



אוניברסיטת בר-אילן
Bar-Ilan University

89688: Statistical Machine Translation

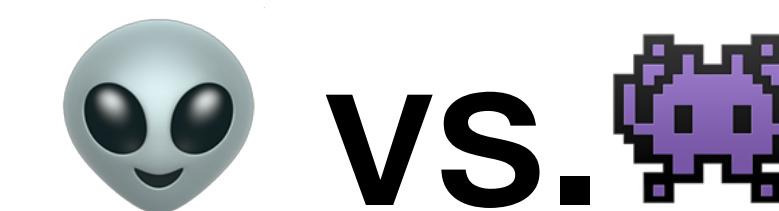
Lecture 3: Statistical Methods, IBM models and the EM algorithm

April 2020

Roee Aharoni
Computer Science Department
Bar Ilan University

Based in part on slides from [Edinburgh University's MT class](#) and by Kevin Knight

How can we learn to translate?



1a. ok-voon ororok sprok .
| | | |

1b. at-voon bichat dat .

2a. ok-drubel ok-voon anok plok sprok .
| | | | \ \

2b. at-drubel at-voon pippat rrat dat .

3a. erok sprok izok hihok ghirok .
| | | X | /

3b. totat dat arrat vat hilat .

4a. ok-voon anok drok brok jok .
| \ \ /

4b. at-voon krat pippat sat lat .

5a. wiwok farok izok stok .
| /

5b. totat jjat quat cat .

6a. lalok sprok izok jok stok .
| \ / \

6b. wat dat krat quat cat .

7a. lalok farok ororok lalok sprok izok enemok .
| / \ / \ / \ /

7b. wat jjat bichat wat dat vat eneat .

8a. lalok brok anok plok nok .
| \ / /

8b. iat lat pippat rrat nnat .

9a. wiwok nok izok kantok ok-yurp .
| \ / \ /

9b. totat nnat quat oloat at-yurp .

10a. lalok mok nok yorok ghirok clok .
| \ /

10b. wat nnat gat mat bat hilat .

How can we learn to translate?

How can we learn to translate?

- What can we learn from?

How can we learn to translate?

- What can we learn from?
 - Parallel Corpora (human translations)

How can we learn to translate?

- What can we learn from?
 - Parallel Corpora (human translations)
 - Monolingual corpora (books, wikipedia, the web...)

How can we learn to translate?

- What can we learn from?
 - Parallel Corpora (human translations)
 - Monolingual corpora (books, wikipedia, the web...)
- What do we learn?

How can we learn to translate?

- What can we learn from?
 - Parallel Corpora (human translations)
 - Monolingual corpora (books, wikipedia, the web...)
- What do we learn?

Translation dictionary:

anok	- pippat
erok	- total
ghirok	- hilat
hihok	- arrat
izok	- vat
ok-drubel	- at-drubel

ok-yurp	- at-yurp
ok-voon	- at-voon
ororok	- bichat
plok	- rrat
sprok	- dat
zanzanok	- zanzanat

Translation
Model

How can we learn to translate?

- What can we learn from?
 - Parallel Corpora (human translations)
 - Monolingual corpora (books, wikipedia, the web...)
- What do we learn?

Translation dictionary:

anok - pippat
erok - total
ghirok - hilat
hihok - arrat
izok - vat
ok-drubel - at-drubel

ok-yurp - at-yurp
ok-voon - at-voon
ororok - bichat
plok - rrat
sprok - dat
zanzanok - zanzanat

Translation
Model

Word pair counts:

1 . erok
7 . lalok
2 . ok-drubel
2 . ok-voon
3 . wiwok
1 anok drok
1 anok ghirok

1 hihok yorok
1 izok enemok
2 izok hihok
1 izok jok
1 izok kantok
1 izok stok
1 izok vok

Language
Model

Let's try to formalize this...

Let's try to formalize this...

- Learning/Training Phase

```
def learn(parallel_data):  
    # do something  
    return parameters
```

$$L : (\Sigma_f^* \times \Sigma_e^*)^* \rightarrow \Theta$$

Let's try to formalize this...

- Learning/Training Phase

```
def learn(parallel_data):  
    # do something  
    return parameters
```

$$L : (\Sigma_f^* \times \Sigma_e^*)^* \rightarrow \Theta$$

- Inference Phase

```
def translate(French, parameters):  
    # do something  
    return English
```

$$T : \Sigma_f^* \times \Theta \rightarrow \Sigma_e^*$$

Let's try to formalize this...

- Learning/Training Phase

```
def learn(parallel_data):  
    # do something  
    return parameters
```

$$L : (\Sigma_f^* \times \Sigma_e^*)^* \rightarrow \Theta$$

- Inference Phase

```
def translate(French, parameters):  
    # do something  
    return English
```

$$T : \Sigma_f^* \times \Theta \rightarrow \Sigma_e^*$$

- How?

Using probability: $T(f, \theta) = \arg \max_{e \in \Sigma_e^*} p_\theta(e|f)$

Why probability?

Why probability?



Al-Khalil ibn Ahmad al-Farahidi, 718-786
Image: Wikipedia

Why probability?

- Formalizes...



Al-Khalil ibn Ahmad al-Farahidi, 718-786
Image: Wikipedia

Why probability?

- Formalizes...
- The concept of **models**



Al-Khalil ibn Ahmad al-Farahidi, 718-786
Image: Wikipedia

Why probability?

- Formalizes...
 - The concept of **models**
 - The concept of **data**



Al-Khalil ibn Ahmad al-Farahidi, 718-786
Image: Wikipedia

Why probability?

- Formalizes...
- The concept of **models**
- The concept of **data**
- The concept of **learning**



Al-Khalil ibn Ahmad al-Farahidi, 718-786
Image: Wikipedia

Why probability?

- Formalizes...
- The concept of **models**
- The concept of **data**
- The concept of **learning**
- The concept of **inference** (prediction)



Al-Khalil ibn Ahmad al-Farahidi, 718-786
Image: Wikipedia

Why probability?

- Formalizes...
- The concept of **models**
- The concept of **data**
- The concept of **learning**
- The concept of **inference** (prediction)
- Enables to model **ambiguity**



Al-Khalil ibn Ahmad al-Farahidi, 718-786
Image: Wikipedia

Translation as a probabilistic problem

Translation as a probabilistic problem

- We would like to **model** the **probability** of a **translation** given a **source sentence**

Translation as a probabilistic problem

- We would like to **model** the **probability** of a **translation** given a **source sentence**

$$p(\text{English}|\text{Chinese}) =$$

Translation as a probabilistic problem

- We would like to **model** the **probability** of a **translation** given a **source sentence**
- We can use **Bayes Rule**

$$p(\text{English}|\text{Chinese}) =$$

Translation as a probabilistic problem

- We would like to **model** the **probability** of a **translation** given a **source sentence**
- We can use **Bayes Rule**

$$p(\text{English}|\text{Chinese}) = \frac{p(\text{English}) \times p(\text{Chinese}|\text{English})}{p(\text{Chinese})}$$

Translation as a probabilistic problem

- We would like to **model** the **probability** of a **translation** given a **source sentence**
- We can use **Bayes Rule**
 - Why would we want that?

$$p(\text{English}|\text{Chinese}) = \frac{p(\text{English}) \times p(\text{Chinese}|\text{English})}{p(\text{Chinese})}$$

Translation as a probabilistic problem

- We would like to **model** the probability of a **translation** given a **source sentence**
 - We can use **Bayes Rule**
 - Why would we want that?

$$\frac{p(\text{English}) \times p(\text{Chinese}|\text{English})}{p(\text{Chinese})}$$

language model

translation model

How do we define $p(\text{Chinese} \mid \text{English})$?

How do we define $p(\text{Chinese} \mid \text{English})$?

- IBM Models (Brown, Dellapietra, Dellapietra, and Mercer, 93')

How do we define $p(\text{Chinese} \mid \text{English})$?

- IBM Models (Brown, Dellapietra, Dellapietra, and Mercer, 93')
- “We define a concept of word-by-word **alignment**”

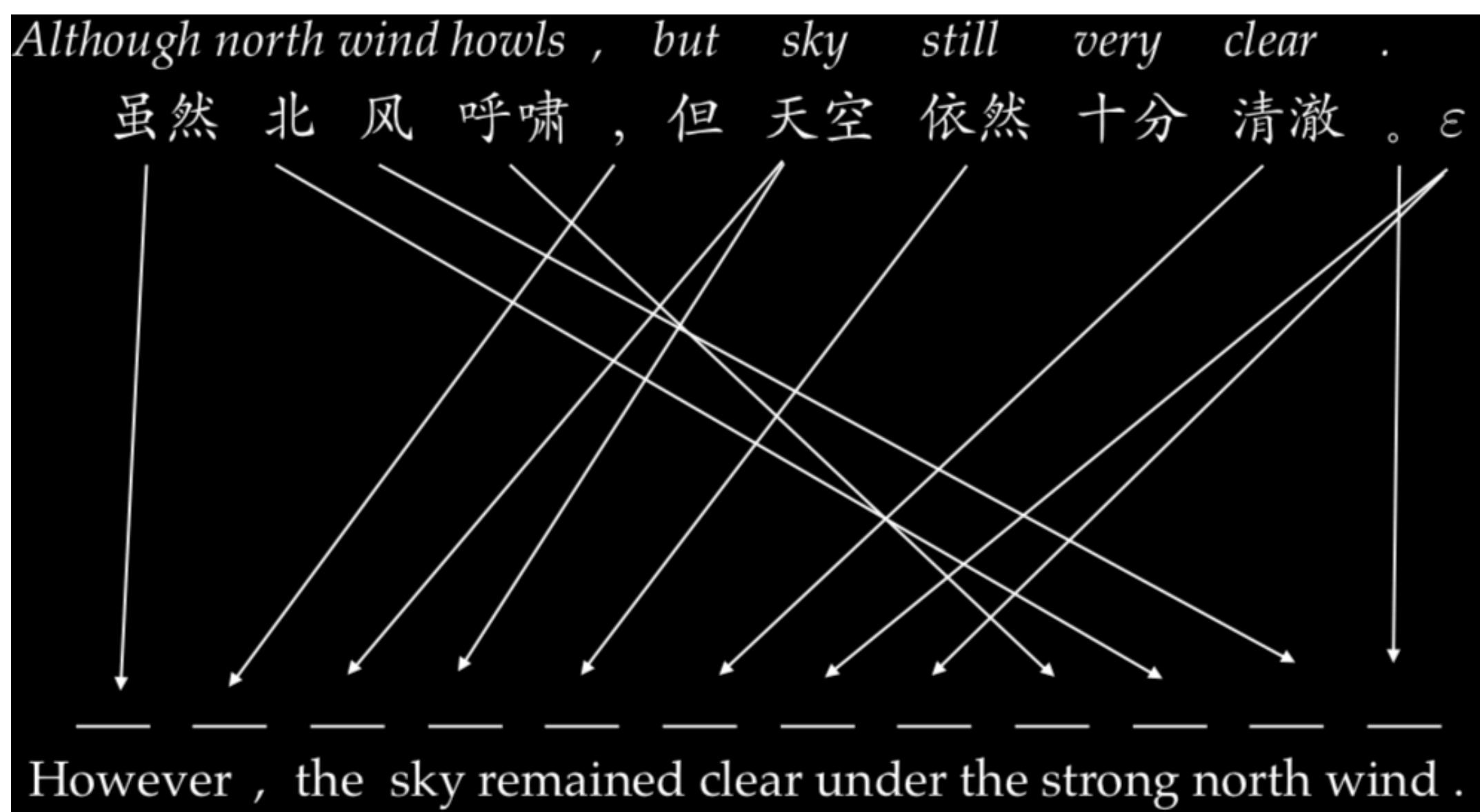
How do we define $p(\text{Chinese} \mid \text{English})$?

- IBM Models (Brown, Dellapietra, Dellapietra, and Mercer, 93')
- “We define a concept of word-by-word **alignment**”



How do we define $p(\text{Chinese} \mid \text{English})$?

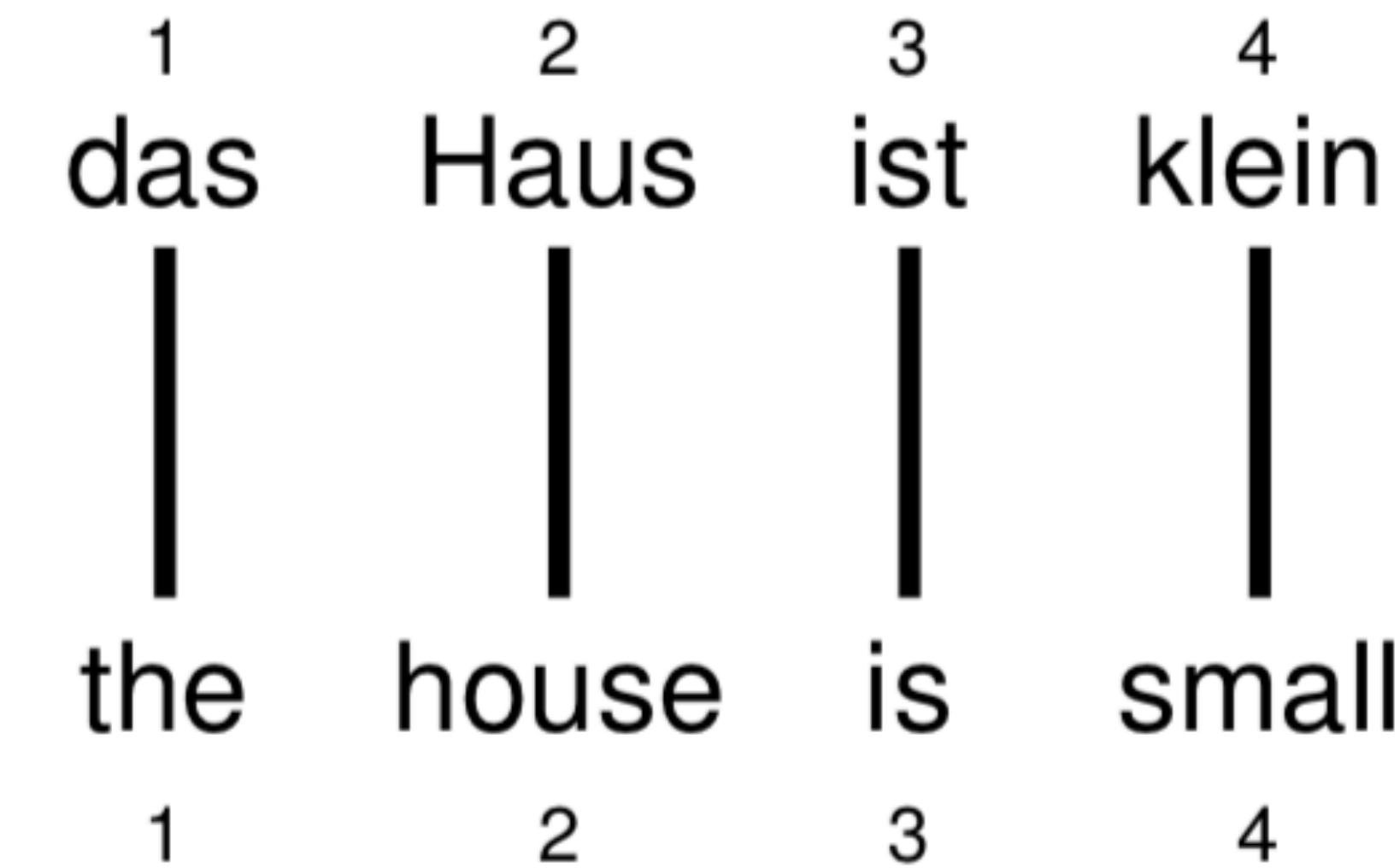
- IBM Models (Brown, Dellapietra, Dellapietra, and Mercer, 93')
- “We define a concept of word-by-word **alignment**”



