

# Einführung in die Computerlinguistik

## Machine Translation

Yihong Liu

Center for Information and Language Processing

2024-15-01

# Outline

- 1 Noisy channel model
- 2 Machine translation
- 3 Language models



Fred Jelinek was a researcher in information theory, automatic speech recognition, and natural language processing. He regarded speech recognition as an information theory problem—a noisy channel...

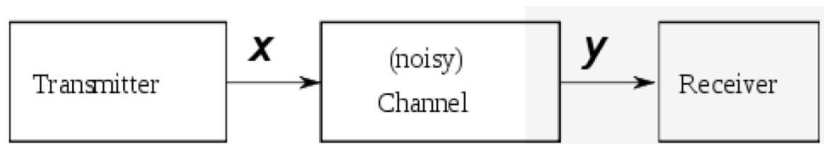
- Every time I fire a linguist, the performance of the speech recognizer goes up.

# IBM Watson approach to NLP

- sequence model
- in most cases: given an observation or **evidence**, select the most likely sequence that caused the observation
- We will only consider **word** sequences for now.

$$\begin{aligned} & \operatorname{argmax}_{\text{word-sequence}} P(\text{word-sequence}|\text{evidence}) \\ = & \operatorname{argmax}_{\text{word-sequence}} \frac{P(\text{evidence}|\text{word-sequence})P(\text{word-sequence})}{P(\text{evidence})} \\ = & \operatorname{argmax}_{\text{word-sequence}} P(\text{evidence}|\text{word-sequence}) \quad P(\text{word-sequence}) \end{aligned}$$

# Noisy channel

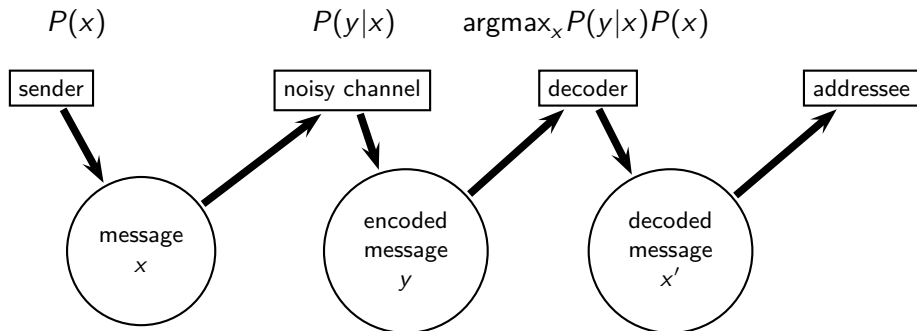


Well-known examples of applications of noisy channel model?

