inflectional systems [infixing (Arabic), fusional (Sanskrit), agglutinative (Marathi)]
- ... and much more

Why is machine translation difficult?

- Ambiguity
 - Same word, multiple meanings:
 - Same meaning, multiple words: जल, पानी, नीर (water)
- Word Order
 - Underlying deeper syntactic structure
 - Phrase structure grammar?
 - Computationally intensive
- Morphological Richness
 - Identifying basic units of words

Machine Translation

- Automatically translate one natural language into another.

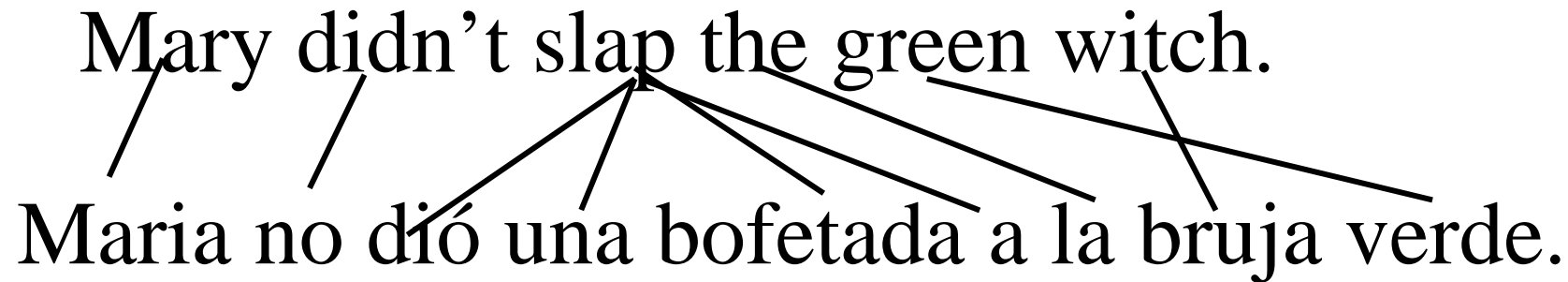
Mary didn't slap the green witch.



Maria no dió una bofetada a la bruja verde.

Word Alignment

- Shows mapping between words in one language and the other.



How do human translate languages?

- Is a bilingual dictionary sufficient?

John loves Mary.
| |
Jean aime Marie.

John told Mary a story.
| / \ / \
Jean a raconté une histoire à Marie.

John is ~~a~~ computer scientist.
| | / \
Jean est informaticien.

John swam across the lake.
| / \ / \
Jean a traversé le lac à la nage.

Correspondences

- One-to-one **A bilingual dictionary is clearly insufficient!**
 - John = Jean, aime = loves, Mary=Marie
- One-to-many/many-to-one
 - Mary = [à Marie]
 - [a computer scientist] = informaticien
- Many-to-many
 - [swam across __] = [a traversé __ à la nage]
- Reordering required
 - told Mary¹ [a story]² = a raconté [une histoire]² [à Marie]¹

Ambiguity Resolution is Required for Translation

- Syntactic and semantic ambiguities must be properly resolved for correct translation:
 - “John plays the guitar.” → “John toca la guitarra.”
 - “John plays soccer.” → “John juega el fútbol.”
- An apocryphal story is that an early MT system gave the following results when translating from English to Russian and then back to English:
 - “The spirit is willing but the flesh is weak.” ⇒
“The liquor is good but the meat is spoiled.”
 - “Out of sight, out of mind.” ⇒ “Invisible idiot.”

Lexical divergences

- Different senses of homonymous words generally have different translations

English	- German
(river) bank	- Ufer

- Different senses of polysemous words may also have different translations

I **know** that he bought the book: Je **sais qu'**il a acheté le livre.

I **know** Peter: Je **connais** Peter.

I **know** math: Je **m'y connais en** maths.

Lexical Gaps

- Some words in one language do not have a corresponding term in the other.
 - **Rivière** (river that flows into ocean) and **fleuve** (river that does not flow into ocean) in French
 - **Schedenfraude** (feeling good about another's pain) in German.
 - **Oyakoko** (filial piety) in Japanese

Syntactic divergences

- Word order
 - SVO (Sbj-Verb-Obj), SOV, VSO,...
 - fixed or free?
- Head-marking vs. dependent-marking
 - Dependent-marking (English): the man's house
 - Head-marking (Hungarian): the man house-his
- Pro-drop languages can omit pronouns
 - Italian (with inflection): I eat = mangio; he eats = mangia
 - Chinese (without inflection): I/he eat: chīfàn

Semantic divergences

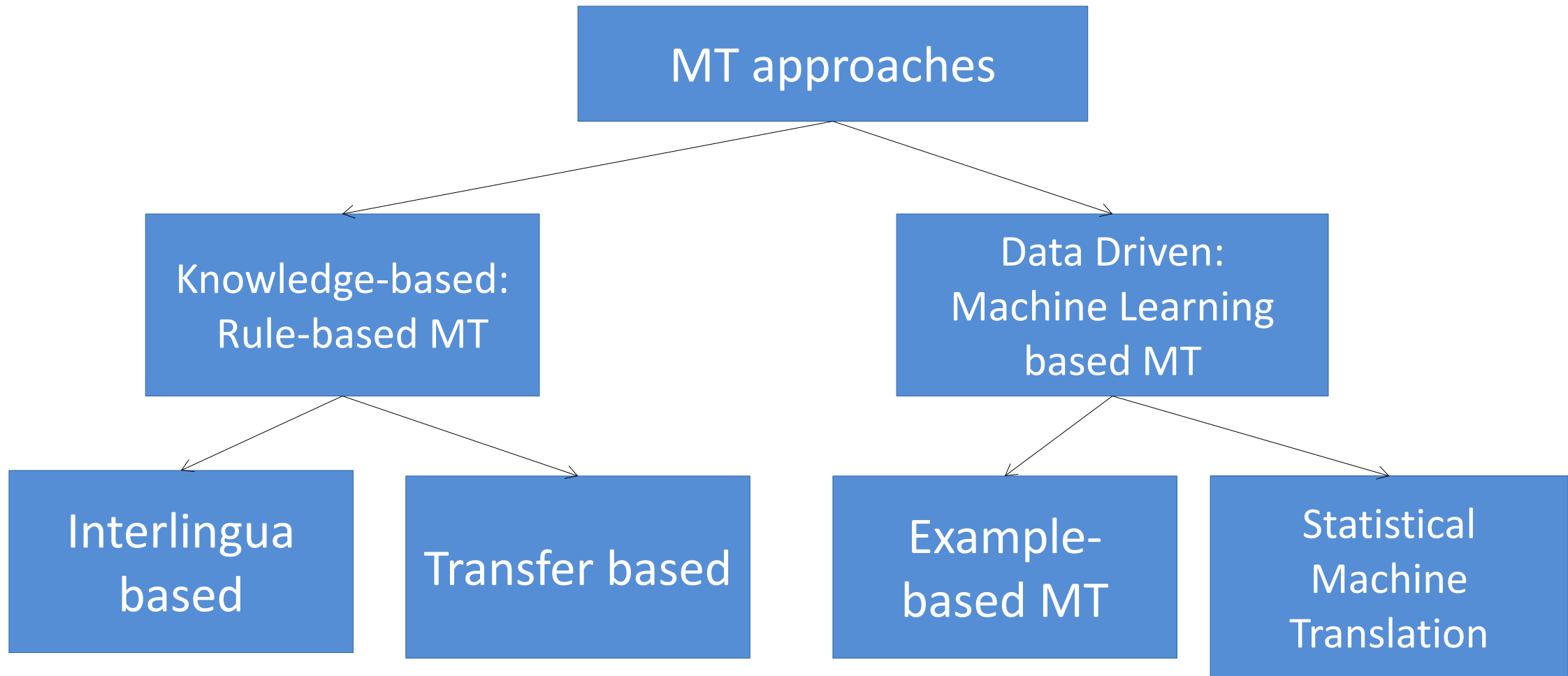
- Aspect
 - English has a progressive aspect
 - ‘Peter swims’ vs. ‘Peter is swimming’
 - German can only express this with an adverb:
 - ‘Peter schwimmt’ vs. ‘Peter schwimmt gerade’

Clearly, a bilingual dictionary is insufficient; and machine translation is difficult!