Understanding Alignments

Understanding Alignments

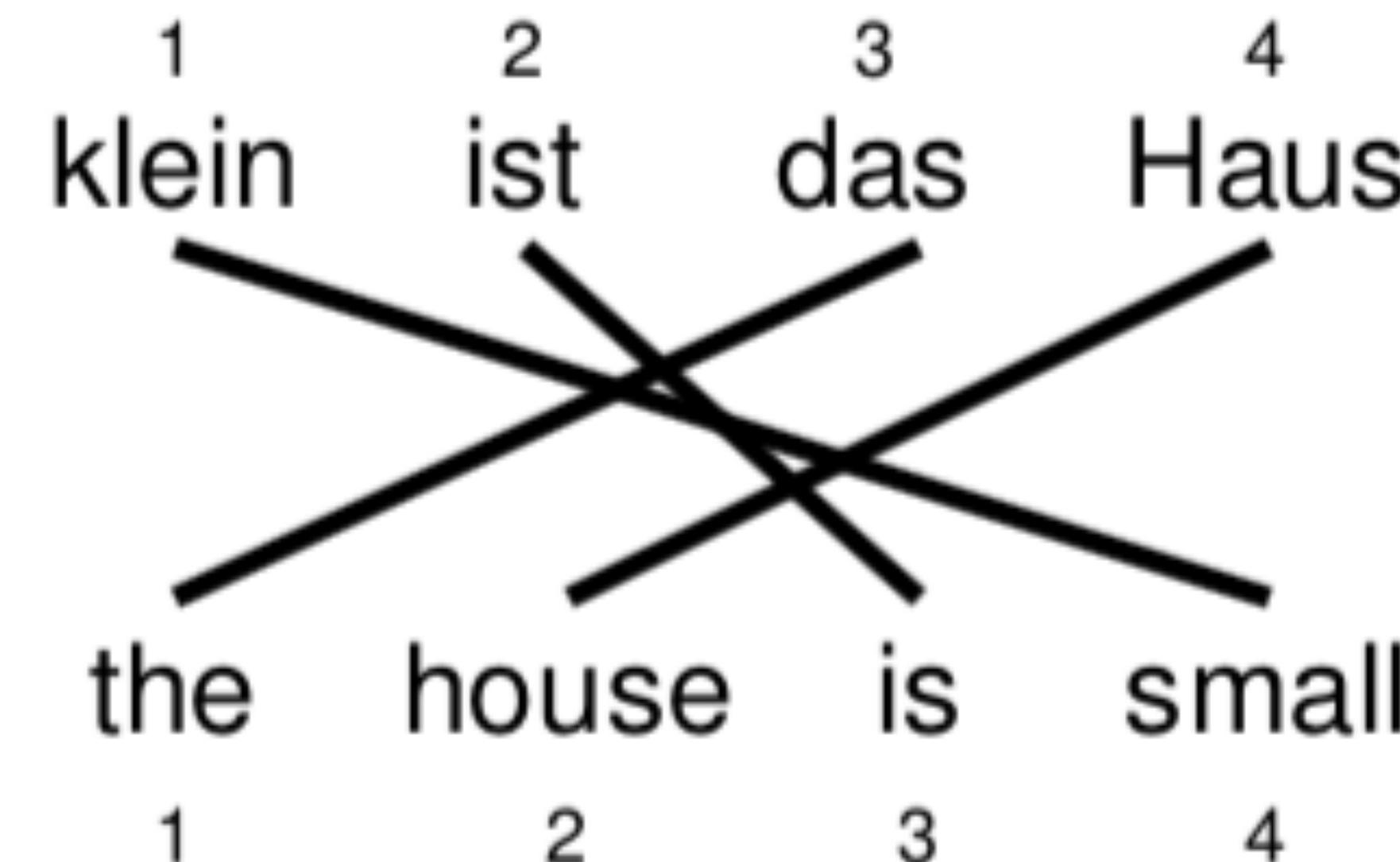
- Alignment function



$$a : \{1 \rightarrow 1, 2 \rightarrow 2, 3 \rightarrow 3, 4 \rightarrow 4\}$$

Understanding Alignments

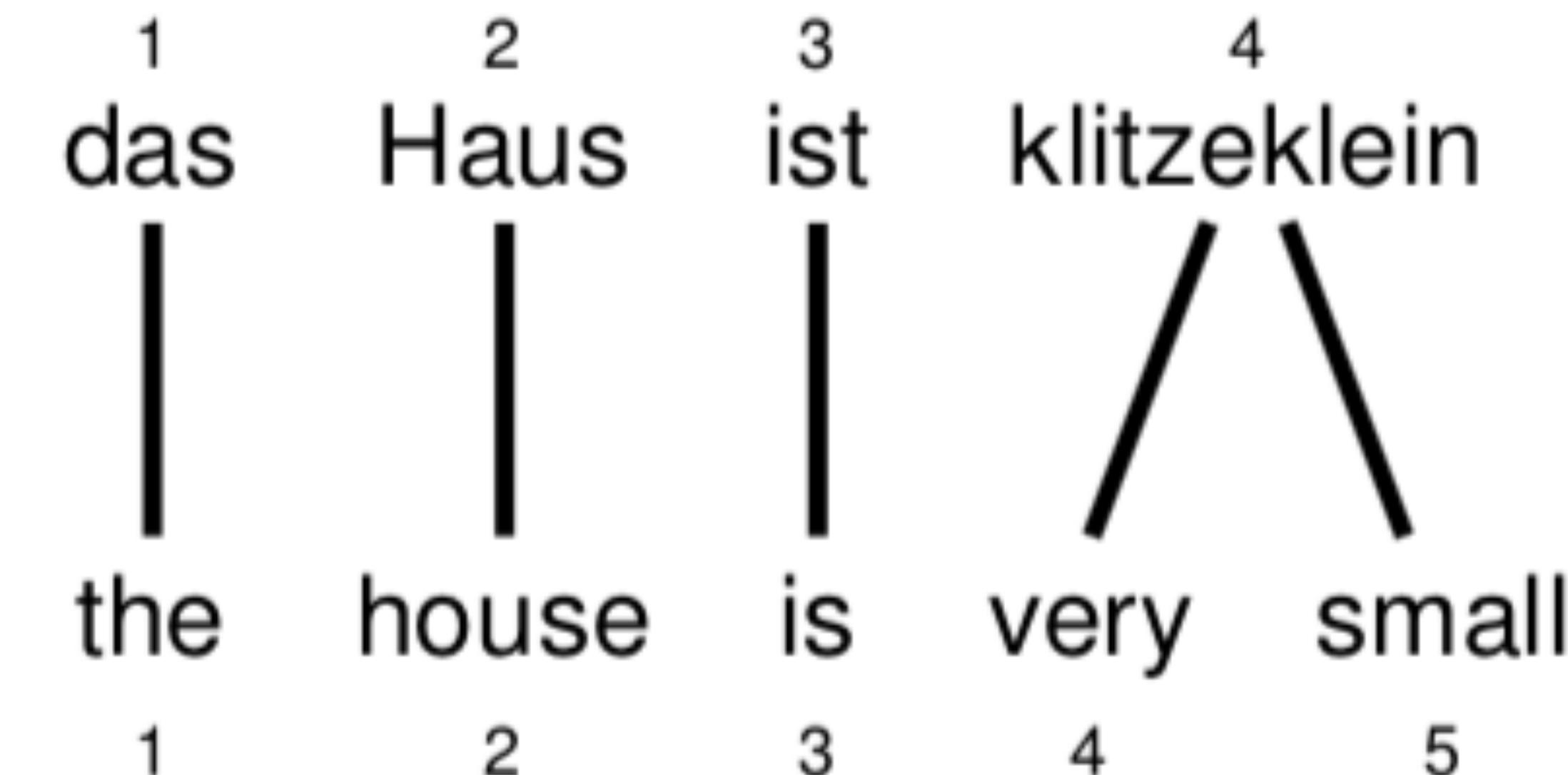
- Alignment function
- Reordering



$$a : \{1 \rightarrow 3, 2 \rightarrow 4, 3 \rightarrow 2, 4 \rightarrow 1\}$$

Understanding Alignments

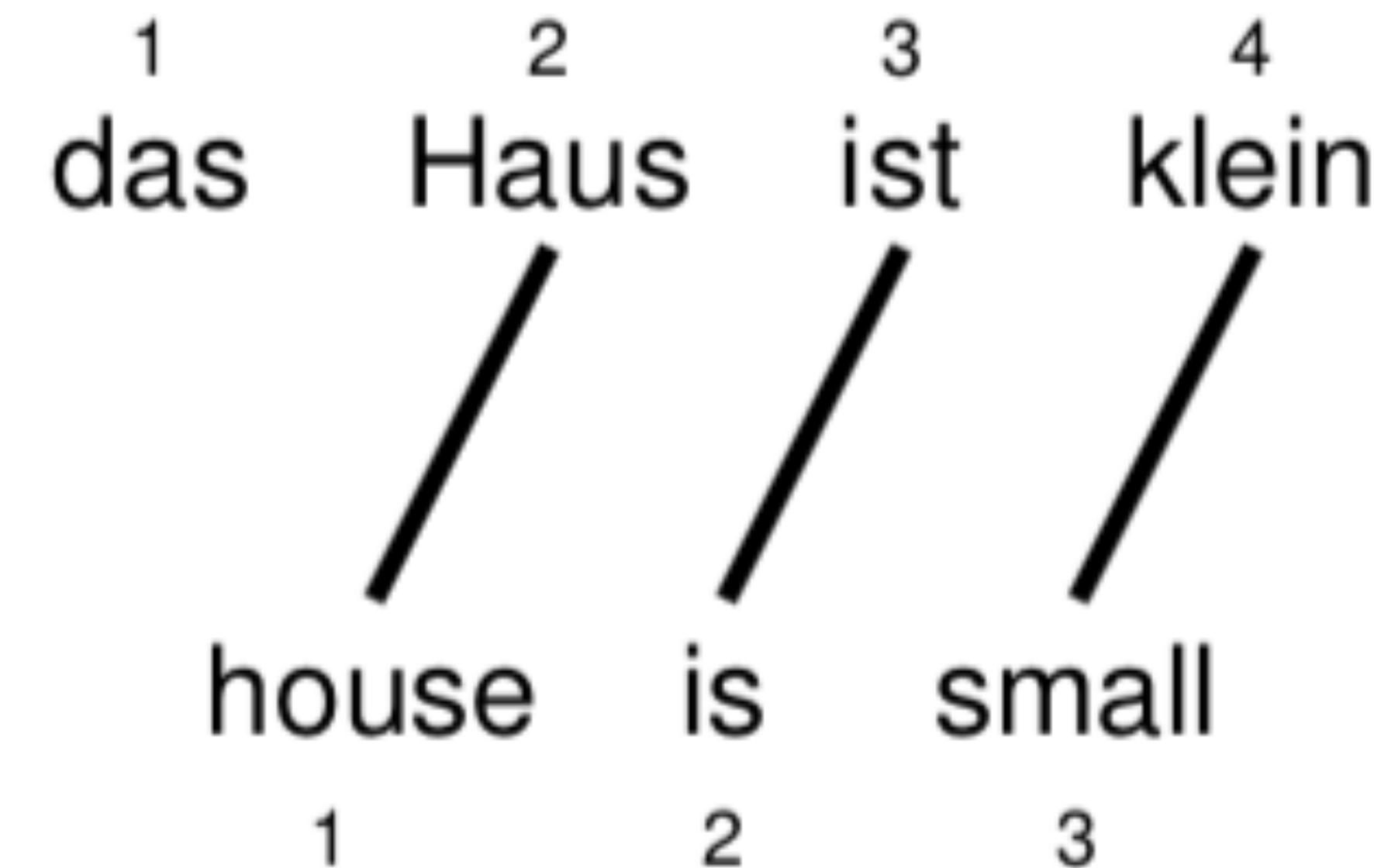
- Alignment function
- Reordering
- One-to-Many



$$a : \{1 \rightarrow 1, 2 \rightarrow 2, 3 \rightarrow 3, 4 \rightarrow 4, 5 \rightarrow 4\}$$