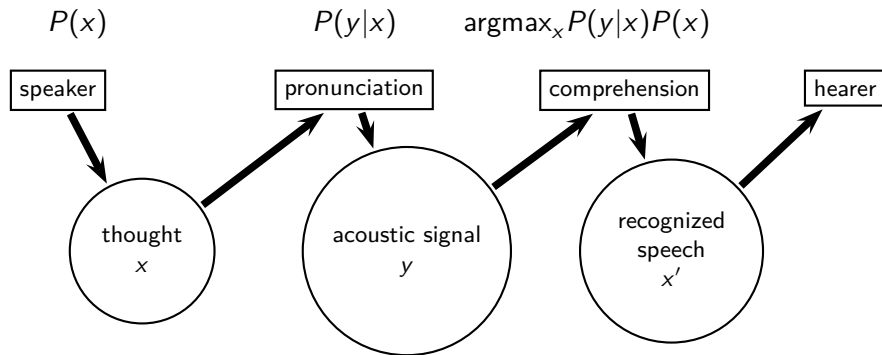
# Decode 788884278



# Noisy channel: Information theory / telecommunications

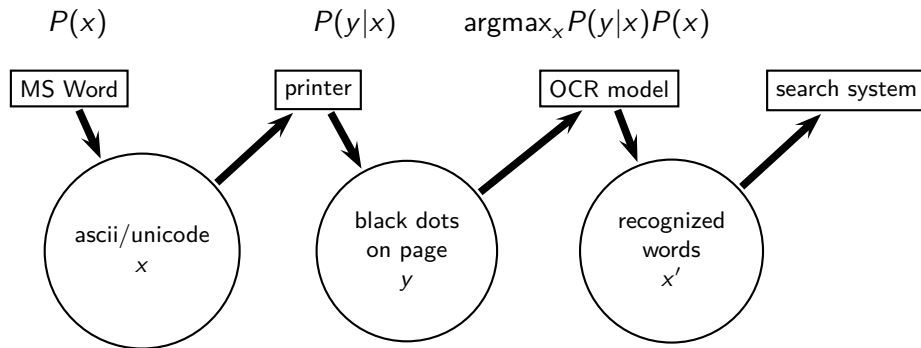


# Noisy channel: Speech recognition





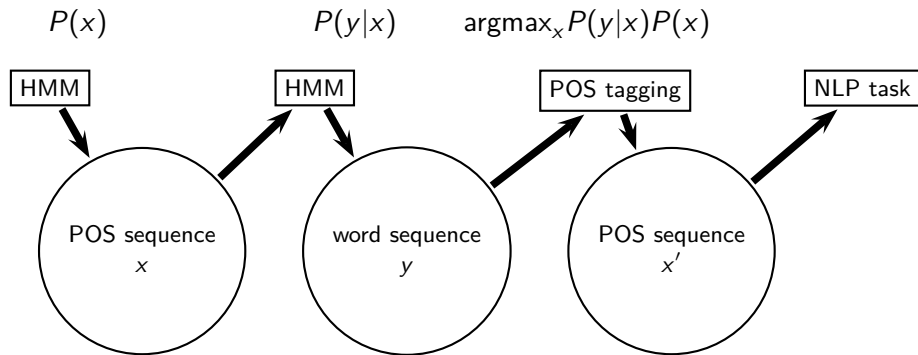
# Noisy channel: Optical character recognition



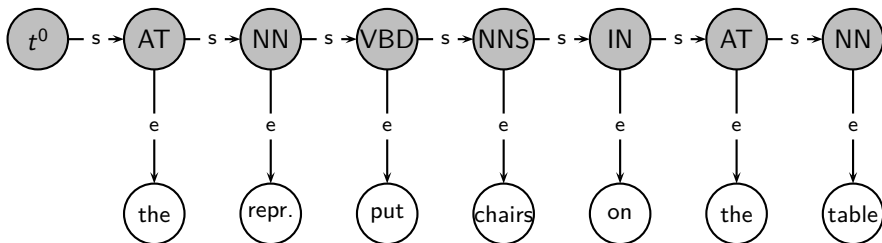
# Part-of-speech tagging

- Given a sequence of words (a sentence), how do we compute the corresponding (disambiguated) part-of-speech sequence?
- Example:
  - Input: “the representative put chairs on the table”
  - Output: “AT NN VBD NNS IN AT NN”
- $t_{1,n} = \operatorname{argmax}_{t_{1,n}} P(t_{1,n}|w_{1,n}) = \operatorname{argmax}_{t_{1,n}} P(w_{1,n}|t_{1,n})P(t_{1,n})$

# Noisy channel: Part-of-speech tagging



# Noisy channel: Part-of-speech tagging



# IBM Watson approach to NLP

- sequence model
- in most cases: given an observation or **evidence**,  
select the most likely sequence that caused the observation
- We will only consider **word** sequences for now.

$$\begin{aligned} & \operatorname{argmax}_{\text{word-sequence}} P(\text{word-sequence}|\text{evidence}) \\ = & \operatorname{argmax}_{\text{word-sequence}} \frac{P(\text{evidence}|\text{word-sequence})P(\text{word-sequence})}{P(\text{evidence})} \\ = & \operatorname{argmax}_{\text{word-sequence}} P(\text{evidence}|\text{word-sequence}) \quad P(\text{word-sequence}) \end{aligned}$$