Linguistic Issues Making MT Difficult

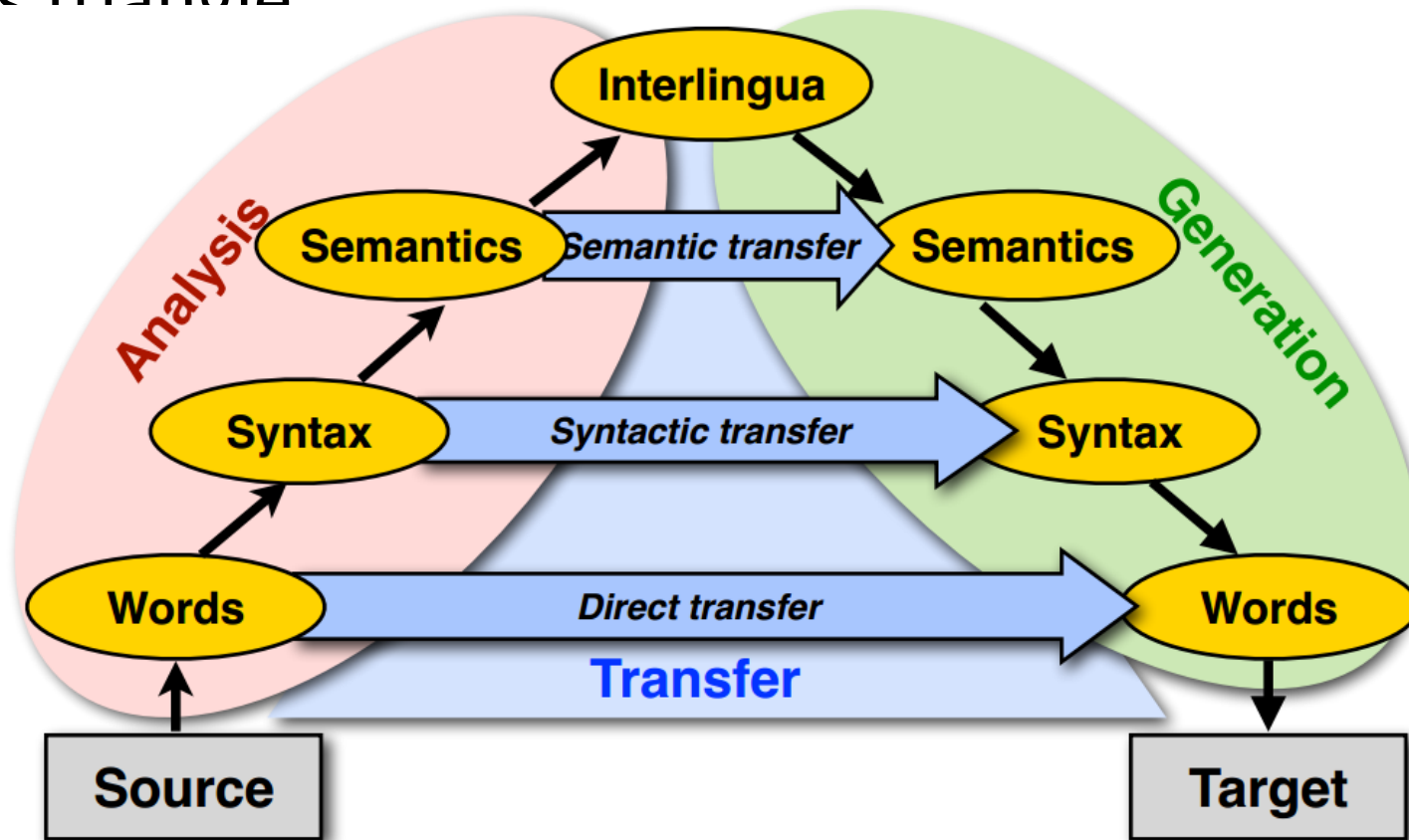
- Morphological issues with ***agglutinative***, ***fusion*** and ***polysynthetic*** languages with complex word structure.
- Syntactic variation between ***SVO*** (e.g. English), ***SOV*** (e.g. Hindi), and ***VSO*** (e.g. Arabic) languages.
 - SVO languages use prepositions
 - SOV languages use postpositions
- ***Pro-drop*** languages regularly omit subjects that must be inferred.

Taxonomy of MT systems

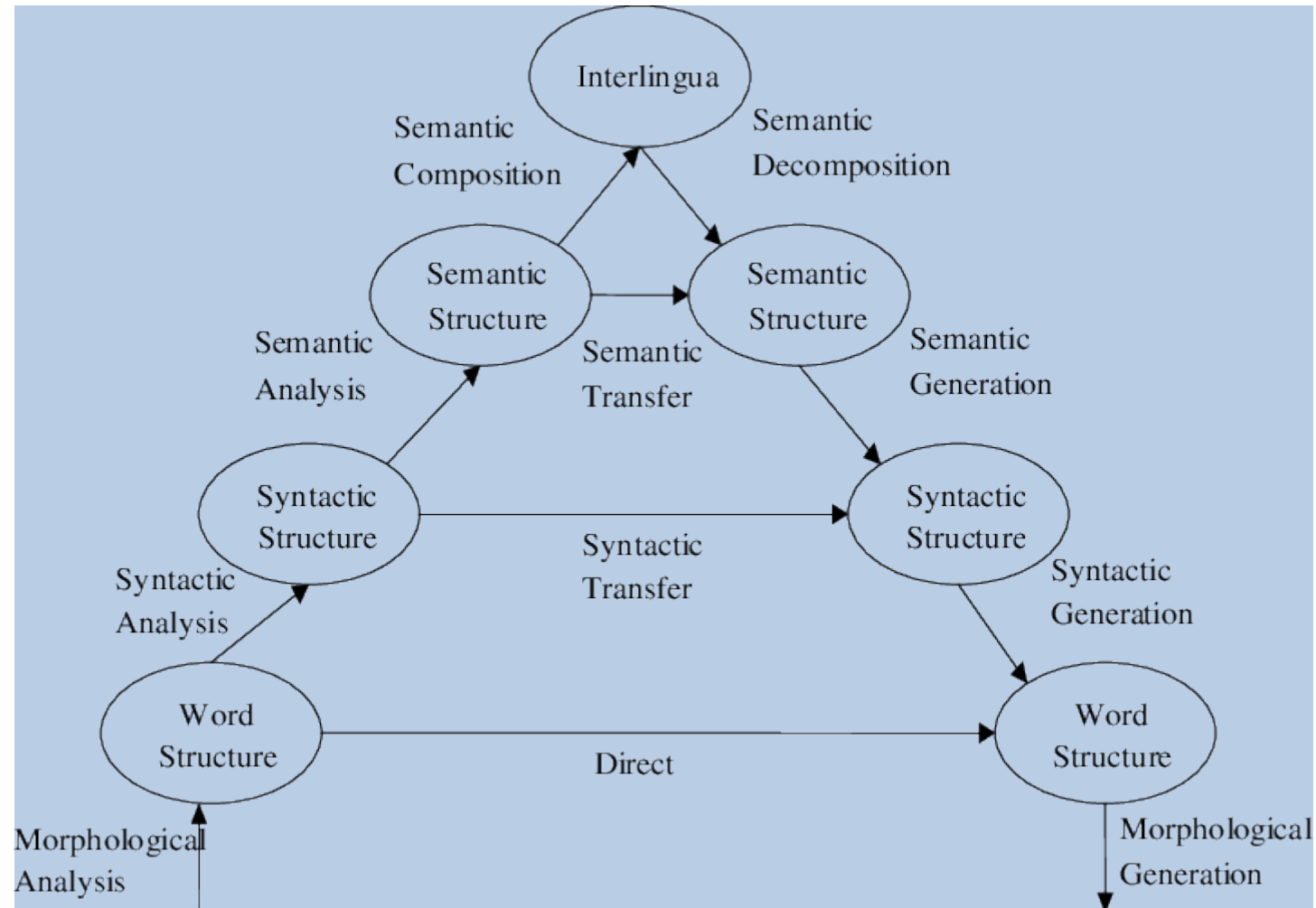


Machine translation approaches

- The Vauquois triangle



Vauquois Triangle



Direct Transfer

- Morphological Analysis
 - Mary didn't slap the green witch. →
Mary DO:PAST not slap the green witch.
- Lexical Transfer
 - Mary DO:PAST not slap the green witch.
 - Maria no dar:PAST una bofetada a la verde bruja.
- Lexical Reordering
 - Maria no dar:PAST una bofetada a la bruja verde.
- Morphological generation
 - Maria no dió una bofetada a la bruja verde.