Understanding Alignments

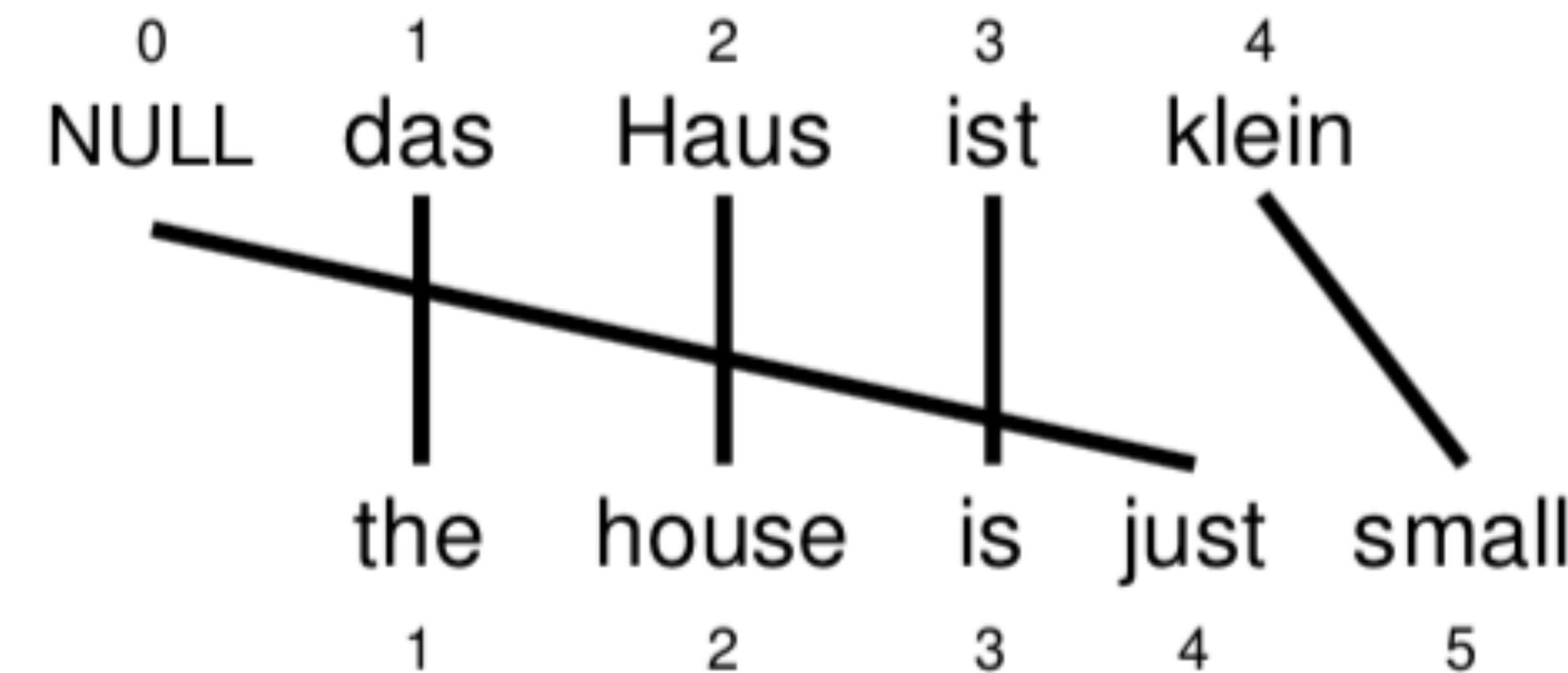
- Alignment function
- Reordering
- One-to-Many
- Dropping words



$$a : \{1 \rightarrow 2, 2 \rightarrow 3, 3 \rightarrow 4\}$$

Understanding Alignments

- Alignment function
- Reordering
- One-to-Many
- Dropping words
- Inserting words



$$a : \{1 \rightarrow 1, 2 \rightarrow 2, 3 \rightarrow 3, 4 \rightarrow 0, 5 \rightarrow 4\}$$

IBM Model1's Generative Story

IBM Model1's Generative Story

- Given a source sentence, how was the target sentence generated?

IBM Model1's Generative Story

- Given a source sentence, how was the target sentence generated?



IBM Model1's Generative Story

- Given a source sentence, how was the target sentence generated?

Although north wind howls , but sky still very clear .
虽然 北 风 呼啸 , 但 天空 依然 十分 清澈 。ε

IBM Model1's Generative Story

- Given a source sentence, how was the target sentence generated?
 - Sample a length for the target sentence

Although north wind howls , but sky still very clear .
虽然 北 风 呼啸 , 但 天 空 依 然 十 分 清 澈 。 ε

$$p(\text{English length} | \text{Chinese length})$$

IBM Model1's Generative Story

- Given a source sentence, how was the target sentence generated?
 - Sample a length for the target sentence
 - For each target position:

Although north wind howls , but sky still very clear .
虽然 北 风 呼 啸 , 但 天 空 依 然 十 分 清 澈 。 ε

$$p(\text{English length} | \text{Chinese length})$$

IBM Model1's Generative Story

- Given a source sentence, how was the target sentence generated?
 - Sample a length for the target sentence
 - For each target position:
 - Sample an alignment (to a position in the source)

Although north wind howls , but sky still very clear .
虽然 北 风 呼啸 , 但 天 空 依 然 十 分 清 澈 。 ε

p(Chinese word position)

IBM Model1's Generative Story

- Given a source sentence, how was the target sentence generated?
 - Sample a length for the target sentence
 - For each target position:
 - Sample an alignment (to a position in the source)
 - Sample a word translation given this alignment

Although north wind howls , but sky still very clear .
虽然 北 风 呼 啸 , 但 天 空 依 然 十 分 清 澈 。 ε

虽然北风呼啸，但天空依然十分清澈。 ε

THE JOURNAL OF CLIMATE

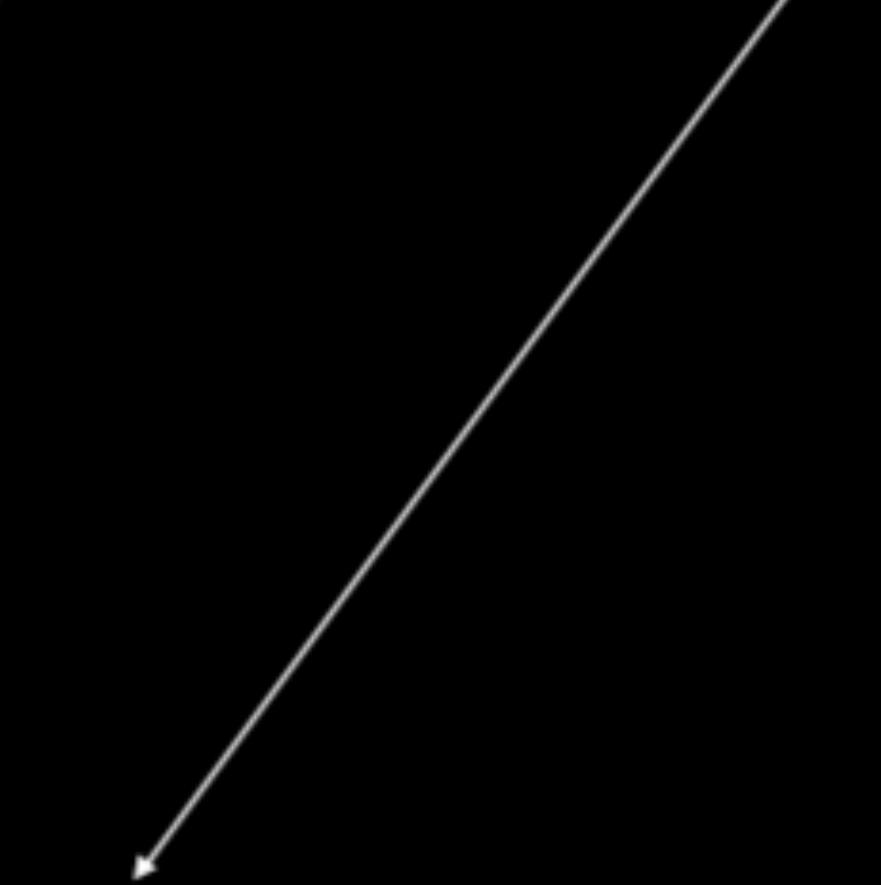
However

$$p(\text{English word} | \text{Chinese word})$$

IBM Model1's Generative Story

- Given a source sentence, how was the target sentence generated?
 - Sample a length for the target sentence
 - For each target position:
 - Sample an alignment (to a position in the source)
 - Sample a word translation given this alignment
 - Repeat until done

Although north wind howls , but sky still very clear .
虽然 北 风 呼啸 , 但 天空 依然 十分 清澈 。 ε

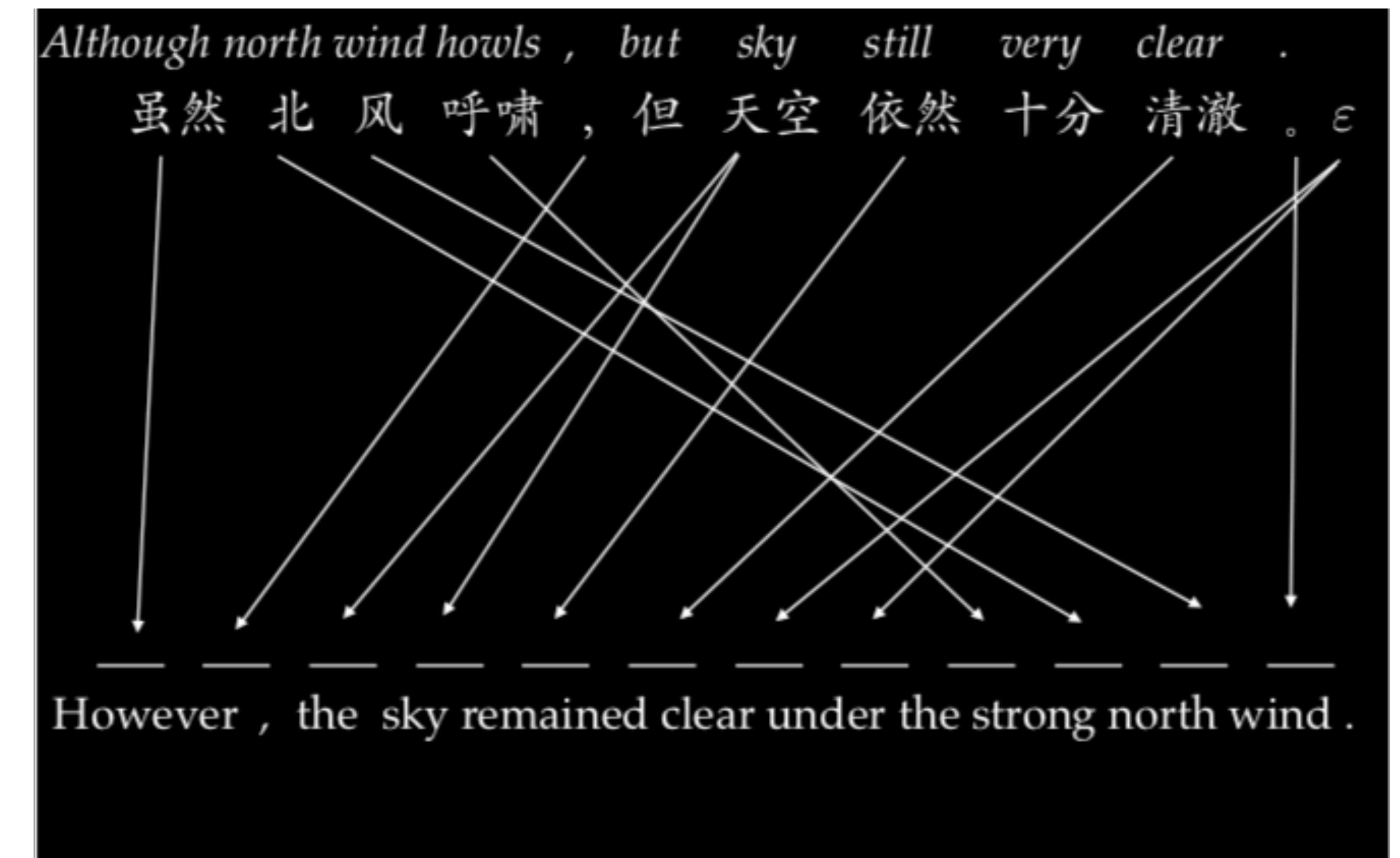


— — — — — — — — — — — — — —
However ,

$p(\text{English word} | \text{Chinese word})$

IBM Model 1's Generative Story

- Given a source sentence, how was the target sentence generated?
- Sample a length for the target sentence
- For each target position:
 - Sample an alignment (to a position in the source)
 - Sample a word translation given this alignment
- Repeat until done



IBM Model 1

$$p(\mathbf{f}, \mathbf{a}|\mathbf{e}) = p(I|J) \prod_{i=1}^I p(a_i|J) \cdot p(f_i|e_{a_i})$$

IBM Model 1

$$p(\mathbf{f}, \mathbf{a}|\mathbf{e}) = p(I|J) \prod_{i=1}^I p(a_i|J) \cdot p(f_i|e_{a_i})$$

**sample
sentence
length**

IBM Model 1

$$p(\mathbf{f}, \mathbf{a}|\mathbf{e}) = p(I|J) \prod_{i=1}^I p(a_i|J) \cdot p(f_i|e_{a_i})$$

sample
alignment

sample
sentence
length

The diagram shows the formula for IBM Model 1. It consists of a central equation: $p(\mathbf{f}, \mathbf{a}|\mathbf{e}) = p(I|J) \prod_{i=1}^I p(a_i|J) \cdot p(f_i|e_{a_i})$. Above the equation, the words "sample alignment" are written in bold black font. Below the equation, the words "sample sentence length" are written in bold black font. Three vertical arrows point upwards from the bottom of the page towards the corresponding parts of the formula: one arrow points to the summation index i , another points to the term $p(f_i|e_{a_i})$, and a third points to the entire product term $\prod_{i=1}^I p(a_i|J)$.

IBM Model 1

$$p(\mathbf{f}, \mathbf{a}|\mathbf{e}) = p(I|J) \prod_{i=1}^I p(a_i|J) \cdot p(f_i|e_{a_i})$$

sample alignment

sample sentence length

sample word translation

The diagram shows the probability formula for IBM Model 1: $p(\mathbf{f}, \mathbf{a}|\mathbf{e}) = p(I|J) \prod_{i=1}^I p(a_i|J) \cdot p(f_i|e_{a_i})$. Above the formula, the text "sample alignment" is centered. Below the formula, three vertical arrows point upwards from the text labels to their corresponding parts in the formula: the first arrow points to the summation index $i=1$, the second to the upper limit I , and the third to the term $p(f_i|e_{a_i})$.

IBM Model 1

- f - foreign sentence (Chinese)

$$p(\mathbf{f}, \mathbf{a}|\mathbf{e}) = p(I|J) \prod_{i=1}^I p(a_i|J) \cdot p(f_i|e_{a_i})$$

sample alignment

sample sentence length

sample word translation

The diagram shows the probability formula for IBM Model 1: $p(\mathbf{f}, \mathbf{a}|\mathbf{e}) = p(I|J) \prod_{i=1}^I p(a_i|J) \cdot p(f_i|e_{a_i})$. The formula is contained within a large black rectangular box. Above the box, the text "sample alignment" is centered. Below the box, three labels are positioned with arrows pointing upwards to specific parts of the formula: "sample sentence length" points to the index $i=1$, "sample word translation" points to the term $p(f_i|e_{a_i})$, and "sample alignment" points to the product symbol \prod .