# Classical approach to speech recognition

$$\begin{aligned} & \operatorname{argmax}_{\text{word-sequence}} P(\text{word-sequence}|\text{evidence}) \\ = & \operatorname{argmax}_{\text{word-sequence}} \frac{P(\text{evidence}|\text{word-sequence})P(\text{word-sequence})}{P(\text{evidence})} \\ = & \operatorname{argmax}_{\text{word-sequence}} P(\text{evidence}|\text{word-sequence}) P(\text{word-sequence}) \end{aligned}$$

- word sequence: sequence of words
- evidence: acoustic signal
- $P(\text{evidence}|\text{word-sequence})$ : a model of how humans translate a sequence of (written) words into acoustics
- The whole framework is a generative model (vs. discriminative model)

# Classical approach to optical character recognition

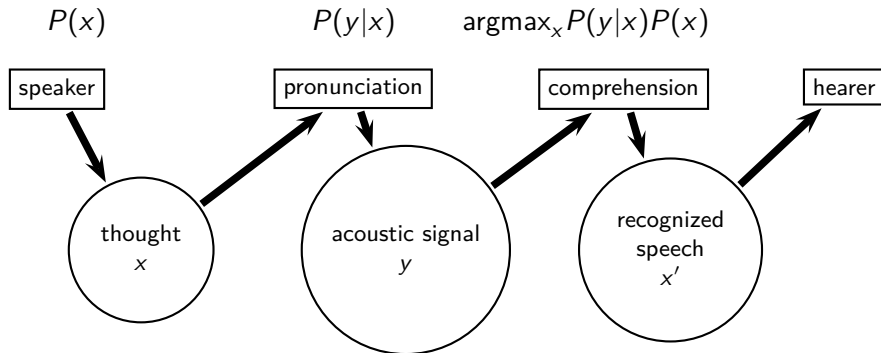
$$\begin{aligned} & \operatorname{argmax}_{\text{word-sequence}} P(\text{word-sequence}|\text{evidence}) \\ = & \operatorname{argmax}_{\text{word-sequence}} \frac{P(\text{evidence}|\text{word-sequence})P(\text{word-sequence})}{P(\text{evidence})} \\ = & \operatorname{argmax}_{\text{word-sequence}} P(\text{evidence}|\text{word-sequence}) P(\text{word-sequence}) \end{aligned}$$

- word sequence: sequence of words
- evidence: image
- $P(\text{evidence}|\text{word-sequence})$ : a model of how a machine (e.g., a desktop printer) translates a sequence of words into printed letters/symbols
- The whole framework is a generative model (vs. discriminative model)

# Exercise: Noisy channel model for machine translation?

speech

- word sequence: sequence of words
- evidence: acoustic signal
- $P(\text{evidence}|\text{word-sequence})$ : a model of how humans translate a sequence of (written) words into acoustics