Syntactic Transfer

- Simple lexical reordering does not adequately handle more dramatic reordering such as that required to translate from an SVO to an SOV language.
- Need syntactic transfer rules that map parse tree for one language into one for another.
 - English to Spanish:
 - $NP \rightarrow Adj\ Nom \Rightarrow NP \rightarrow Nom\ ADJ$
 - English to Japanese:
 - $VP \rightarrow V\ NP \Rightarrow VP \rightarrow NP\ V$
 - $PP \rightarrow P\ NP \Rightarrow PP \rightarrow NP\ P$

Semantic Transfer

- Some transfer requires semantic information.
- Semantic roles can determine how to properly express information in another language.
- In Chinese, PPs that express a goal, destination, or benefactor occur *before* the verb but those expressing a recipient occur *after* the verb.
- Transfer Rule
 - English to Chinese
 - $VP \rightarrow V \text{ PP[+benefactor]} \Rightarrow VP \rightarrow \text{PP[+benefactor]} V$

Statistical MT

- Manually encoding comprehensive bilingual lexicons and transfer rules is difficult.
- SMT acquires knowledge needed for translation from a *parallel corpus* or *bitext* that contains the same set of documents in two languages.
- First align the sentences in the corpus based on simple methods that use coarse cues like sentence length to give bilingual sentence pairs.

Picking a Good Translation

- A good translation should be ***faithful*** and correctly convey the information and tone of the original source sentence.
- A good translation should also be ***fluent***, grammatically well structured and readable in the target language.
- Final objective:

$$T_{best} = \underset{T \in \text{Target}}{\operatorname{argmax}} \text{faithfulness}(T, S) \text{fluency}(T)$$

Noisy Channel Model

- Based on analogy to information-theoretic model used to decode messages transmitted via a communication channel that adds errors.
- Assume that source sentence was generated by a “noisy” transformation of some target language sentence and then use Bayesian analysis to recover the most likely target sentence that generated it.

Translate foreign language sentence

$$F = f_1, f_2, \dots, f_m$$

to an English sentence

$$\hat{E} = e_1, e_2, \dots, e_I$$

that maximizes $P(E | F)$

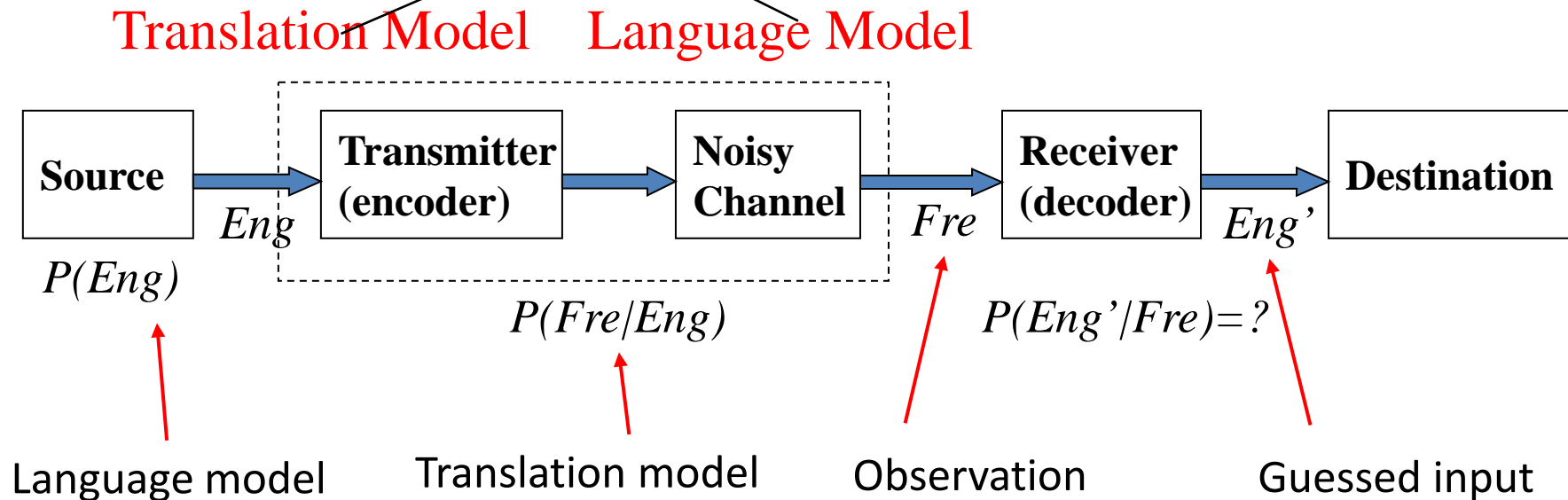


- Task is to recover e from noisy f .
- $P(F|E)$: Translation model
- $P(E)$: Language model

Bayesian Analysis of Noisy Channel

$$\begin{aligned}\hat{E} &= \operatorname{argmax}_{E \in \text{English}} P(E | F) \\ &= \operatorname{argmax}_{E \in \text{English}} \frac{P(F | E)P(E)}{P(F)} \\ &= \operatorname{argmax}_{E \in \text{English}} \underbrace{P(F | E)}_{\text{Translation Model}} \underbrace{P(E)}_{\text{Language Model}}\end{aligned}$$

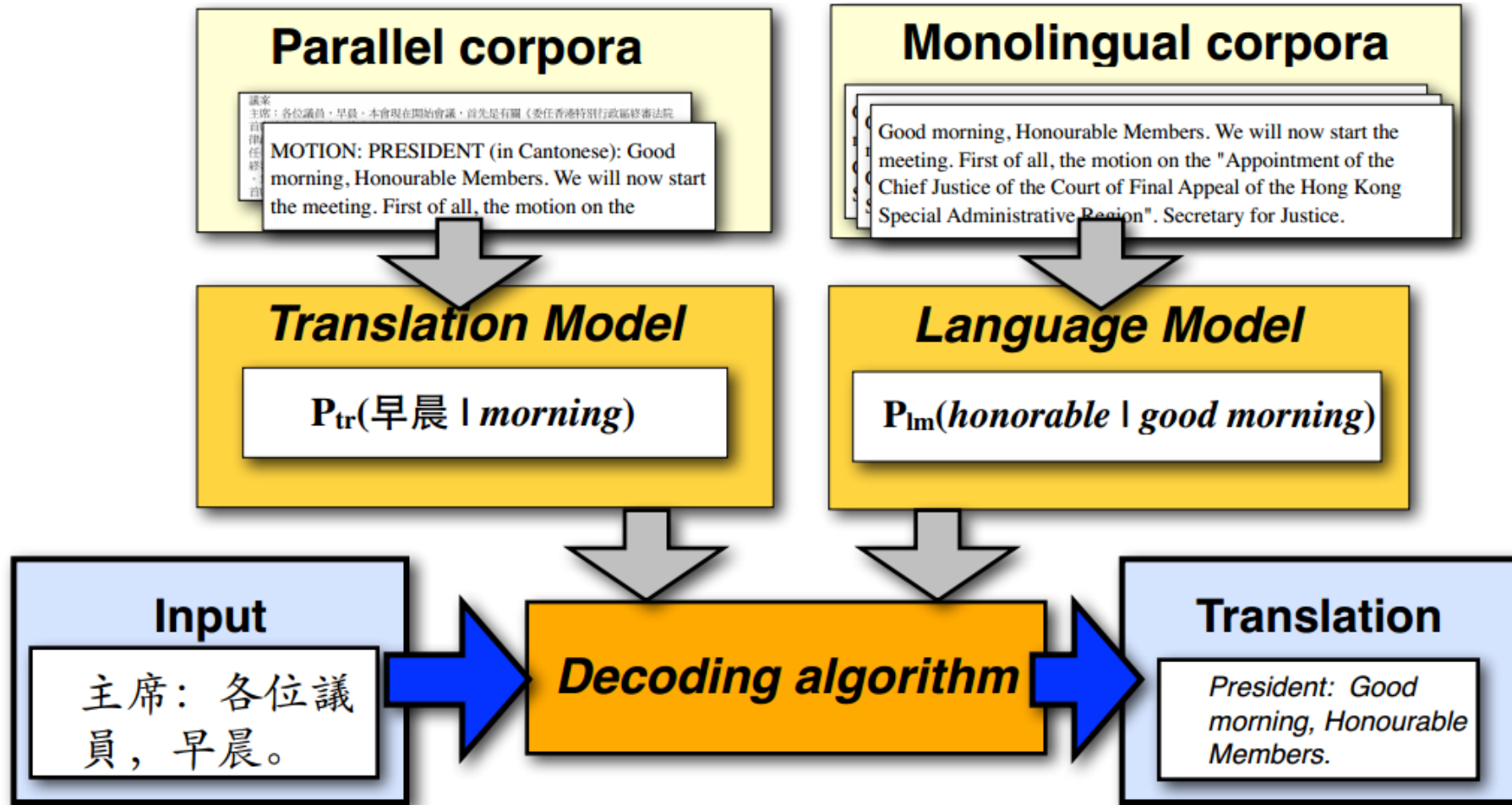
A **decoder** determines the most probable translation \hat{E} given F



Language Model

- Use a standard n -gram language model for $P(E)$ or ..
- Can be trained on a large, unsupervised mono-lingual corpus for the target language E .