IBM Model 1

- f - foreign sentence (Chinese)
- a - alignment

$$p(\mathbf{f}, \mathbf{a}|\mathbf{e}) = p(I|J) \prod_{i=1}^I p(a_i|J) \cdot p(f_i|e_{a_i})$$

sample
alignment

sample
sentence
length

sample
word
translation

IBM Model 1

- f - foreign sentence (Chinese)
- a - alignment
- e - English sentence

$$p(\mathbf{f}, \mathbf{a}|\mathbf{e}) = p(I|J) \prod_{i=1}^I p(a_i|J) \cdot p(f_i|e_{a_i})$$

sample alignment

sample sentence length

sample word translation

IBM Model 1

- f - foreign sentence (Chinese)
- a - alignment
- e - English sentence
- l - foreign sentence length

$$p(\mathbf{f}, \mathbf{a}|\mathbf{e}) = p(I|J) \prod_{i=1}^I p(a_i|J) \cdot p(f_i|e_{a_i})$$

sample
alignment

sample
sentence
length

sample
word
translation

IBM Model 1

- f - foreign sentence (Chinese)
- a - alignment
- e - English sentence
- I - foreign sentence length
- J - English sentence length

$$p(\mathbf{f}, \mathbf{a}|\mathbf{e}) = p(I|J) \prod_{i=1}^I p(a_i|J) \cdot p(f_i|e_{a_i})$$

sample
alignment

sample
sentence
length

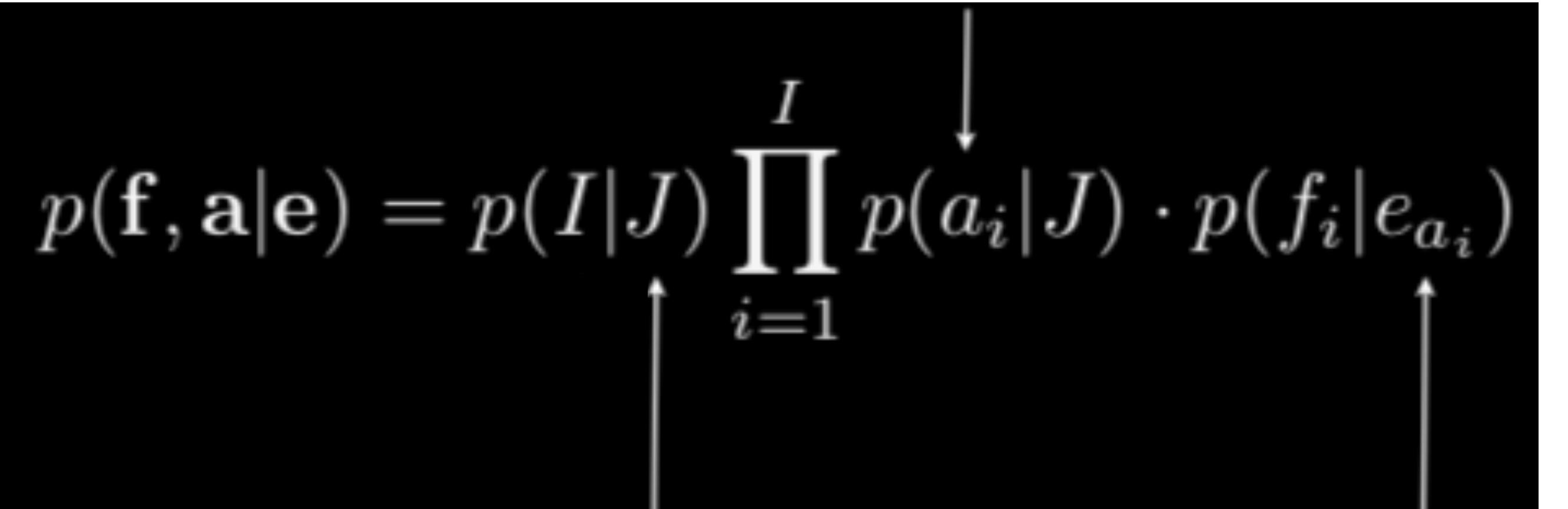
sample
word
translation

IBM Model 1

- f - foreign sentence (Chinese)
- a - alignment
- e - English sentence
- I - foreign sentence length
- J - English sentence length
- a_i - alignment of i-th foreign word

$$p(\mathbf{f}, \mathbf{a}|\mathbf{e}) = p(I|J) \prod_{i=1}^I p(a_i|J) \cdot p(f_i|e_{a_i})$$

**sample
alignment**
**sample
sentence
length**
**sample
word
translation**



IBM Model 1

- f - foreign sentence (Chinese)
- a - alignment
- e - English sentence
- I - foreign sentence length
- J - English sentence length
- a_i - alignment of i -th foreign word
- f_i - foreign word in position i

$$p(\mathbf{f}, \mathbf{a}|\mathbf{e}) = p(I|J) \prod_{i=1}^I p(a_i|J) \cdot p(f_i|e_{a_i})$$

sample alignment

sample sentence length

sample word translation

IBM Model 1

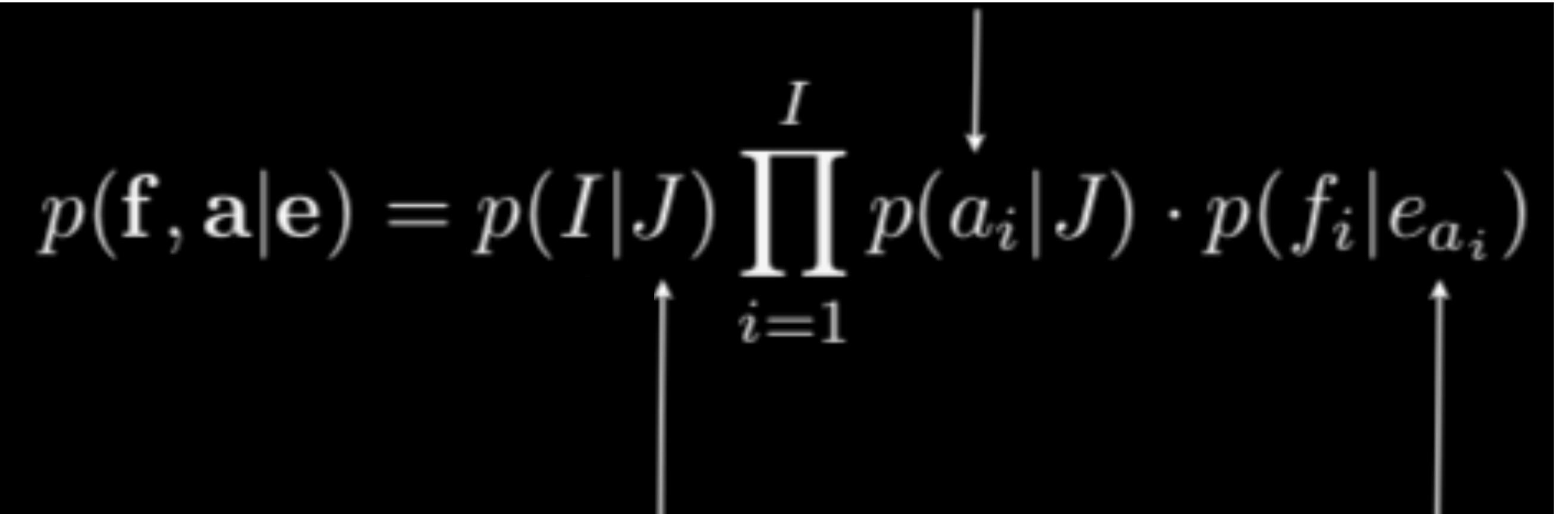
- f - foreign sentence (Chinese)
- a - alignment
- e - English sentence
- I - foreign sentence length
- J - English sentence length
- a_i - alignment of i -th foreign word
- f_i - foreign word in position i
- $e_{\{a_i\}}$ - English word in position a_i

$$p(f, a|e) = p(I|J) \prod_{i=1}^I p(a_i|J) \cdot p(f_i|e_{a_i})$$

**sample
alignment**

**sample
sentence
length**

**sample
word
translation**

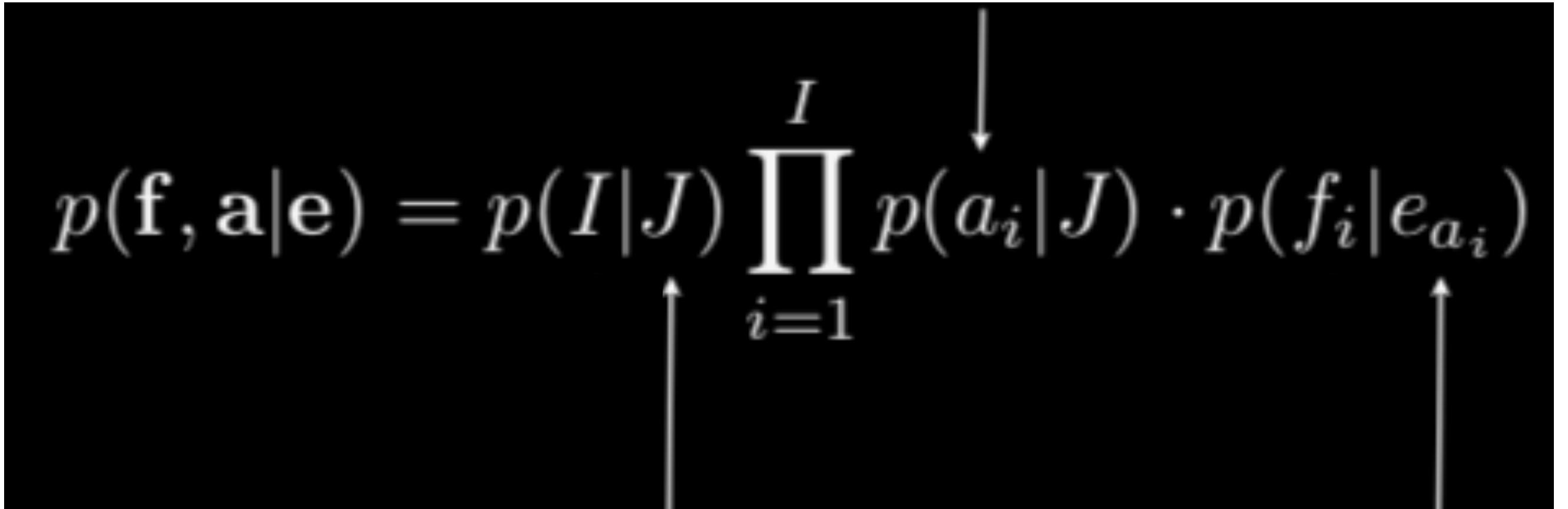


How can we learn this from data?

$$p(\mathbf{f}, \mathbf{a}|\mathbf{e}) = p(I|J) \prod_{i=1}^I p(a_i|J) \cdot p(f_i|e_{a_i})$$

How can we learn this from data?

- What are the parameters we should learn?

$$p(\mathbf{f}, \mathbf{a}|\mathbf{e}) = p(I|J) \prod_{i=1}^I p(a_i|J) \cdot p(f_i|e_{a_i})$$
A diagram consisting of a black rectangle containing a mathematical equation. The equation is $p(\mathbf{f}, \mathbf{a}|\mathbf{e}) = p(I|J) \prod_{i=1}^I p(a_i|J) \cdot p(f_i|e_{a_i})$. Above the equation, there are five white arrows pointing downwards towards the corresponding terms: one arrow points to the variable I in the first term, another to J in the first term, two arrows point to a_i in the second term, and one arrow points to e_{a_i} in the third term.

How can we learn this from data?

- What are the parameters we should learn?
- Sentence length distributions

$$p(\mathbf{f}, \mathbf{a}|\mathbf{e}) = p(I|J) \prod_{i=1}^I p(a_i|J) \cdot p(f_i|e_{a_i})$$

**Sentence length
distributions**

How can we learn this from data?

- What are the parameters we should learn?
- Sentence length distributions
- Alignment distributions

$$p(\mathbf{f}, \mathbf{a}|\mathbf{e}) = p(I|J) \prod_{i=1}^I p(a_i|J) \cdot p(f_i|e_{a_i})$$

Alignment distributions

Sentence length distributions

The diagram illustrates the components of a statistical machine translation model. A large black box contains the formula $p(\mathbf{f}, \mathbf{a}|\mathbf{e}) = p(I|J) \prod_{i=1}^I p(a_i|J) \cdot p(f_i|e_{a_i})$. Three arrows point upwards from below the box to specific parts of the formula: one to the term $p(I|J)$ labeled "Sentence length distributions", one to the term $p(a_i|J)$ labeled "Alignment distributions", and one to the term $p(f_i|e_{a_i})$ also labeled "Alignment distributions".

How can we learn this from data?

- What are the parameters we should learn?
- Sentence length distributions
- Alignment distributions
- Word translation distributions

$$p(\mathbf{f}, \mathbf{a}|\mathbf{e}) = p(I|J) \prod_{i=1}^I p(a_i|J) \cdot p(f_i|e_{a_i})$$

Alignment distributions

Sentence length distributions

Word translation distributions

```
graph TD; A[Alignment distributions] --> B[p(I|J)]; C["Sentence length distributions"] --> D["product i=1 to I p(ai|J)"]; E["Word translation distributions"] --> F["p(fi|ei)"];
```

How can we learn this from data?

How can we learn this from data?

- If we **know** the **alignments**, easy:

How can we learn this from data?

- If we **know** the **alignments**, easy:
- $p(I|J)$ - learn by counting

$$p(I|J) = \frac{\text{\# Aligned Chinese sentences of length I}}{\text{\# English sentences of length J}}$$

How can we learn this from data?

- If we **know** the **alignments**, easy:

- $p(l|J)$ - learn by counting
- Alignment distributions - use uniform distribution

$$p(l|J) = \frac{\text{\# Aligned Chinese sentences of length } l}{\text{\# English sentences of length } J}$$

$$p(a_i|J) = \frac{1}{J+1}$$

How can we learn this from data?