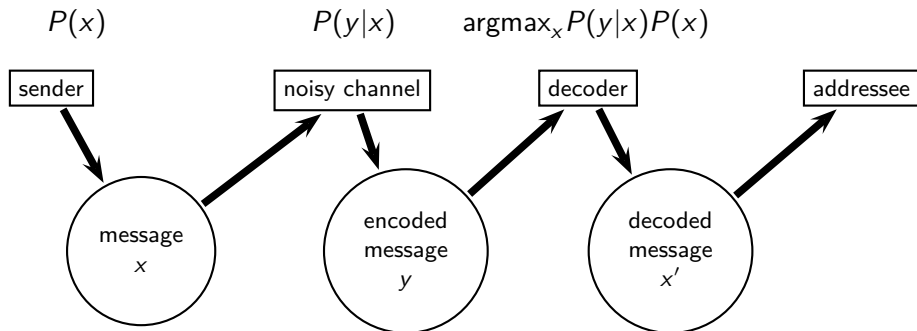


# Classical approach to machine translation (French→English)

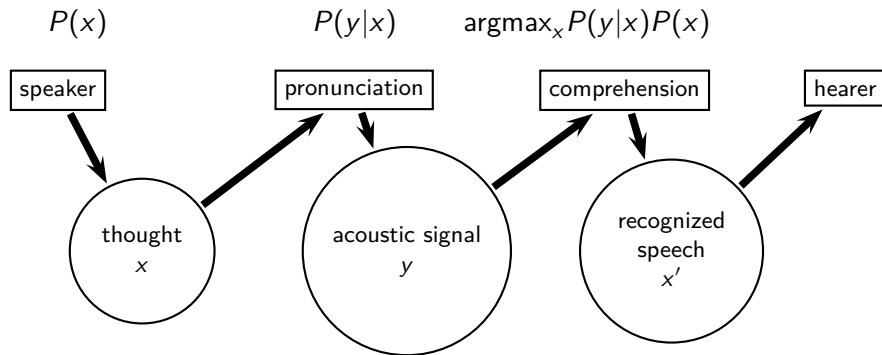
$$\begin{aligned} & \operatorname{argmax}_{\text{word-sequence}} P(\text{word-sequence}|\text{evidence}) \\ = & \operatorname{argmax}_{\text{word-sequence}} \frac{P(\text{evidence}|\text{word-sequence})P(\text{word-sequence})}{P(\text{evidence})} \\ = & \operatorname{argmax}_{\text{word-sequence}} P(\text{evidence}|\text{word-sequence}) P(\text{word-sequence}) \end{aligned}$$

- word sequence: sequence of English words
- evidence: sequence of French words
- $P(\text{evidence}|\text{word-sequence})$ : a model of how humans translate a sequence of English words into a sequence of French words

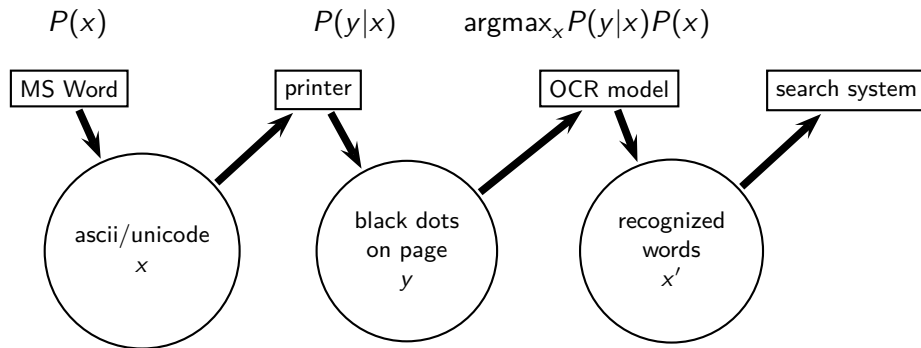
# Noisy channel: Information theory / telecommunications