Statistical machine translation



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
 -

$$Eng^* = \operatorname{argmax}_{Eng} p(Fre|Eng)p(Eng)$$

Word Alignment

- Directly constructing phrase alignments is difficult, so rely on first constructing word alignments.
- Can learn to align from supervised word alignments, but human-aligned bitexts are rare and expensive to construct.
- Typically use an unsupervised EM-based approach to compute a word alignment from unannotated parallel corpus.

Word alignment

- One to one, one to many and reordering

John told Mary a story.
Jean a raconté une histoire à Marie.

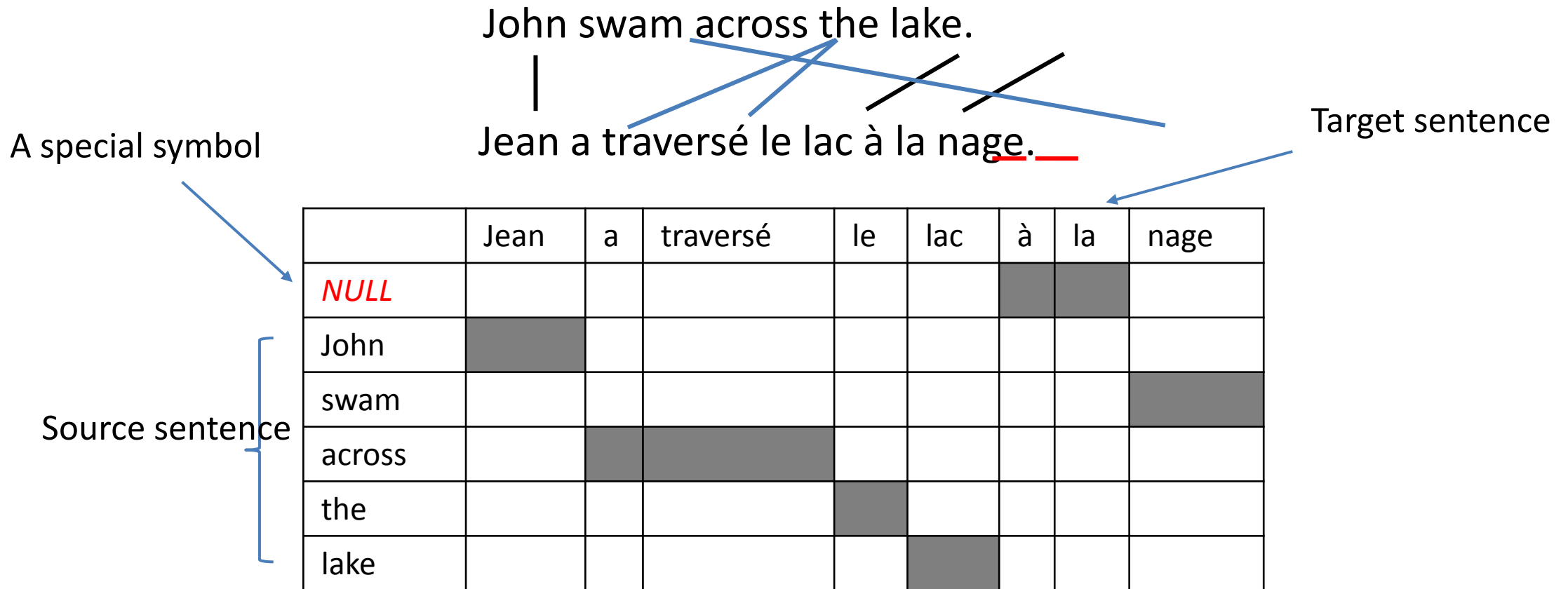
Target sentence

Source sentence

	Jean	a	raconté	une	histoire	à	Marie
John							
told							
Mary							
a							
story							

Word alignment

- Many to one and missing word



Representing word alignments

- Alignment table

		1	2	3	4	5	6	7	8
		Jean	a	traversé	le	lac	à	la	nage
0	NULL								
1	John								
2	swam								
3	across								
4	the								
5	lake								



Target Position	1	2	3	4	5	6	7	8
Source Position	1	3	3	4	5	0	0	2

One to Many Alignment

- To simplify the problem, typically assume each word in F aligns to 1 word in E (but assume each word in E may generate more than one word in F).
- Some words in F may be generated by the NULL element of E .
- Therefore, alignment can be specified by a vector A giving, for each word in F , the index of the word in E which generated it.

IBM Model 1

- First model proposed in seminal paper by Brown *et al.* in 1993 as part of CANDIDE, the first complete SMT system.
- Assumes following simple generative model of producing F from $E=e_1, e_2, \dots e_I$
 - Choose length, J , of F sentence: $F=f_1, f_2, \dots f_J$
 - Choose a 1 to many alignment $A=a_1, a_2, \dots a_J$
 - For each position in F , generate a word f_j from the aligned word in E :
 e_{a_j}

IBM translation models

- Translation model with word alignment

- $p(Fre|Eng) = \sum_{a \in A(Eng, Fre)} p(Fre, a|Eng)$

marginalize over all possible alignments a

- Generate the words of \mathbf{f} with respect to alignment \mathbf{a}

$$p(\mathbf{f}, \mathbf{a} | \mathbf{e}) = \boxed{p(m | \mathbf{e})} \prod_{j=1}^m \boxed{p(a_j | a_{1..j-1}, f_{1..j-1}, m, \mathbf{e})} \boxed{p(f_j | a_{1..j}, f_{1..j-1}, m, \mathbf{e})}$$


Length of target sentence \mathbf{f} Word alignment a_j Translation of f_j

IBM translation models

- Sequence of 5 translation models
 - Different assumptions and realization of the components in the translation models, i.e., length model, alignment model and translation model
 - Model 1 is the simplest and becomes the basis of follow-up IBM translation models

Parameters in Model 1

- Length probability $p(m|\mathbf{e})$
 - Probability of generating a source sentence of length m given a target sentence \mathbf{e}
 - Assumed to be a constant - $p(m|\mathbf{e}) = \epsilon$
- Alignment probability $p(a|\mathbf{e})$
 - Probability of source position i is aligned to target position j
 - Assumed to be uniform - $p(a|\mathbf{e}) = \frac{1}{n}$

 *length of source sentence*

Parameters in Model 1

- Translation probability $p(f|a, e)$
 - Probability of English word e_i is translated to French word f_j -
 $p(f_j|e_{a_j})$
- After the simplification, Model 1 becomes

$$\begin{aligned} p(\mathbf{f}, \mathbf{a}|\mathbf{e}) &= p(m|\mathbf{e}) \prod_{j=1}^m p(a_j|a_{1..j-1}, f_{1..j-1}, m, \mathbf{e}) p(f_j|a_{1..j}, f_{1..j-1}, m, \mathbf{e}) \\ &= \frac{\epsilon}{(n+1)^m} \prod_{j=1}^m p(f_j|e_{a_j}) \end{aligned}$$

 We add a NULL word in the source sentence

Recap: IBM translation models

- Translation model with word alignment

- $p(Fre|Eng) = \sum_{a \in A(Eng, Fre)} p(Fre, a|Eng)$

marginalize over all possible alignments a

- Generate the words of f with respect to alignment a

$$p(\mathbf{f}, \mathbf{a} | \mathbf{e}) = \boxed{p(m | \mathbf{e})} \prod_{j=1}^m \boxed{p(a_j | a_{1..j-1}, f_{1..j-1}, m, \mathbf{e})} \boxed{p(f_j | a_{1..j}, f_{1..j-1}, m, \mathbf{e})}$$

Length of target sentence f Word alignment a_j Translation of f_j

Generative process in Model 1

For a particular English sentence $e = e_1..e_n$ of length n

0	1	2	3	4	5
NULL	John	swam	across	the	lake

1. Choose a length m for the target sentence (e.g $m = 8$)

1	2	3	4	5	6	7	8
?	?	?	?	?	?	?	?