- If we **know** the **alignments**, easy:

- $p(l|J)$ - learn by counting
- Alignment distributions - use uniform distribution
- Word translation distributions - again by counting

$$p(l|J) = \frac{\text{\# Aligned Chinese sentences of length } l}{\text{\# English sentences of length } J}$$

$$p(a_i|J) = \frac{1}{J+1}$$

$$p(\text{however} | \text{然而}) = \frac{\text{\# times “然而” aligned to “however”}}{\text{\# times “然而” aligned to any word}}$$

How can we learn this from data?

- If we **know** the **alignments**, easy:

- $p(l|J)$ - learn by counting

- Alignment distributions - use uniform distribution

- Word translation distributions - again by counting

- But do we know the alignments?

$$p(l|J) = \frac{\text{\# Aligned Chinese sentences of length } l}{\text{\# English sentences of length } J}$$

$$p(a_i|J) = \frac{1}{J+1}$$

$$p(\text{however} | \text{然而}) = \frac{\text{\# times “然而” aligned to “however”}}{\text{\# times “然而” aligned to any word}}$$

Alignments as Latent Variables

Alignments as Latent Variables

- Parameters (translation probabilities) and alignments are both unknown!

Alignments as Latent Variables

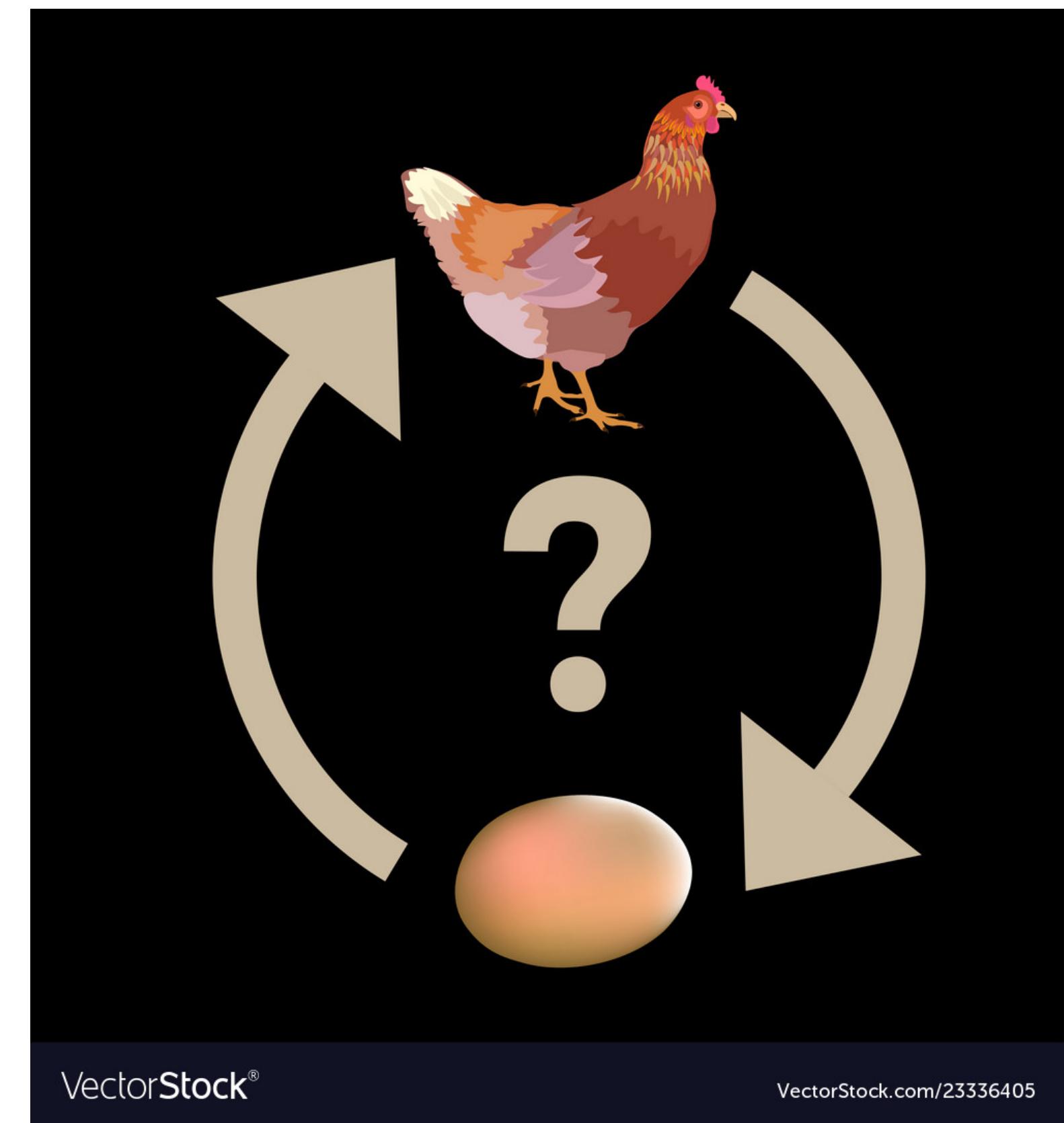
- Parameters (translation probabilities) and alignments are both unknown!
- If we knew the alignments in the training data, we could calculate the parameters by counting (prev. slide)

Alignments as Latent Variables

- Parameters (translation probabilities) and alignments are both unknown!
- If we knew the alignments in the training data, we could calculate the parameters by counting (prev. slide)
- If we knew the parameters, we could calculate the *expected* alignment counts

Alignments as Latent Variables

- Parameters (translation probabilities) and alignments are both unknown!
- If we knew the alignments in the training data, we could calculate the parameters by counting (prev. slide)
- If we knew the parameters, we could calculate the *expected* alignment counts



The Expectation-Maximization (EM) Algorithm

The Expectation-Maximization (EM) Algorithm

- Dempster, Laird and Rubin (1977)

The Expectation-Maximization (EM) Algorithm

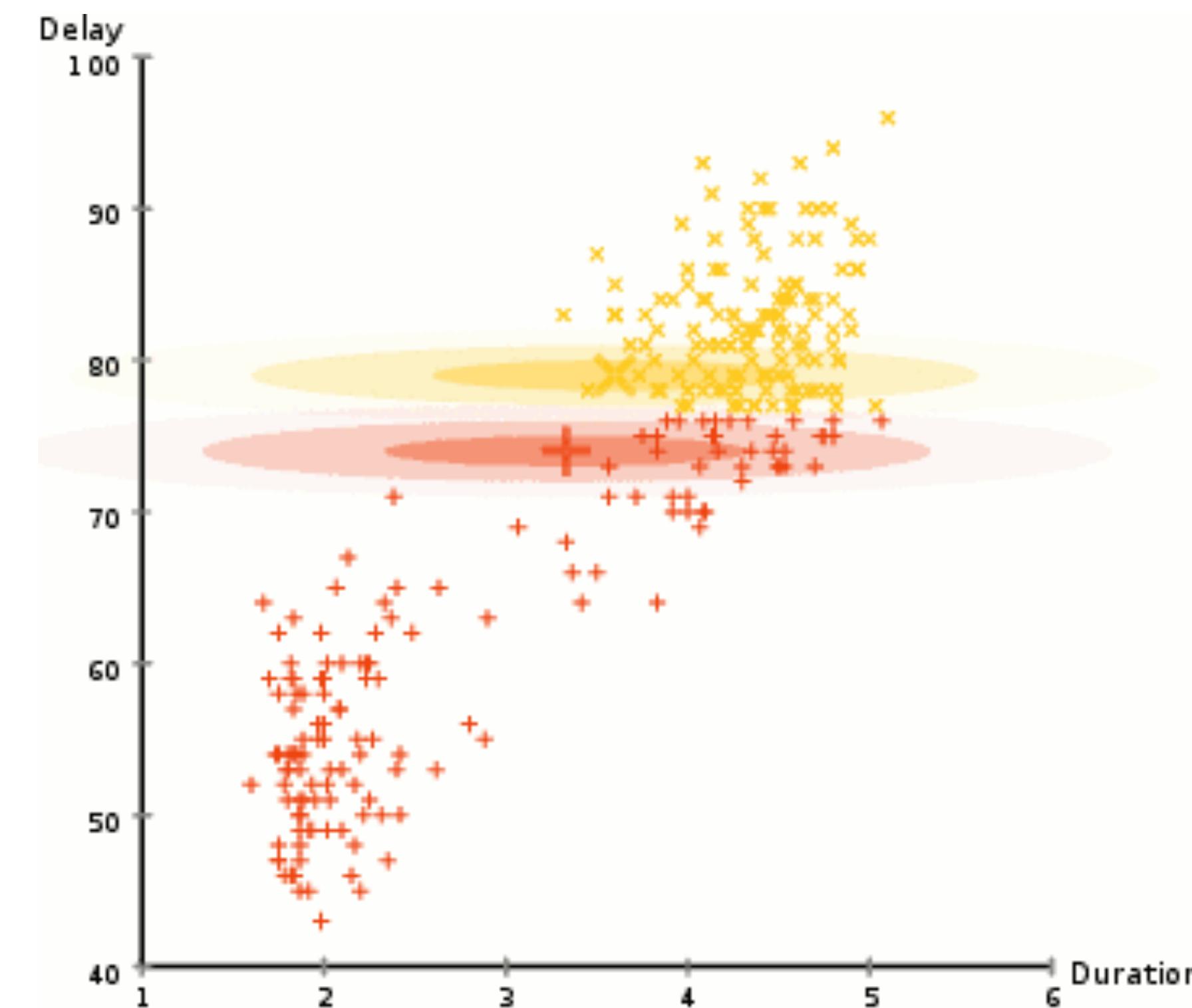
- Dempster, Laird and Rubin (1977)
- One of the most widely used algorithms in machine learning

The Expectation-Maximization (EM) Algorithm

- Dempster, Laird and Rubin (1977)
- One of the most widely used algorithms in machine learning
- Many applications, e.g. clustering

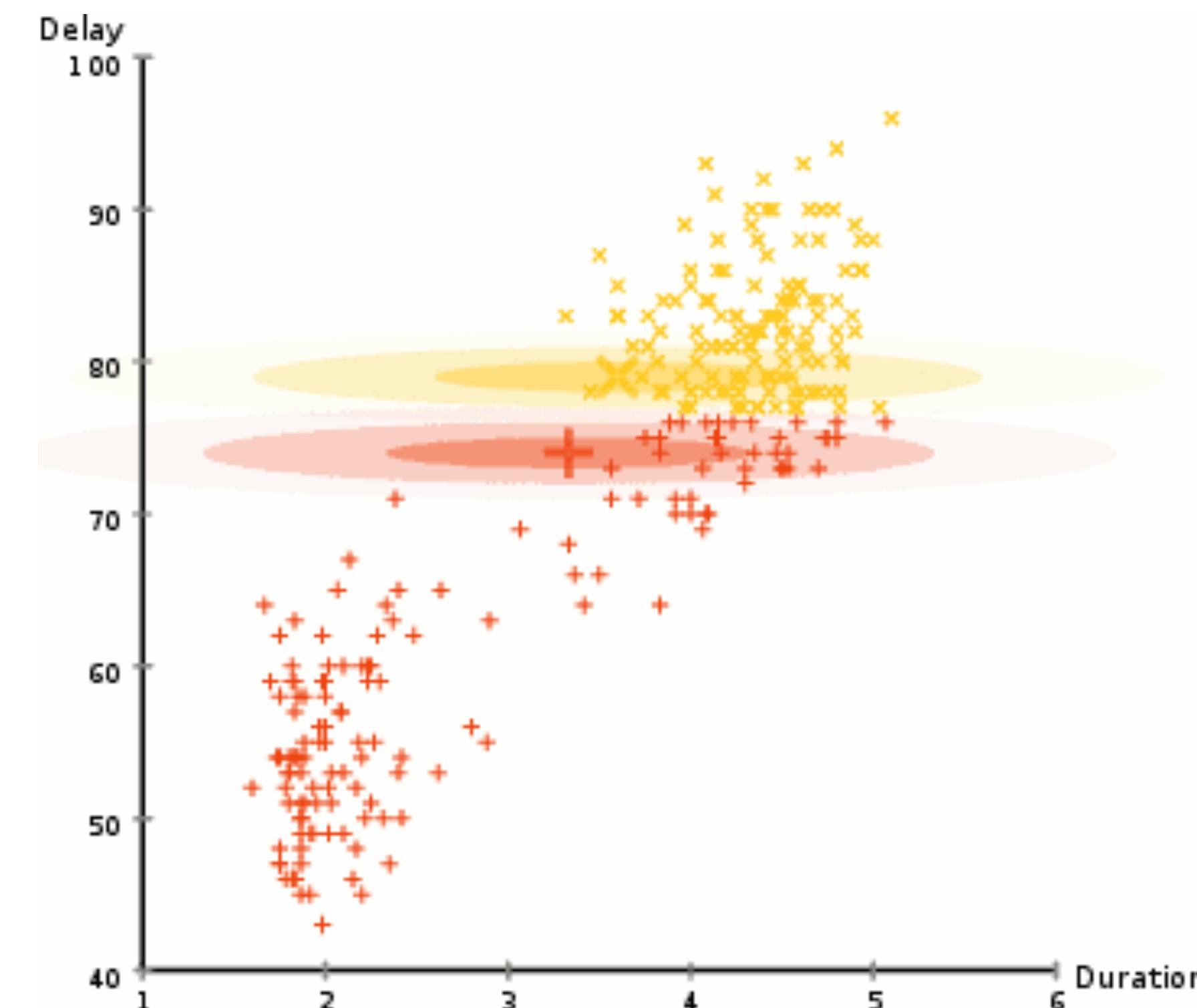
The Expectation-Maximization (EM) Algorithm

- Dempster, Laird and Rubin (1977)
- One of the most widely used algorithms in machine learning
- Many applications, e.g. clustering
- Main idea - start randomly, and iterate until convergence:



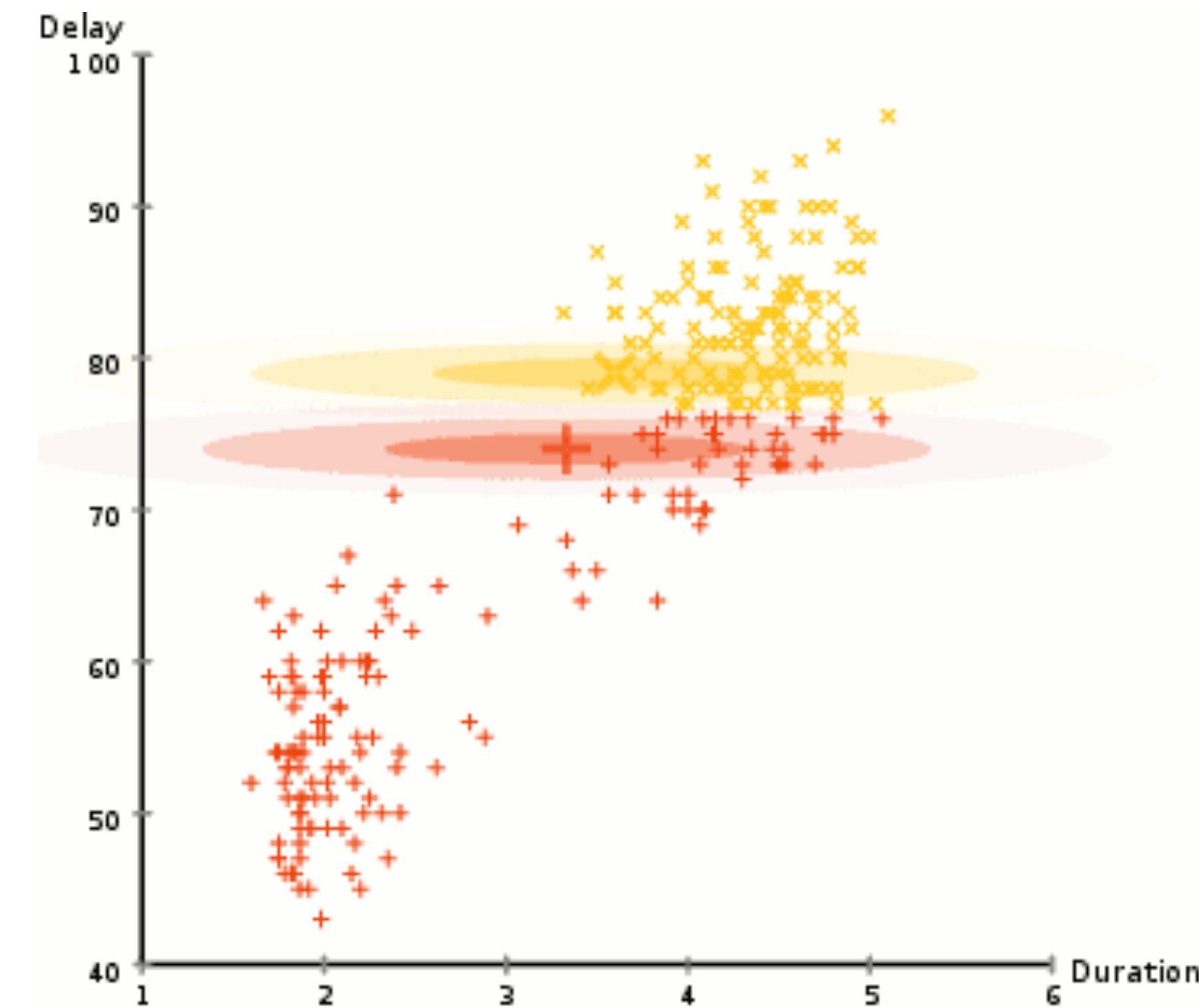
The Expectation-Maximization (EM) Algorithm

- Dempster, Laird and Rubin (1977)
- One of the most widely used algorithms in machine learning
- Many applications, e.g. clustering
- Main idea - start randomly, and iterate until convergence:
 - Calculate expected counts for missing data (expectation, or E-step) using current model



The Expectation-Maximization (EM) Algorithm

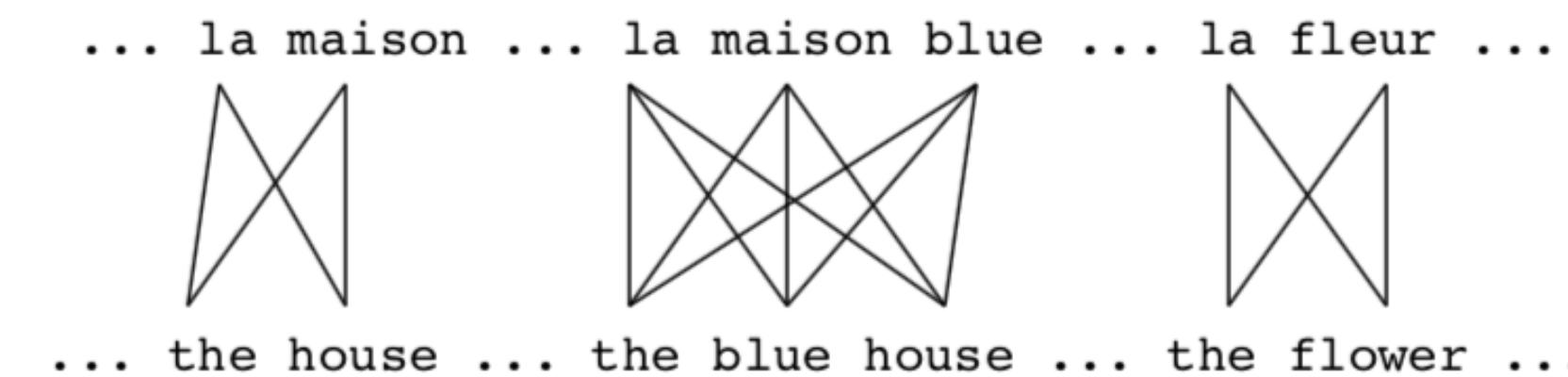
- Dempster, Laird and Rubin (1977)
- One of the most widely used algorithms in machine learning
- Many applications, e.g. clustering
- Main idea - start randomly, and iterate until convergence:
 - Calculate expected counts for missing data (expectation, or E-step) using current model
 - Find new model parameters that maximize the data likelihood (maximization, or M-step)



EM for IBM Model1 - Overview

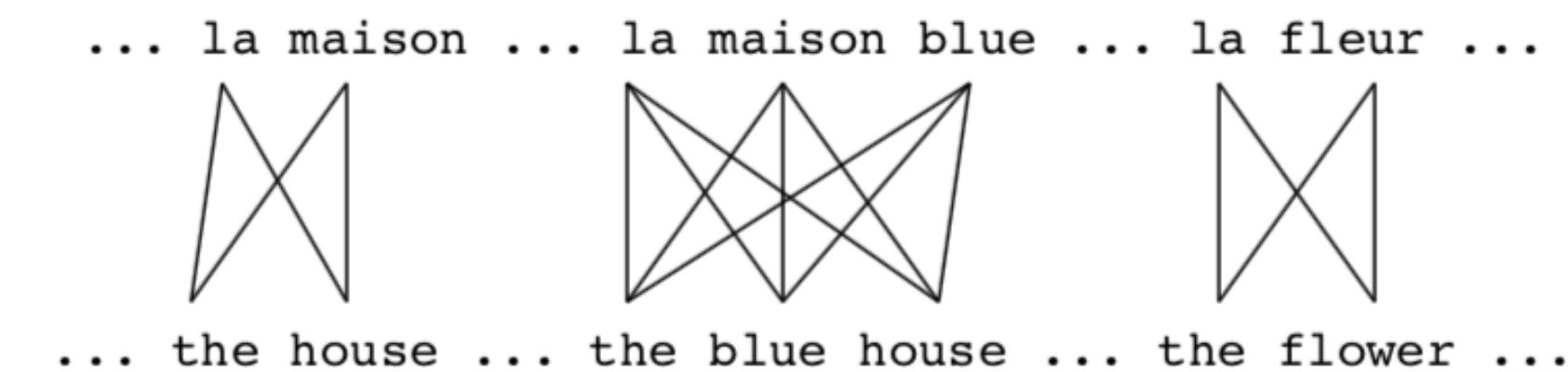
EM for IBM Model 1 - Overview

- Start with all alignments equally likely



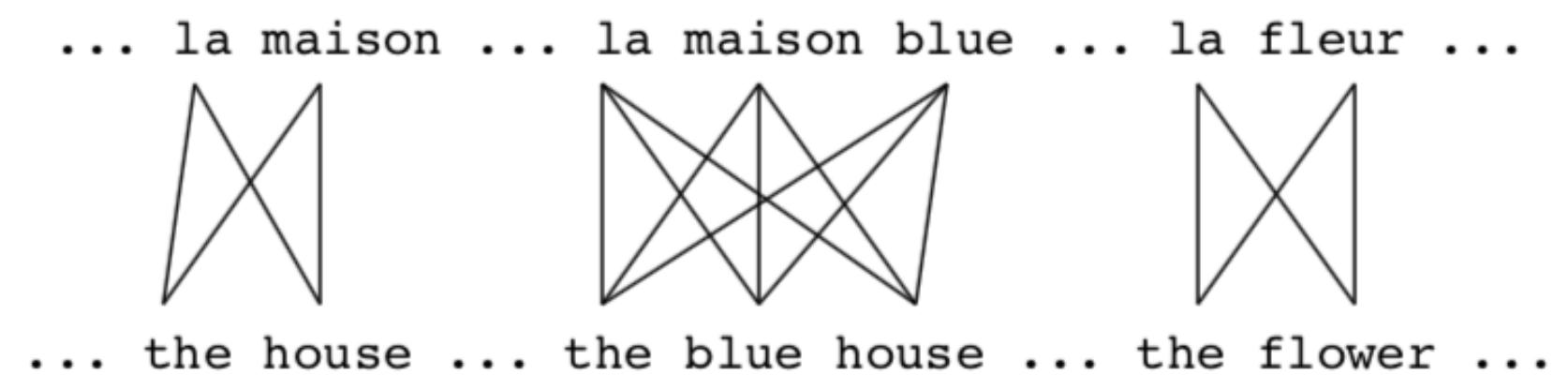
EM for IBM Model 1 - Overview

- Start with all alignments equally likely
- In each iteration:



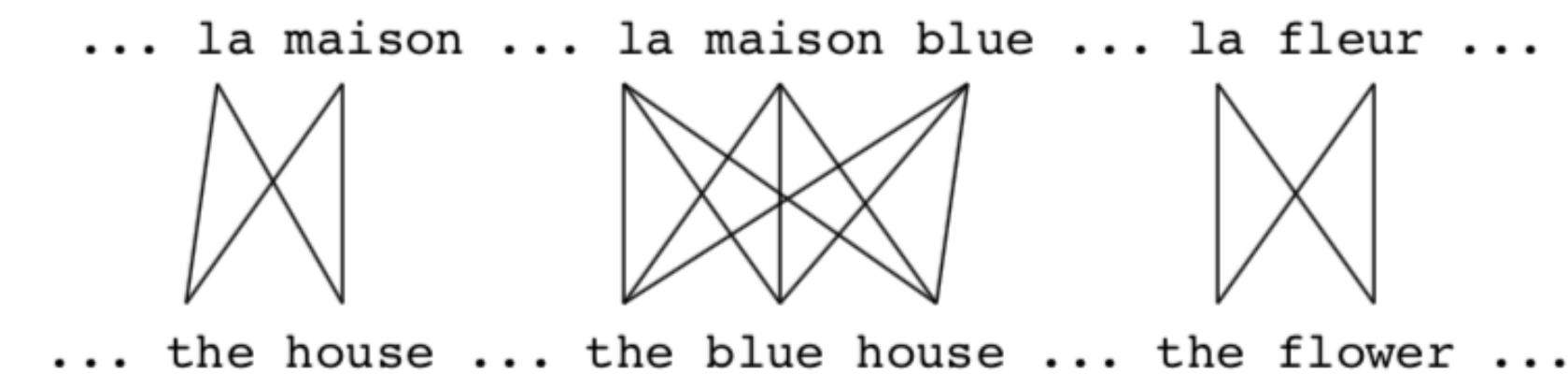
EM for IBM Model 1 - Overview

- Start with all alignments equally likely
- In each iteration:
 - look at the entire dataset and sum the (expected) alignments we saw for each word and its possible translations (E-step)



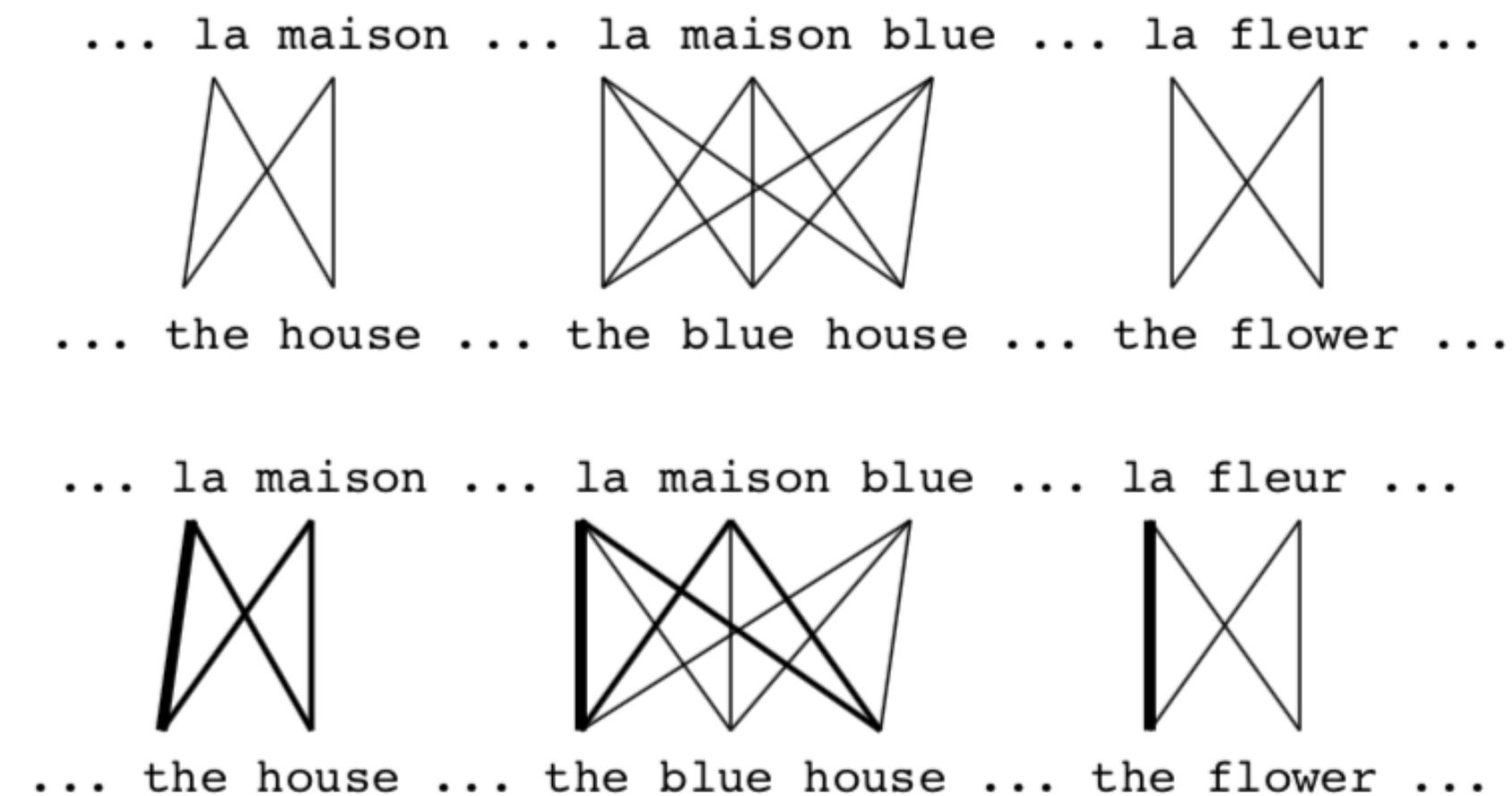
EM for IBM Model 1 - Overview

- Start with all alignments equally likely
- In each iteration:
 - look at the entire dataset and sum the (expected) alignments we saw for each word and its possible translations (E-step)
 - Update the translation probabilities according to those global counts (M-step)...