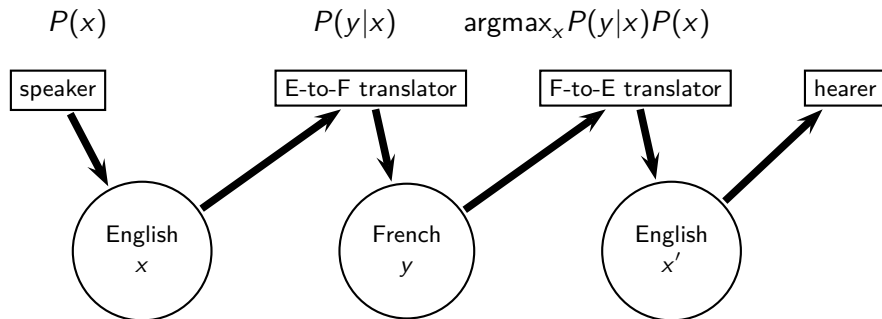
# Noisy channel: Speech recognition



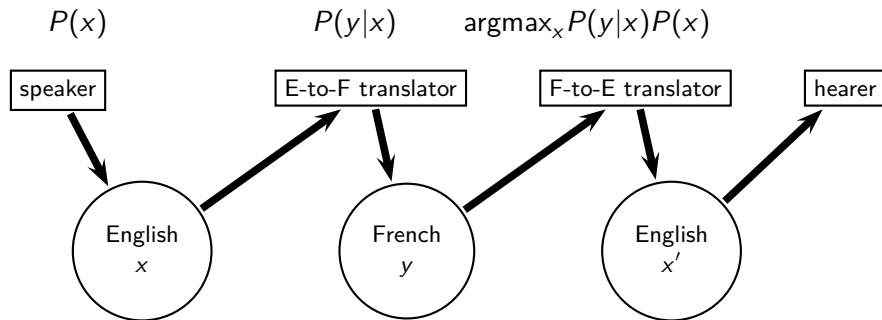
# Noisy channel: Optical character recognition



# Noisy channel: French-to-English machine translation



# Noisy channel: French-to-English machine translation

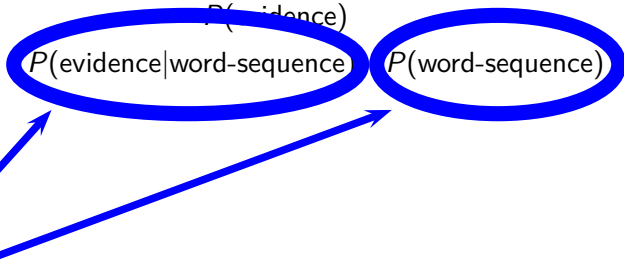


# The two key components of the model

$$\begin{aligned} & \operatorname{argmax}_{\text{word-sequence}} P(\text{word-sequence}|\text{evidence}) \\ = & \operatorname{argmax}_{\text{word-sequence}} \frac{P(\text{evidence}|\text{word-sequence})P(\text{word-sequence})}{P(\text{evidence})} \\ = & \operatorname{argmax}_{\text{word-sequence}} P(\text{evidence}|\text{word-sequence}) P(\text{word-sequence}) \end{aligned}$$

translation model

language model



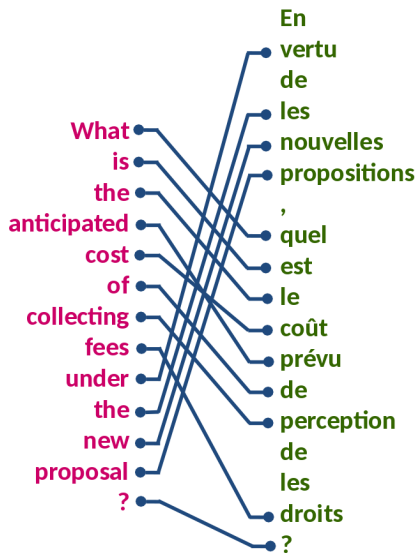
# How to build a translation model

- Find a **parallel corpus** – a body of text where each sentence is available in two or more languages
- IBM Watson used the Canadian Hansards, the proceedings of the Canadian Parliament.
- Compute a **word alignment** for the parallel corpus (next slide)
- Estimate a **translation model** from the word alignment (that is, the model that models how humans generate French sentences from English sentences)



- Our model is a **generative** model: The French sentence (what we observe) is generated based on the English sentence.
- Every French word is “caused” by an English word.
- causation = alignment
- But many French words are **not aligned**, i.e., they have no plausible English word they correspond to.
- To cover these unaligned French words, we introduce the “empty cept”  $e_0$ .
- The empty cept  $e_0$  is an **artificial** English word that all unaligned French words are aligned with.
- Now every French word is “caused” by an English word.