2. Choose an alignment $a = a_1 ... a_m$ for the source sentence

Target Position	1	2	3	4	5	6	7	8
Source Position	1	3	3	4	5	0	0	2

3. Translate each source word e_{a_j} into the target language

English	John	across	across	the	lake	NULL	NULL	swam
Alignment	1	3	3	4	5	0	0	2
Encoded	Jean	a	traversé	le	lac	à	la	nage

Source

Order of action



Transmitter

Decoding process in Model 1

$$p(\mathbf{e}|\mathbf{f}) = 1e^{-55}$$

For a particular English sentence $e = e_1..e_n$ of length n

$$p(\mathbf{e})$$

0	1	2	3	4	5
NULL	John	flies	across	the	river

Search through all English sentences

1. Choose a length m for the target sentence (e.g $m = 8$)

$$p(m|\mathbf{e}) = \epsilon$$

1	2	3	4	5	6	7	8
?	?	?	?	?	?	?	?

Search through all possible alignments

2. Choose an alignment $a = a_1 ... a_m$ for the source sentence

Target Position	1	2	3	4	5	6	7	8
Source Position	1	2	4	5	5	2	0	3

3. Translate each source word e_{a_j} into the target language

$$p(a|\mathbf{e}) = \frac{1}{n} \prod_{j=1}^m p(f_j|e_{a_j})$$

English	John	flies	the	river	river	flies	NULL	across
Alignment	1	2	4	5	5	2	0	3
Encoded	Jean	a	traversé	le	lac	à	la	nage

Order of action

Receiver

Decoding process in Model 1

$p(e|f) = 1e^{-15}$

For a particular English sentence $e = e_1..e_n$ of length n

$p(e)$

0	1	2	3	4	5
NULL	John	swam	across	the	lake

Search through all English sentences

1. Choose a length m for the target sentence (e.g $m = 8$)

$p(m|e) = \epsilon$

1	2	3	4	5	6	7	8
?	?	?	?	?	?	?	?

Search through all possible alignments

2. Choose an alignment $a = a_1 ... a_m$ for the source sentence

Target Position	1	2	3	4	5	6	7	8
Source Position	1	3	3	4	5	0	0	2

3. Translate each source word e_{a_j} into the target language

$p(a|e) = \frac{1}{n}$
 $\prod_{j=1}^m p(f_j|e_{a_j})$

English	John	across	across	the	lake	NULL	NULL	swam
Alignment	1	3	3	4	5	0	0	2
Encoded	Jean	a	traversé	le	lac	à	la	nage

Order of action
Receiver

Decoding process in Model 1

- Search space is huge
 - Presumably all “sentences” in English
 - English sentence length is unknown
 - All permutation of words in the vocabulary
 - Heuristics to reduce search space
 - Trade-off between translation accuracy and efficiency

Alignments

- The generative process explains only one way of generating a sentence pair
 - Each way corresponds to an alignment
- Total probability of the sentence pair is the sum of probability over all alignments

$$\Pr(\mathbf{f}|\mathbf{e}) = \sum_{\mathbf{a}} \Pr(\mathbf{f}, \mathbf{a}|\mathbf{e})$$

- Input: Parallel sentences 1...S in languages \mathbf{E} and \mathbf{F}
- But alignments are not known
- Goal: Learn the model $P(\mathbf{f}|\mathbf{e})$

Estimation of translation probability

- If we do not have ground-truth word-alignments, appeal to Expectation Maximization algorithm
 - Intuitively, guess the alignment based on the current translation probability first; and then update the translation probability

Training Algorithm

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

Repeat the E-step and M-step till $t(f|e)$ values converge

$c(f|e)$ is the
expected count
that f and e are
aligned

E-Step

- **for** each sentence in training corpus
 - **for** each f,e pair :
 Compute $c(f|e;f(s),e(s))$
 - Use $t(f|e)$ values from previous iteration

M-Step

- for each f,e pair: compute $t(f|e)$
- Use the $c(f|e)$ values computed in E-step

$$c(f|e; \mathbf{f}, \mathbf{e}) = \frac{t(f|e)}{t(f|e_0) + \dots + t(f|e_l)} \underbrace{\sum_{j=1}^m \delta(f, f_j)}_{\text{count of } f \text{ in } \mathbf{f}} \underbrace{\sum_{i=0}^l \delta(e, e_i)}_{\text{count of } e \text{ in } \mathbf{e}}$$

$$t(f|e) = \lambda_e^{-1} \sum_{s=1}^S c(f|e; \mathbf{f}^{(s)}, \mathbf{e}^{(s)}).$$
$$\lambda_e = \sum_{s=1}^S \sum_{f \text{ in Vocab}(\mathbf{F})} c(f|e; \mathbf{f}^{(s)}, \mathbf{e}^{(s)})$$

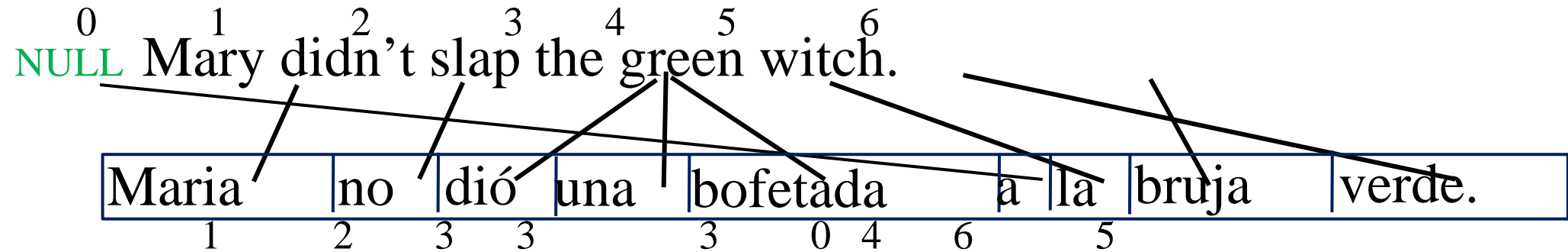
Other translation models

- IBM models 2-5 are more complex
 - Word order and string position of the aligned words
 - Phase-based translation in the source and target languages
 - Incorporate syntax or quasi-syntactic structures
 - Greatly reduce search space

What you should know

- Challenges in machine translation
 - Lexicon/syntactic/semantic divergences
- Statistical machine translation
 - Source-channel framework for statistical machine translation
 - Generative process
 - IBM model 1
 - Idea of word alignment

Sample IBM Model 1 Generation



Computing $P(F \mid E)$ in IBM Model 1

- Assume some length distribution $P(J \mid E)$
- Assume all alignments are equally likely. Since there are $(I + 1)^J$ possible alignments:

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

- Assume $t(f_x, e_y)$ is the probability of translating e_y as f_x , therefore:

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

- Determine $P(F \mid E)$ by summing over all alignments:

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

Decoding for IBM Model 1

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

$$\begin{aligned}\hat{A} &= \operatorname{argmax}_A P(F, A | E) \\ &= \operatorname{argmax}_A \frac{P(J | E)}{(I + 1)^J} \prod_{j=1}^J t(f_j, e_{a_j}) \\ &= \operatorname{argmax}_A \prod_{j=1}^J t(f_j, e_{a_j})\end{aligned}$$

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

$$a_j = \operatorname{argmax}_{0 \leq i \leq I} t(f_j, e_i) \quad 1 \leq j \leq J$$

HMM-Based Word Alignment

- IBM Model 1 assumes all alignments are equally likely and does not take into account *locality*:
 - If two words appear together in one language, then their translations are likely to appear together in the result in the other language.
- An alternative model of word alignment based on an HMM model **does** account for locality by making longer jumps in switching from translating one word to another less likely.

HMM Model

- Assumes the hidden state is the specific word occurrence e_i in E currently being translated (i.e. there are I states, one for each word in E).
- Assumes the observations from these hidden states are the possible translations f_j of e_i .
- Generation of F from E then consists of moving to the initial E word to be translated, generating a translation, moving to the next word to be translated, and so on.