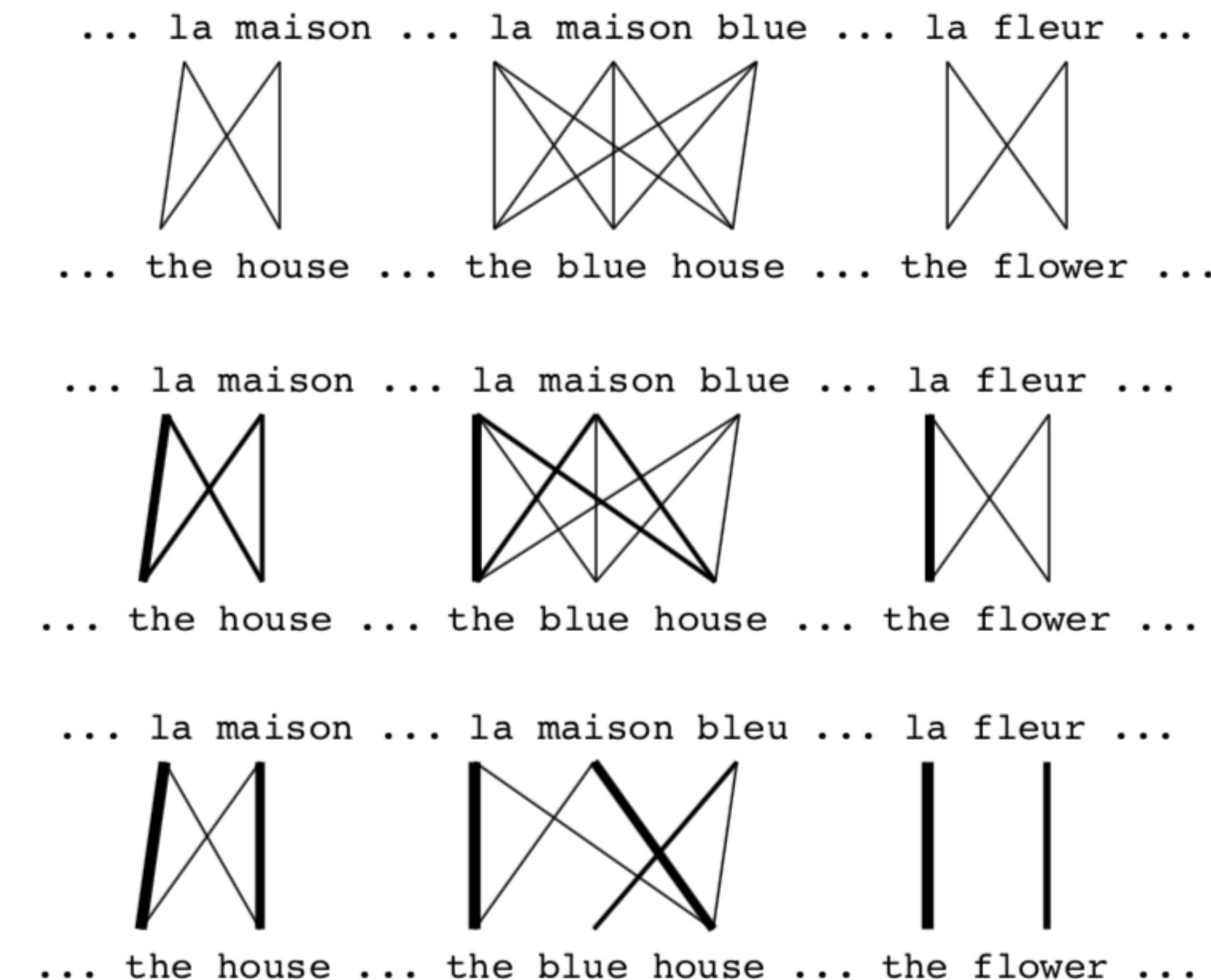
EM for IBM Model1 - Overview

- Start with all alignments equally likely
- In each iteration:
 - look at the entire dataset and sum the (expected) alignments we saw for each word and its possible translations (E-step)
 - Update the translation probabilities according to those global counts (M-step)...
 - ...which will update the alignment counts



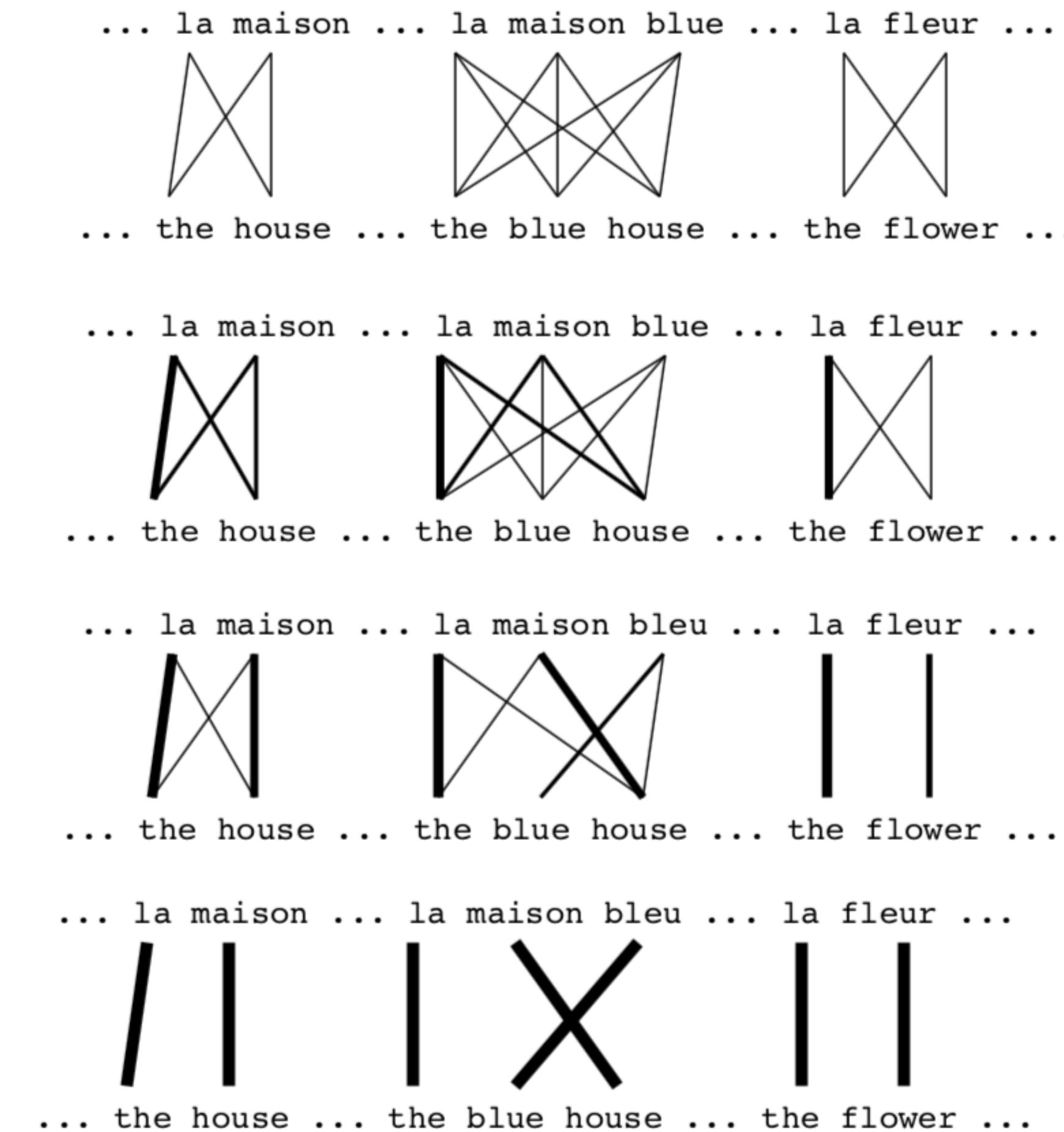
EM for IBM Model1 - Overview

- Start with all alignments equally likely
- In each iteration:
 - look at the entire dataset and sum the (expected) alignments we saw for each word and its possible translations (E-step)
 - Update the translation probabilities according to those global counts (M-step)...
 - ...which will update the alignment counts
 - Repeat steps above



EM for IBM Model 1 - Overview

- Start with all alignments equally likely
- In each iteration:
 - look at the entire dataset and sum the (expected) alignments we saw for each word and its possible translations (E-step)
 - Update the translation probabilities according to those global counts (M-step)...
 - ...which will update the alignment counts
- Repeat steps above
- Until convergence



EM for IBM Model1 - Computation

EM for IBM Model1 - Computation

- We need to compute:

EM for IBM Model1 - Computation

- We need to compute:
 - Expectation step: probability of alignments

$$p(a|\mathbf{e}, \mathbf{f})$$

EM for IBM Model1 - Computation

- We need to compute:
 - Expectation step: probability of alignments
 - Maximization step: probability of word translations

$$p(a|\mathbf{e}, \mathbf{f})$$

$$t(e|f; \mathbf{e}, \mathbf{f})$$

EM for IBM Model 1 - Expectation Step

EM for IBM Model 1 - Expectation Step

- Apply the chain rule

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

EM for IBM Model 1 - Expectation Step

- Apply the chain rule
- Numerator - IBM Model 1 definition

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

EM for IBM Model 1 - Expectation Step

- Apply the chain rule
- Numerator - IBM Model 1 definition
- Denominator:

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

$$p(\mathbf{e}|\mathbf{f}) = \sum_a p(\mathbf{e}, a|\mathbf{f})$$

$$= \sum_{a(1)=0}^{l_f} \dots \sum_{a(l_e)=0}^{l_f} p(\mathbf{e}, a|\mathbf{f})$$

$$= \sum_{a(1)=0}^{l_f} \dots \sum_{a(l_e)=0}^{l_f} \frac{\epsilon}{(l_f + 1)^{l_e}} \prod_{j=1}^{l_e} t(e_j | f_{a(j)})$$

EM for IBM Model 1 - Expectation Step

- Apply the chain rule
- Numerator - IBM Model 1 definition
- Denominator:
 - Marginalize over all possible alignments

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

$$p(\mathbf{e}|\mathbf{f}) = \sum_a p(\mathbf{e}, a|\mathbf{f})$$

$$= \sum_{a(1)=0}^{l_f} \dots \sum_{a(l_e)=0}^{l_f} p(\mathbf{e}, a|\mathbf{f})$$

$$= \sum_{a(1)=0}^{l_f} \dots \sum_{a(l_e)=0}^{l_f} \frac{\epsilon}{(l_f + 1)^{l_e}} \prod_{j=1}^{l_e} t(e_j | f_{a(j)})$$

EM for IBM Model 1 - Expectation Step

- Apply the chain rule
- Numerator - IBM Model 1 definition
- Denominator:
 - Marginalize over all possible alignments
 - IBM model 1 definition

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

$$p(\mathbf{e}|\mathbf{f}) = \sum_a p(\mathbf{e}, a|\mathbf{f})$$

$$= \sum_{a(1)=0}^{l_f} \dots \sum_{a(l_e)=0}^{l_f} p(\mathbf{e}, a|\mathbf{f})$$

$$= \sum_{a(1)=0}^{l_f} \dots \sum_{a(l_e)=0}^{l_f} \frac{\epsilon}{(l_f + 1)^{l_e}} \prod_{j=1}^{l_e} t(e_j | f_{a(j)})$$

EM for IBM Model 1 - Expectation Step

$$\begin{aligned} p(\mathbf{e}|\mathbf{f}) &= \sum_{a(1)=0}^{l_f} \dots \sum_{a(l_e)=0}^{l_f} \frac{\epsilon}{(l_f + 1)^{l_e}} \prod_{j=1}^{l_e} t(e_j | f_{a(j)}) \\ &= \frac{\epsilon}{(l_f + 1)^{l_e}} \sum_{a(1)=0}^{l_f} \dots \sum_{a(l_e)=0}^{l_f} \prod_{j=1}^{l_e} t(e_j | f_{a(j)}) \\ &= \frac{\epsilon}{(l_f + 1)^{l_e}} \prod_{j=1}^{l_e} \sum_{i=0}^{l_f} t(e_j | f_i) \end{aligned}$$

EM for IBM Model 1 - Expectation Step

- We want to get rid of the exponential number of multiplications

$$\begin{aligned} p(\mathbf{e}|\mathbf{f}) &= \sum_{a(1)=0}^{l_f} \dots \sum_{a(l_e)=0}^{l_f} \frac{\epsilon}{(l_f + 1)^{l_e}} \prod_{j=1}^{l_e} t(e_j | f_{a(j)}) \\ &= \frac{\epsilon}{(l_f + 1)^{l_e}} \sum_{a(1)=0}^{l_f} \dots \sum_{a(l_e)=0}^{l_f} \prod_{j=1}^{l_e} t(e_j | f_{a(j)}) \\ &= \frac{\epsilon}{(l_f + 1)^{l_e}} \prod_{j=1}^{l_e} \sum_{i=0}^{l_f} t(e_j | f_i) \end{aligned}$$