# Exercise: Estimating word translation probabilities



Estimate:

$$P(e_i | \text{nouvelles})$$

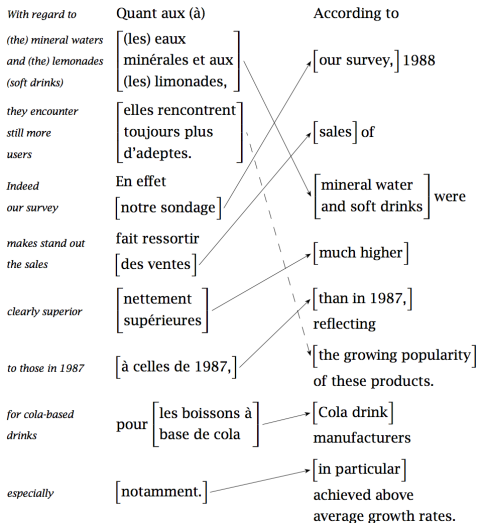
$$P(f_j | \text{fees})$$

$$P(f_j | \text{the})$$

$$P(f_j | e_0)$$

By counting!

# “Linguistic” word/phrase alignment of a parallel corpus

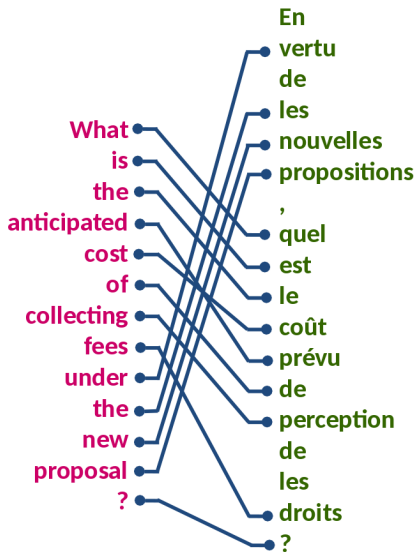


# Basic translation model

$$P(f|e) \propto \sum_{a_1=0}^l \cdots \sum_{a_m=0}^l P(\langle a_1, \dots, a_m \rangle) \prod_{j=1}^m P(f_j | e_{a_j})$$

- $e$ : English sentence,  $e_i$ :  $i^{th}$  word in  $e$
- $l$ : length of English sentence
- $f$ : French sentence,  $f_j$ :  $j^{th}$  word in  $f$
- $m$ : length of French sentence
- $e_{a_j}$  is the English word that  $f_j$  is aligned with – this assumes that the alignment is a (total) function:  
 $a : \{1, 2, \dots, m\} \mapsto \{0, 1, \dots, l\}$
- There is a special word  $e_0$ , the empty cept, that all unaligned French words are aligned to.
- $P(f_j | e_{a_j})$  is the probability of  $e_{a_j}$  being translated as  $f_j$ .
- $P(\langle a_1, \dots, a_m \rangle)$  is the probability of alignment (an ordered list of each French word ( $f_j$  being aligned with  $e_{a_j}$ )).

# Exercise: Estimating word translation probabilities



Estimate:

$$P(e_i | \text{nouvelles})$$

$$P(f_j | \text{fees})$$

$$P(f_j | \text{the})$$

$$P(f_j | e_0)$$

By counting!

# Basic translation model

$$P(f|e) \propto \sum_{a_1=0}^l \cdots \sum_{a_m=0}^l P(\langle a_1, \dots, a_m \rangle) \prod_{j=1}^m P(f_j | e_{a_j})$$

- $e$ : English sentence,  $e_i$ :  $i^{th}$  word in  $e$
- $l$ : length of English sentence
- $f$ : French sentence,  $f_j$ :  $j^{th}$  word in  $f$
- $m$ : length of French sentence
- $e_{a_j}$  is the English word that  $f_j$  is aligned with – this assumes that the alignment is a (total) function:  
 $a : \{1, 2, \dots, m\} \mapsto \{0, 1, \dots, l\}$
- There is a special word  $e_0$ , the empty cept, that all unaligned French words are aligned to.
- $P(f_j | e_{a_j})$  is the probability of  $e_{a_j}$  being translated as  $f_j$ .
- $P(\langle a_1, \dots, a_m \rangle)$  is the probability of alignment (an ordered list of each French word ( $f_j$  being aligned with  $e_{a_j}$ )).