Sample HMM Generation

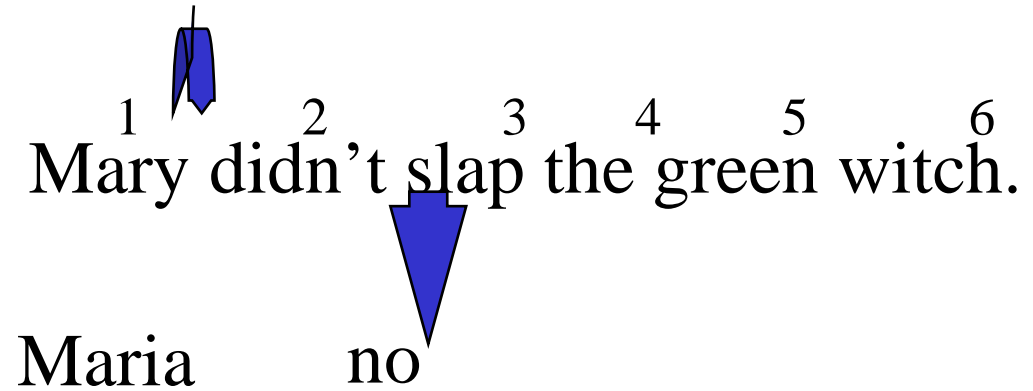


¹Mary² didn't³ slap⁴ the⁵ green⁶ witch.

Maria

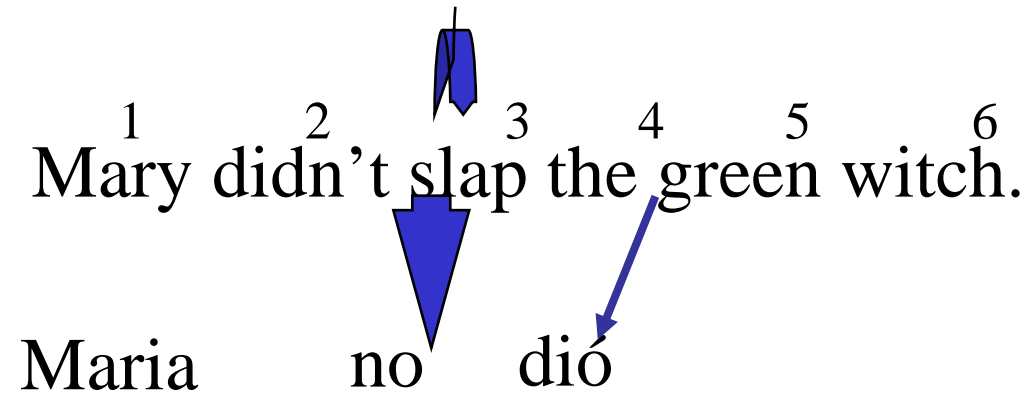
Sample HMM Generation

1 2 3 4 5 6
Mary didn't slap the green witch.
Maria no



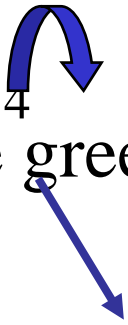
The diagram illustrates the generation of a sentence from a Hidden Markov Model (HMM). The sentence "Mary didn't slap the green witch." is shown with phonetic annotations. Above the words, numbers 1 through 6 are placed. A blue arrow points from the number 1 to the word "Mary". Another blue arrow points from the number 2 to the word "no". The word "Maria" is written below "Mary", and "no" is written below "no".

Sample HMM Generation



Sample HMM Generation

1 2 3 4 5 6
Mary didn't slap the green witch.
Maria no dió una



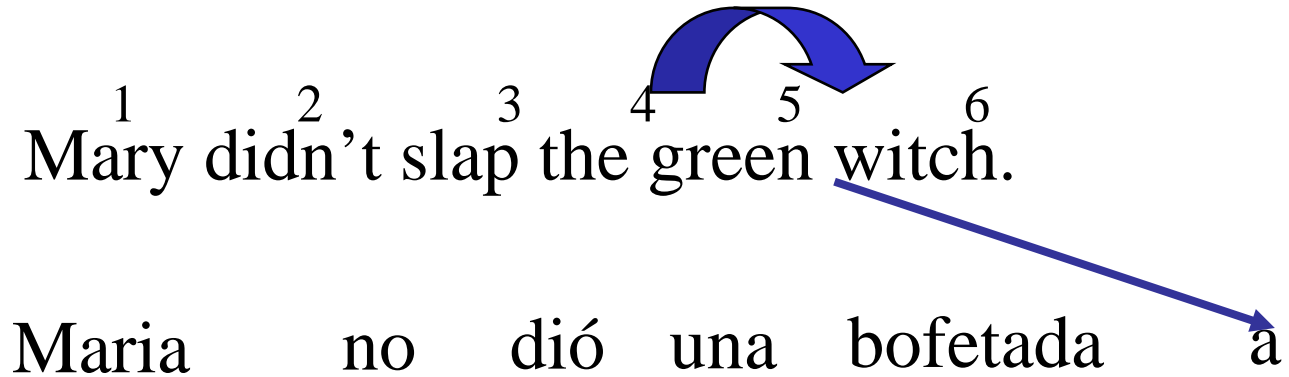
Sample HMM Generation

¹ Mary ² didn't ³ slap ⁴ the ⁵ green ⁶ witch.



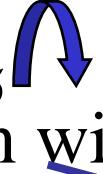
Maria no dió una bofetada

Sample HMM Generation



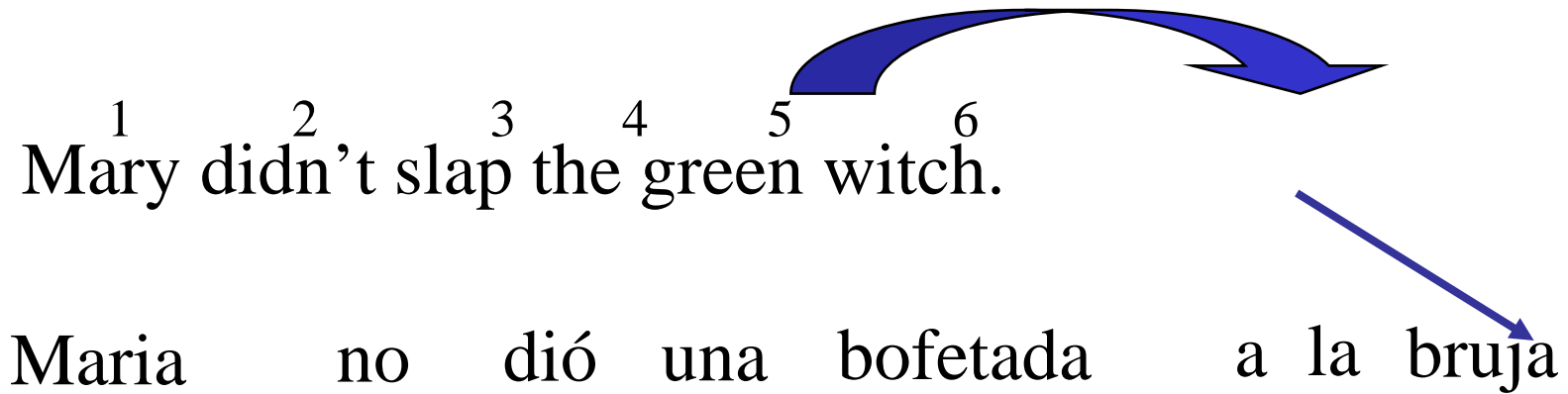
Sample HMM Generation

¹ Mary ² didn't ³ slap ⁴ the ⁵ green ⁶ witch.

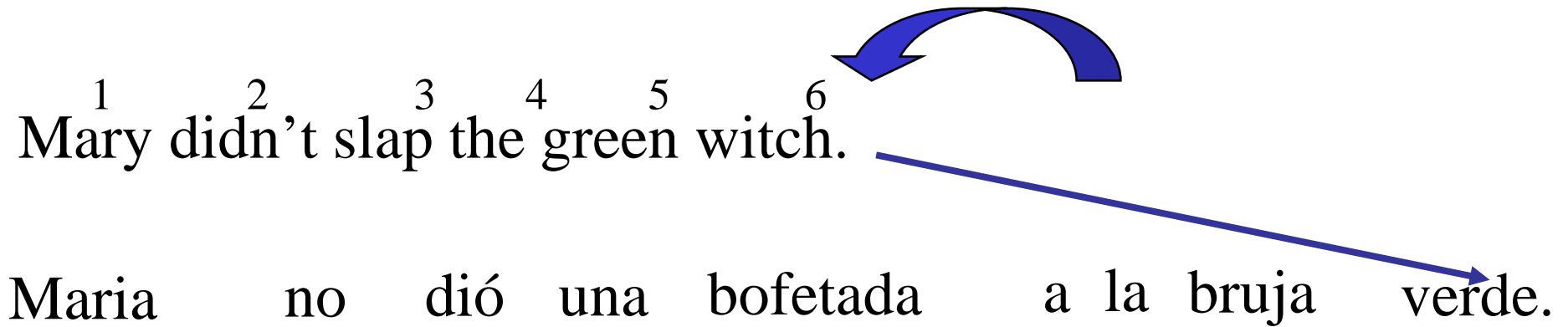
A blue curved arrow originates from the word 'green' (position 5) and points to the word 'witch' (position 6), indicating a transition in the HMM generation process.

Maria no dió una bofetada a la

Sample HMM Generation



Sample HMM Generation



Sample HMM Generation

¹Mary ²didn't ³slap ⁴the ⁵green ⁶witch.

Maria no dió una bofetada a la bruja verde.

HMM Parameters

- Transition and observation parameters of states for HMMs for all possible source sentences are “tied” to reduce the number of free parameters that have to be estimated.
- **Observation probabilities:** $b_j(f_i) = P(f_i \mid e_j)$ the same for all states representing an occurrence of the same English word.
- **State transition probabilities:** $a_{ij} = s(j-i)$ the same for all transitions that involve the same *jump width* (and direction).

Computing $P(F \mid E)$ in the HMM Model

- Given the observation and state-transition probabilities, $P(F \mid E)$ (observation likelihood) can be computed using the standard ***forward algorithm*** for HMMs.

Decoding for the HMM Model

- Use the standard ***Viterbi algorithm*** to efficiently compute the most likely alignment (i.e. most likely state sequence).

Training Word Alignment Models

- Both the IBM model 1 and HMM model can be trained on a parallel corpus to set the required parameters.
- For supervised (hand-aligned) training data, parameters can be estimated directly using frequency counts.
- For unsupervised training data, EM can be used to estimate parameters, e.g. Baum-Welch for the HMM model.

Sketch of EM Algorithm for Word Alignment

Randomly set model parameters.

(making sure they represent legal distributions)

Until converge (i.e. parameters no longer change) do:

E Step: Compute the probability of all possible alignments of the training data using the current model.

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

Note: Use dynamic programming (as in Baum-Welch) to avoid explicitly enumerating all possible alignments

Sample EM Trace for Alignment

(IBM Model 1 with no NULL Generation)

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

$P(A, F | E)$

green house
casa verde

$$1/3 \times 1/3 = 1/9$$

~~green house~~
~~casa verde~~

$$1/3 \times 1/3 = 1/9$$

the house
la casa

$$1/3 \times 1/3 = 1/9$$

~~the house~~
~~la casa~~

$$1/3 \times 1/3 = 1/9$$

Normalize
to get
 $P(A | F, E)$

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

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

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

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

Example cont.

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

green
house
the

	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 $P(f | e)$

green
house
the

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

Example cont.


Translation
Probabilities

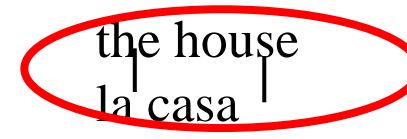
green
house
the

	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
 $1/2 \times 1/4 = 1/8$

 ~~green house~~
~~casa verde~~
 $1/2 \times 1/2 = 1/4$

 ~~the house~~
~~la casa~~
 $1/2 \times 1/2 = 1/4$

~~the house~~
~~la casa~~
 $1/2 \times 1/4 = 1/8$

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

Decoding

- Goal is to find a translation that maximizes the product of the translation and language models.

$$\operatorname{argmax}_{E \in \text{English}} P(F | E) P(E)$$

- Cannot explicitly enumerate and test the combinatorial space of all possible translations.
- The optimal decoding problem for all reasonable model's (e.g. IBM model 1) is NP-complete.
- Heuristically search the space of translations using A*, beam-search, etc. to approximate the solution to this difficult optimization problem.

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. [redacted] [redacted]

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.

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

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

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.

$$c = 6, \quad r = 7, \quad BP = e^{(1-7/6)} = 0.846$$
$$BLEU = 0.846 \times 0.408 = 0.345$$

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

Best Ref: Mary did not smack the green witch.

$$c = 10, \quad r = 7, \quad BP = 1$$
$$BLEU = 1 \times 0.558 = 0.558$$

BLEU Score Issues

- BLEU has been shown to correlate with human evaluation when comparing outputs from different SMT systems.
- However, it 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:*
 $X \rightarrow X$ 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