# Formalization of alignment

$e_0$	$e_1$	$e_2$
empty cept	they	descended

$f_1$	$f_2$	$f_3$
runter	gingen	sie

$a_1$	$a_2$	$a_3$	$a_1$	$a_2$	$a_3$	$a_1$	$a_2$	$a_3$
0	0	0	1	0	0	2	0	0
0	0	1	1	0	1	2	0	1
0	0	2	1	0	2	2	0	2
0	1	0	1	1	0	2	1	0
0	1	1	1	1	1	2	1	1
0	1	2	1	1	2	2	1	2
0	2	0	1	2	0	2	2	0
0	2	1	1	2	1	2	2	1
0	2	2	1	2	2	2	2	2

# Basic translation model

$$P(f|e) \propto \sum_{a_1=0}^l \cdots \sum_{a_m=0}^l P(\langle a_1, \dots, a_m \rangle) \prod_{j=1}^m P(f_j | e_{a_j})$$

- $e$ : English sentence,  $e_i$ :  $i^{th}$  word in  $e$
- $l$ : length of English sentence
- $f$ : French sentence,  $f_j$ :  $j^{th}$  word in  $f$
- $m$ : length of French sentence
- $e_{a_j}$  is the English word that  $f_j$  is aligned with – this assumes that the alignment is a (total) function:  
 $a : \{1, 2, \dots, m\} \mapsto \{0, 1, \dots, l\}$
- There is a special word  $e_0$ , the empty cept, that all unaligned French words are aligned to.
- $P(f_j | e_{a_j})$  is the probability of  $e_{a_j}$  being translated as  $f_j$ .
- $P(\langle a_1, \dots, a_m \rangle)$  is the probability of alignment (an ordered list of each French word ( $f_j$  being aligned with  $e_{a_j}$ )).



What's bad about this model? What type of linguistic phenomenon will not be translated correctly?

# What's bad about this model

- Collocations, noncompositional combinations: “piece of cake”
  - Assumption violated: Each English word generates German translations **independent** of the other words.
- Compounds: “Kirschkuchen” vs. “cherry pie”
  - Assumption violated: For each German/French word there is a **single** English word responsible for it.
- Unlikely alignments: “siehst Du” vs. “(do) you see”
  - Assumption violated: The probability of a particular alignment **is independent of the words**.