Input: 俄亥俄州的首府是甚麼？

$X \rightarrow X$ 是甚麼 / What is X

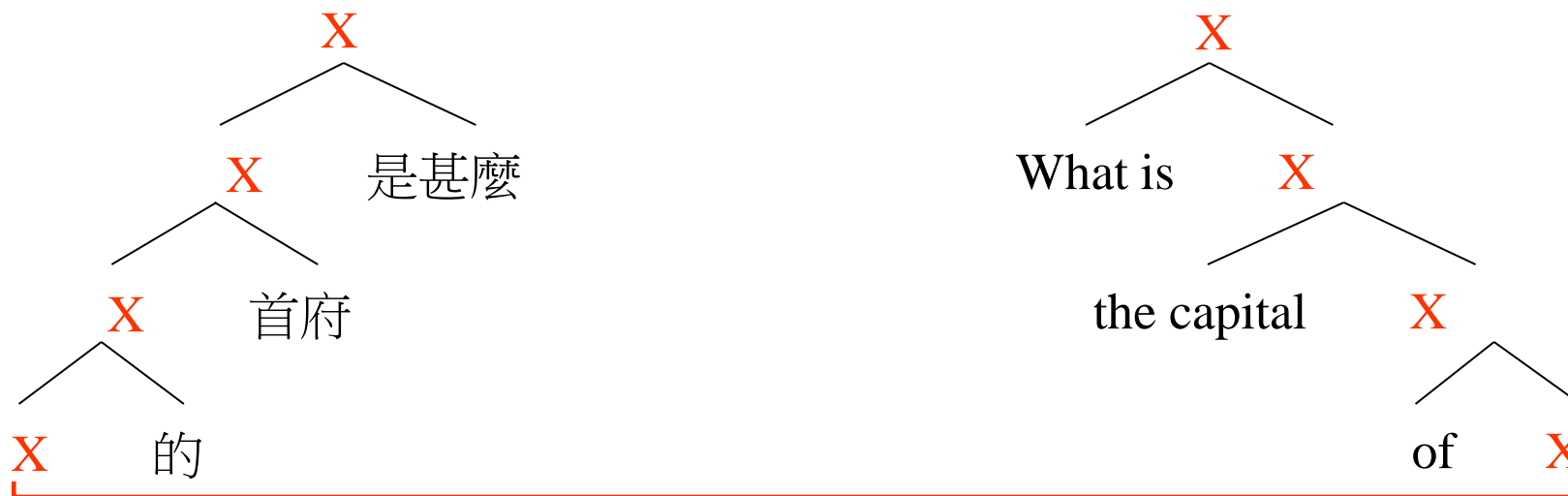
Syntax-Based MT Example



Input: 俄亥俄州的首府是甚麼？

X → X 首府 / the capital X

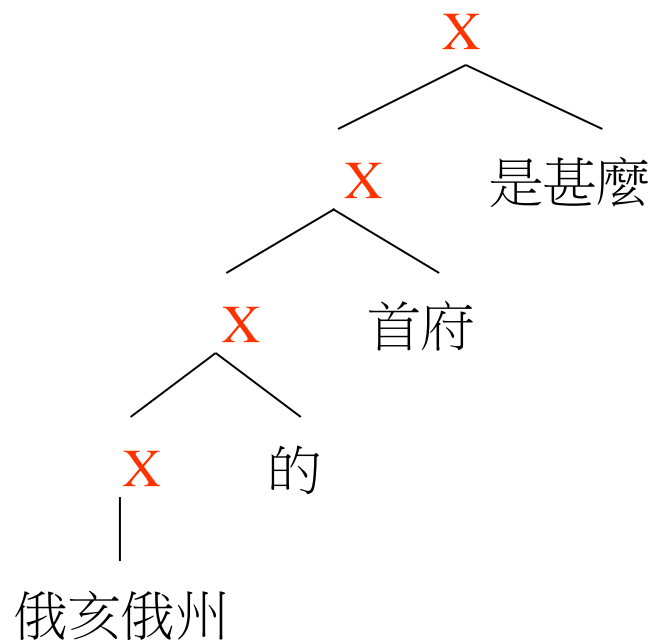
Syntax-Based MT Example



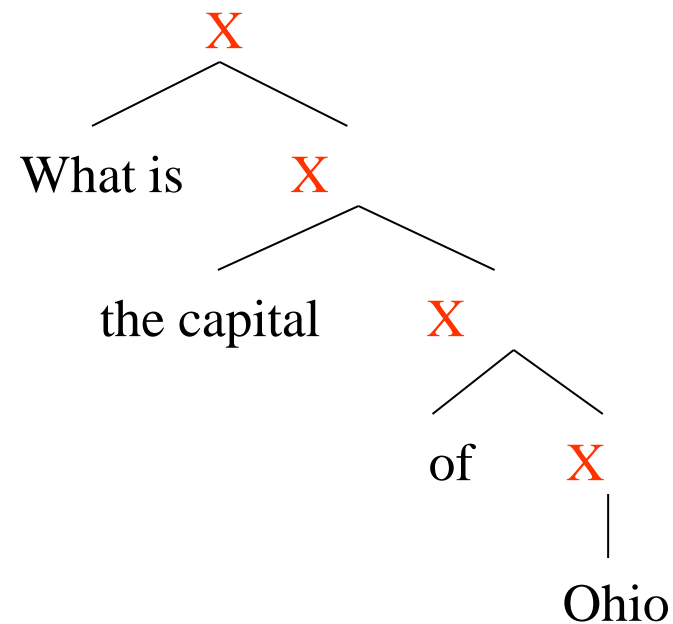
Input: 俄亥俄州的首府是甚麼？

$X \rightarrow X \text{ 的 / of } X$

Syntax-Based MT Example

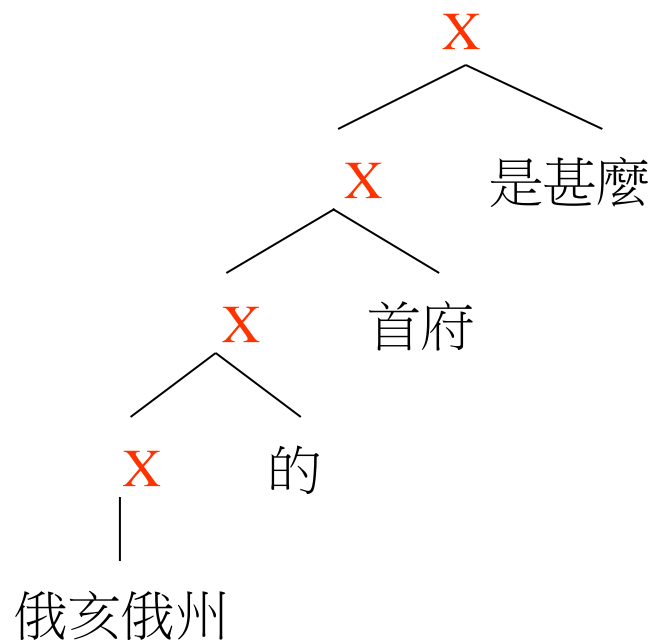


Input: 俄亥俄州的首府是甚麼？

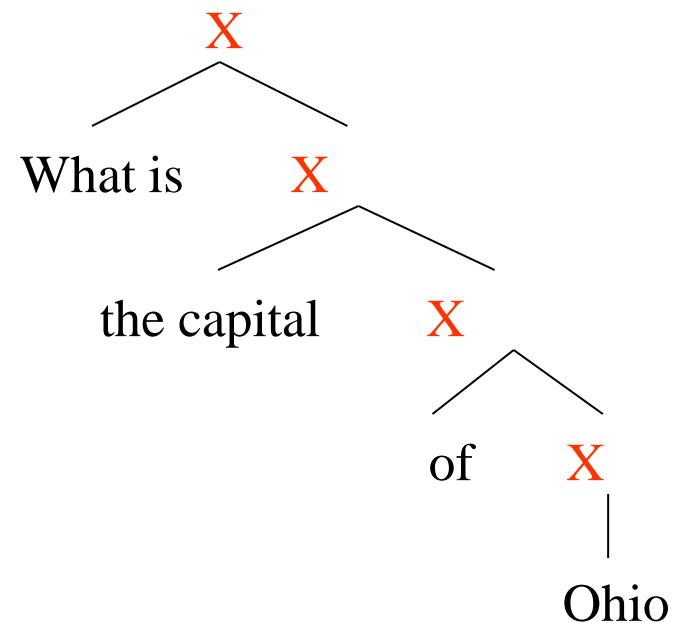


$X \rightarrow$ 俄亥俄州 / Ohio

Syntax-Based MT Example



Input: 俄亥俄州的首府是甚麼？



Output: What is the capital of Ohio?

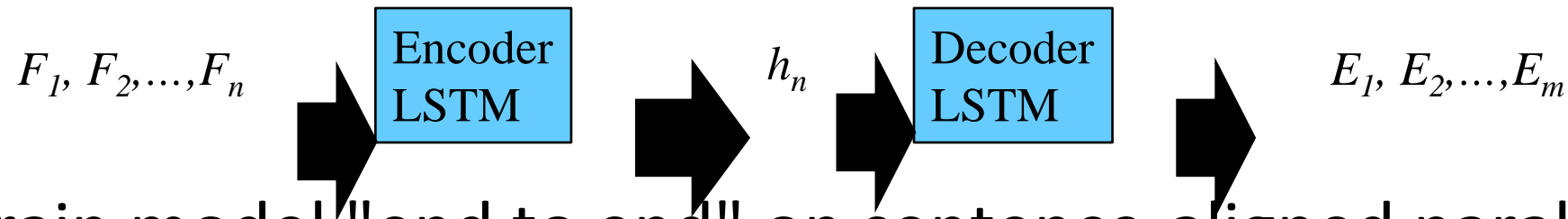
Synchronous Derivations and Translation Model

- Need to make a probabilistic version of synchronous grammars to create a translation model for $P(F \mid E)$.
- 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.

$$\lambda^* = \arg \max_{\lambda} \sum_j \log \text{Pr}_{\lambda}(\mathbf{f}_j | \mathbf{e}_j)$$

Neural Machine Translation (NMT)

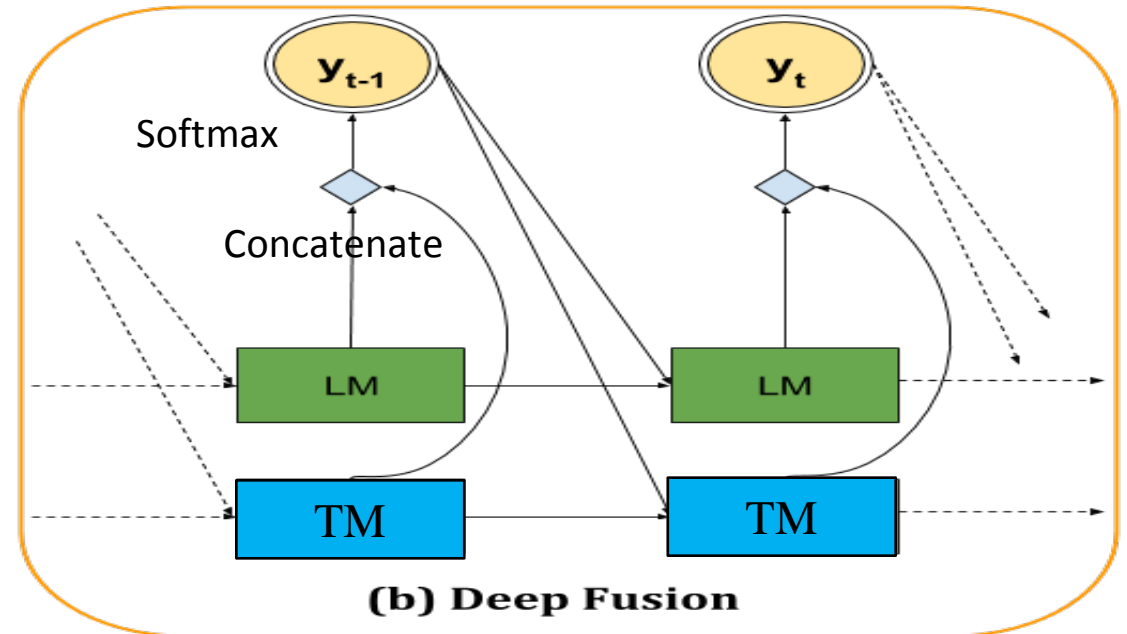
- Encoder/Decoder framework maps sentence in source language to a "deep vector" then another LSTM maps this vector to a sentence in the target language



- 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.