## What's bad about this model (2)

- Morphology: “Kind” – “Kindes”
- Gender and case
- Syntax

Possible solution: Neural machine translation with subword tokenization

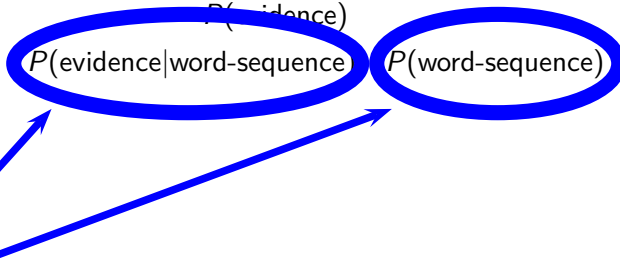
# Google Translate

# The two key components of the model

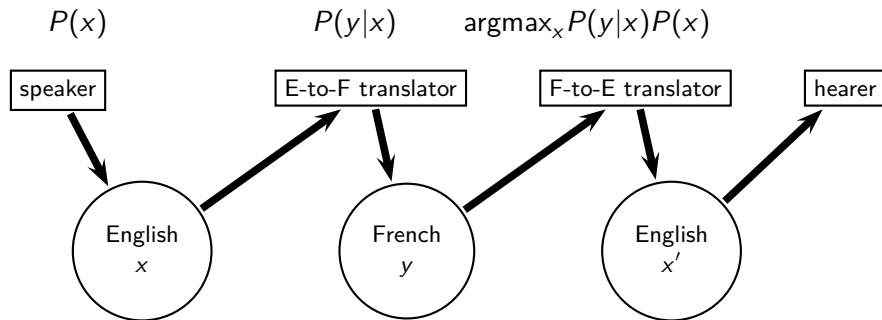
$$\begin{aligned} & \operatorname{argmax}_{\text{word-sequence}} P(\text{word-sequence}|\text{evidence}) \\ = & \operatorname{argmax}_{\text{word-sequence}} \frac{P(\text{evidence}|\text{word-sequence})P(\text{word-sequence})}{P(\text{evidence})} \\ = & \operatorname{argmax}_{\text{word-sequence}} P(\text{evidence}|\text{word-sequence}) P(\text{word-sequence}) \end{aligned}$$

translation model

language model



# Noisy channel: French-to-English machine translation



# Why the language model is important

- Classical example from speech recognition
- The following two are almost indistinguishable acoustically.
- “wreck a nice beach”
- “recognize speech”
- If we had only the translation model  $P(y|x)$ , then we would not be able to make a good decision.
- We need the language model for this decision.
- $P(\text{“wreck a nice beach”}) \ll P(\text{“recognize speech”})$
- We’ll choose “recognize speech” based on this.

# Bigram language model

$$P(w_{1,2,\dots,n}) = \prod_{i=1}^n P(w_i|w_{i-1})$$

- Key problem: How do we estimate the parameters?
- Specifically, how  $P(w_i|w_{i-1})$ ?



# Maximum likelihood = Relative frequency

$$P_{ML}(w_2|w_1) = \frac{C(w_1 w_2)}{C(w_1)}$$

where  $C(e)$  is the number of times the event  $e$  occurred in the training set.

Example:

$$p_{ML}(\text{be}|\text{would}) = \frac{C(\text{would be})}{C(\text{would})} = \frac{18454}{83735} \approx 0.22$$

Quiz: what is  $\sum_w p_{ML}(w|\text{would})$ ?

# Why maximum likelihood does not work

- Suppose that “Dr.” and “Cooper” are frequent in our corpus. Frequency of “Dr.” = 10000
- But suppose that the sequence “Dr. Cooper” does not occur in the corpus.
- What is the maximum likelihood estimate of  $P(\text{Cooper}|\text{Dr.})$ ?

- 

$$P_{ML}(\text{Cooper}|\text{Dr.}) = \frac{C(\text{Dr. Cooper})}{C(\text{Dr.})} = \frac{0}{10000} = 0$$

- This means that in machine translation, any English sentence containing “Dr. Cooper” would be **deemed impossible** and could not be output by the translator.
- This problem is called **sparseness**.
- Ideally, we would need knowledge about events and their probability **that never occurred in our training corpus**.

## Laplace = Add-one smoothing

$$P_L(w_2|w_1) = \frac{C(w_1 w_2) + 1}{C(w_1) + |V|}$$

where  $C(e)$  is the number of times the event  $e$  occurred in the training set,  $V$  is the vocabulary of the training set and  $w_{i,j}$  is the sequence of words  $w_i, w_{i+1}, \dots, w_{j-1}, w_j$ .

Better estimator:

$$P_L(\text{Cooper}|\text{Dr.}) = \frac{0 + 1}{10000 + 256873} \approx 0.0000037 > 0$$

So now our machine translation system has a chance of finding a good English translation that contains the phrase “Dr. Cooper”.

the three women saw the small mountain behind the large mountain

Compute maximum likelihood and laplace estimates for:  
 $P(\text{three}|\text{the})$  and  $P(\text{saw}|\text{the})$

- Noisy channel model
- Translation models
- Estimation of translation models
- Language models
- Estimation of language models

- $P(e)$
- $P(f|e)$
- empty cept
- $\operatorname{argmax}_e P(f|e)P(e)$

Credits: Ivan Habernal, Alexander Fraser, Hinrich Schütze