EM for IBM Model 1 - Expectation Step

- We want to get rid of the exponential number of multiplications
- Move the constants out

$$\begin{aligned} p(\mathbf{e}|\mathbf{f}) &= \sum_{a(1)=0}^{l_f} \dots \sum_{a(l_e)=0}^{l_f} \frac{\epsilon}{(l_f + 1)^{l_e}} \prod_{j=1}^{l_e} t(e_j | f_{a(j)}) \\ &= \frac{\epsilon}{(l_f + 1)^{l_e}} \sum_{a(1)=0}^{l_f} \dots \sum_{a(l_e)=0}^{l_f} \prod_{j=1}^{l_e} t(e_j | f_{a(j)}) \\ &= \frac{\epsilon}{(l_f + 1)^{l_e}} \prod_{j=1}^{l_e} \sum_{i=0}^{l_f} t(e_j | f_i) \end{aligned}$$

EM for IBM Model 1 - Expectation Step

- We want to get rid of the exponential number of multiplications
- Move the constants out
- Last trick - change sum of products to product of sums

$$\begin{aligned} p(\mathbf{e}|\mathbf{f}) &= \sum_{a(1)=0}^{l_f} \dots \sum_{a(l_e)=0}^{l_f} \frac{\epsilon}{(l_f + 1)^{l_e}} \prod_{j=1}^{l_e} t(e_j | f_{a(j)}) \\ &= \frac{\epsilon}{(l_f + 1)^{l_e}} \sum_{a(1)=0}^{l_f} \dots \sum_{a(l_e)=0}^{l_f} \prod_{j=1}^{l_e} t(e_j | f_{a(j)}) \\ &= \frac{\epsilon}{(l_f + 1)^{l_e}} \prod_{j=1}^{l_e} \sum_{i=0}^{l_f} t(e_j | f_i) \end{aligned}$$

EM for IBM Model 1 - Expectation Step

EM for IBM Model 1 - Expectation Step

- So finally we got:

EM for IBM Model 1 - Expectation Step

- So finally we got:

$$p(\mathbf{a}|\mathbf{e}, \mathbf{f}) = p(\mathbf{e}, \mathbf{a}|\mathbf{f})/p(\mathbf{e}|\mathbf{f})$$

$$= \frac{\frac{\epsilon}{(l_f+1)^{l_e}} \prod_{j=1}^{l_e} t(e_j|f_{a(j)})}{\frac{\epsilon}{(l_f+1)^{l_e}} \prod_{j=1}^{l_e} \sum_{i=0}^{l_f} t(e_j|f_i)}$$

$$= \prod_{j=1}^{l_e} \frac{t(e_j|f_{a(j)})}{\sum_{i=0}^{l_f} t(e_j|f_i)}$$

EM for IBM Model 1 - Expectation Step

- So finally we got:

$$p(\mathbf{a}|\mathbf{e}, \mathbf{f}) = p(\mathbf{e}, \mathbf{a}|\mathbf{f})/p(\mathbf{e}|\mathbf{f})$$

- Probability of alignment given a sentence pair

$$= \frac{\frac{\epsilon}{(l_f+1)^{l_e}} \prod_{j=1}^{l_e} t(e_j|f_{a(j)})}{\frac{\epsilon}{(l_f+1)^{l_e}} \prod_{j=1}^{l_e} \sum_{i=0}^{l_f} t(e_j|f_i)}$$

$$= \prod_{j=1}^{l_e} \frac{t(e_j|f_{a(j)})}{\sum_{i=0}^{l_f} t(e_j|f_i)}$$

EM for IBM Model 1 - Expectation Step

- So finally we got:

$$p(\mathbf{a}|\mathbf{e}, \mathbf{f}) = p(\mathbf{e}, \mathbf{a}|\mathbf{f})/p(\mathbf{e}|\mathbf{f})$$

- Probability of alignment given a sentence pair

$$= \frac{\frac{\epsilon}{(l_f+1)^{l_e}} \prod_{j=1}^{l_e} t(e_j|f_{a(j)})}{\frac{\epsilon}{(l_f+1)^{l_e}} \prod_{j=1}^{l_e} \sum_{i=0}^{l_f} t(e_j|f_i)}$$

- Based on the translation probabilities alone

$$= \prod_{j=1}^{l_e} \frac{t(e_j|f_{a(j)})}{\sum_{i=0}^{l_f} t(e_j|f_i)}$$

EM for IBM Model 1 - Expectation Step

- So finally we got:

$$p(\mathbf{a}|\mathbf{e}, \mathbf{f}) = p(\mathbf{e}, \mathbf{a}|\mathbf{f})/p(\mathbf{e}|\mathbf{f})$$

- Probability of alignment given a sentence pair
- Based on the translation probabilities alone
- We will use this to get the expected alignment counts

$$\begin{aligned} &= \frac{\frac{\epsilon}{(l_f+1)^{l_e}} \prod_{j=1}^{l_e} t(e_j|f_{a(j)})}{\frac{\epsilon}{(l_f+1)^{l_e}} \prod_{j=1}^{l_e} \sum_{i=0}^{l_f} t(e_j|f_i)} \\ &= \prod_{j=1}^{l_e} \frac{t(e_j|f_{a(j)})}{\sum_{i=0}^{l_f} t(e_j|f_i)} \end{aligned}$$

EM for IBM Model 1 - Maximization Step

EM for IBM Model 1 - Maximization Step

- We want to collect alignment counts to compute the new translation parameters, using the **current** translation parameters

EM for IBM Model 1 - Maximization Step

- We want to collect alignment counts to compute the new translation parameters, using the **current** translation parameters
- For each possible pair (e,f) in each example (e,f) sum the expected counts of this translation

EM for IBM Model 1 - Maximization Step

- We want to collect alignment counts to compute the new translation parameters, using the **current** translation parameters
- For each possible pair (e,f) in each example (\mathbf{e},\mathbf{f}) sum the expected counts of this translation

$$c(e|f; \mathbf{e}, \mathbf{f}) = \sum_a p(a|\mathbf{e}, \mathbf{f}) \sum_{j=1}^{l_e} \delta(e, e_j) \delta(f, f_{a(j)})$$

EM for IBM Model 1 - Maximization Step

- We want to collect alignment counts to compute the new translation parameters, using the **current** translation parameters
- For each possible pair (e,f) in each example (\mathbf{e}, \mathbf{f}) sum the expected counts of this translation

$$c(e|f; \mathbf{e}, \mathbf{f}) = \sum_a p(a|\mathbf{e}, \mathbf{f}) \sum_{j=1}^{l_e} \delta(e, e_j) \delta(f, f_{a(j)})$$

$$c(e|f; \mathbf{e}, \mathbf{f}) = \frac{t(e|f)}{\sum_{i=0}^{l_f} t(e|f_i)} \sum_{j=1}^{l_e} \delta(e, e_j) \sum_{i=0}^{l_f} \delta(f, f_i)$$

EM for IBM Model 1 - Maximization Step

- We want to collect alignment counts to compute the new translation parameters, using the **current** translation parameters
- For each possible pair (e,f) in each example (\mathbf{e}, \mathbf{f}) sum the expected counts of this translation

$$\begin{array}{ll} p(\text{the}|la) = 0.7 & p(\text{house}|la) = 0.05 \\ p(\text{the}|maison) = 0.1 & p(\text{house}|maison) = 0.8 \end{array}$$

$$c(e|f; \mathbf{e}, \mathbf{f}) = \sum_a p(a|\mathbf{e}, \mathbf{f}) \sum_{j=1}^{l_e} \delta(e, e_j) \delta(f, f_{a(j)})$$

$$c(e|f; \mathbf{e}, \mathbf{f}) = \frac{t(e|f)}{\sum_{i=0}^{l_f} t(e|f_i)} \sum_{j=1}^{l_e} \delta(e, e_j) \sum_{i=0}^{l_f} \delta(f, f_i)$$

EM for IBM Model 1 - Maximization Step

- We want to collect alignment counts to compute the new translation parameters, using the **current** translation parameters
- For each possible pair (e, f) in each example (\mathbf{e}, \mathbf{f}) sum the expected counts of this translation

$$\begin{aligned} p(\text{the|la}) &= 0.7 & p(\text{house|la}) &= 0.05 \\ p(\text{the|maison}) &= 0.1 & p(\text{house|maison}) &= 0.8 \end{aligned} \rightarrow \begin{matrix} \text{la maison} \\ \diagup 0.059 \\ \diagdown 0.125 \\ \text{the house} \end{matrix} \quad 0.875 \quad 0.941$$

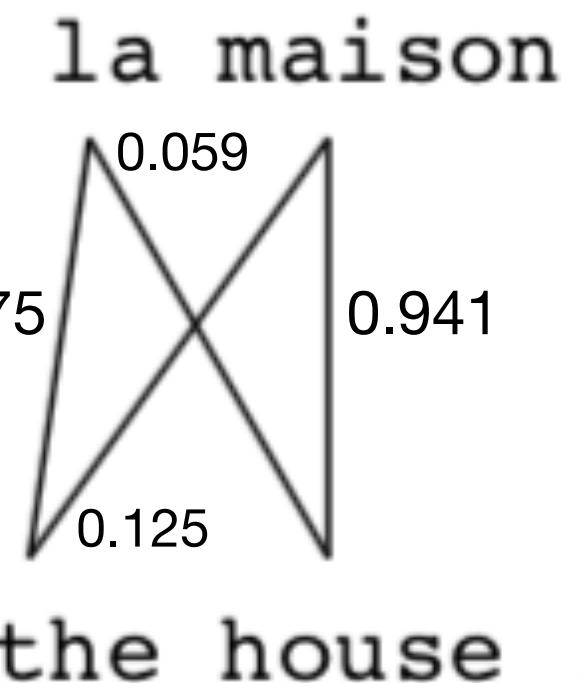
$$c(e|f; \mathbf{e}, \mathbf{f}) = \sum_a p(a|\mathbf{e}, \mathbf{f}) \sum_{j=1}^{l_e} \delta(e, e_j) \delta(f, f_{a(j)})$$

$$c(e|f; \mathbf{e}, \mathbf{f}) = \frac{t(e|f)}{\sum_{i=0}^{l_f} t(e|f_i)} \sum_{j=1}^{l_e} \delta(e, e_j) \sum_{i=0}^{l_f} \delta(f, f_i)$$

EM for IBM Model 1 - Maximization Step

- We want to collect alignment counts to compute the new translation parameters, using the **current** translation parameters
- For each possible pair (e, f) in each example (\mathbf{e}, \mathbf{f}) sum the expected counts of this translation
- Maximization: after we do this over the entire corpus, sum and normalize to get the **new** parameters

$$p(\text{the}|\text{la}) = 0.7 \quad p(\text{house}|\text{la}) = 0.05 \\ p(\text{the}|\text{maison}) = 0.1 \quad p(\text{house}|\text{maison}) = 0.8 \rightarrow$$



$$c(e|f; \mathbf{e}, \mathbf{f}) = \sum_a p(a|\mathbf{e}, \mathbf{f}) \sum_{j=1}^{l_e} \delta(e, e_j) \delta(f, f_{a(j)})$$

$$c(e|f; \mathbf{e}, \mathbf{f}) = \frac{t(e|f)}{\sum_{i=0}^{l_f} t(e|f_i)} \sum_{j=1}^{l_e} \delta(e, e_j) \sum_{i=0}^{l_f} \delta(f, f_i)$$

$$t(e|f; \mathbf{e}, \mathbf{f}) = \frac{\sum_{(\mathbf{e}, \mathbf{f})} c(e|f; \mathbf{e}, \mathbf{f}))}{\sum_f \sum_{(\mathbf{e}, \mathbf{f})} c(e|f; \mathbf{e}, \mathbf{f}))}$$

EM for IBM Model 1 - Pseudo Code

EM for IBM Model 1 - Pseudo Code

- Finally:

EM for IBM Model 1 - Pseudo Code

- Finally:

```

Input: set of sentence pairs (e, f)
Output: translation prob.  $t(e|f)$ 

1: initialize  $t(e|f)$  uniformly
2: while not converged do
3:   // initialize
4:   count( $e|f$ ) = 0 for all  $e, f$ 
5:   total( $f$ ) = 0 for all  $f$ 
6:   for all sentence pairs (e,f) do
7:     // compute normalization
8:     for all words  $e$  in e do
9:       s-total( $e$ ) = 0
10:      for all words  $f$  in f do
11:        s-total( $e$ ) +=  $t(e|f)$ 
12:      end for
13:    end for
14:    // collect counts
15:    for all words  $e$  in e do
16:      for all words  $f$  in f do
17:        count( $e|f$ ) +=  $\frac{t(e|f)}{\text{s-total}(e)}$ 
18:        total( $f$ ) +=  $\frac{t(e|f)}{\text{s-total}(e)}$ 
19:      end for
20:    end for
21:  end for
22:  // estimate probabilities
23:  for all foreign words  $f$  do
24:    for all English words  $e$  do
25:       $t(e|f) = \frac{\text{count}(e|f)}{\text{total}(f)}$ 
26:    end for
27:  end for
28: end while

```

EM for IBM Model 1 - Pseudo Code

- Finally:
- You will implement this in Exercise 1

```

Input: set of sentence pairs (e, f)
Output: translation prob.  $t(e|f)$ 

1: initialize  $t(e|f)$  uniformly
2: while not converged do
3:   // initialize
4:   count( $e|f$ ) = 0 for all  $e, f$ 
5:   total( $f$ ) = 0 for all  $f$ 
6:   for all sentence pairs (e,f) do
7:     // compute normalization
8:     for all words  $e$  in e do
9:       s-total( $e$ ) = 0
10:      for all words  $f$  in f do
11:        s-total( $e$ ) +=  $t(e|f)$ 
12:      end for
13:    end for
14:    // collect counts
15:    for all words  $e$  in e do
16:      for all words  $f$  in f do
17:        count( $e|f$ ) +=  $\frac{t(e|f)}{\text{s-total}(e)}$ 
18:        total( $f$ ) +=  $\frac{t(e|f)}{\text{s-total}(e)}$ 
19:      end for
20:    end for
21:  end for
22:  // estimate probabilities
23:  for all foreign words  $f$  do
24:    for all English words  $e$  do
25:       $t(e|f) = \frac{\text{count}(e|f)}{\text{total}(f)}$ 
26:    end for
27:  end for
28: end while

```

Alignment Error Rate (AER)

Alignment Error Rate (AER)

- How can we measure the alignment quality?

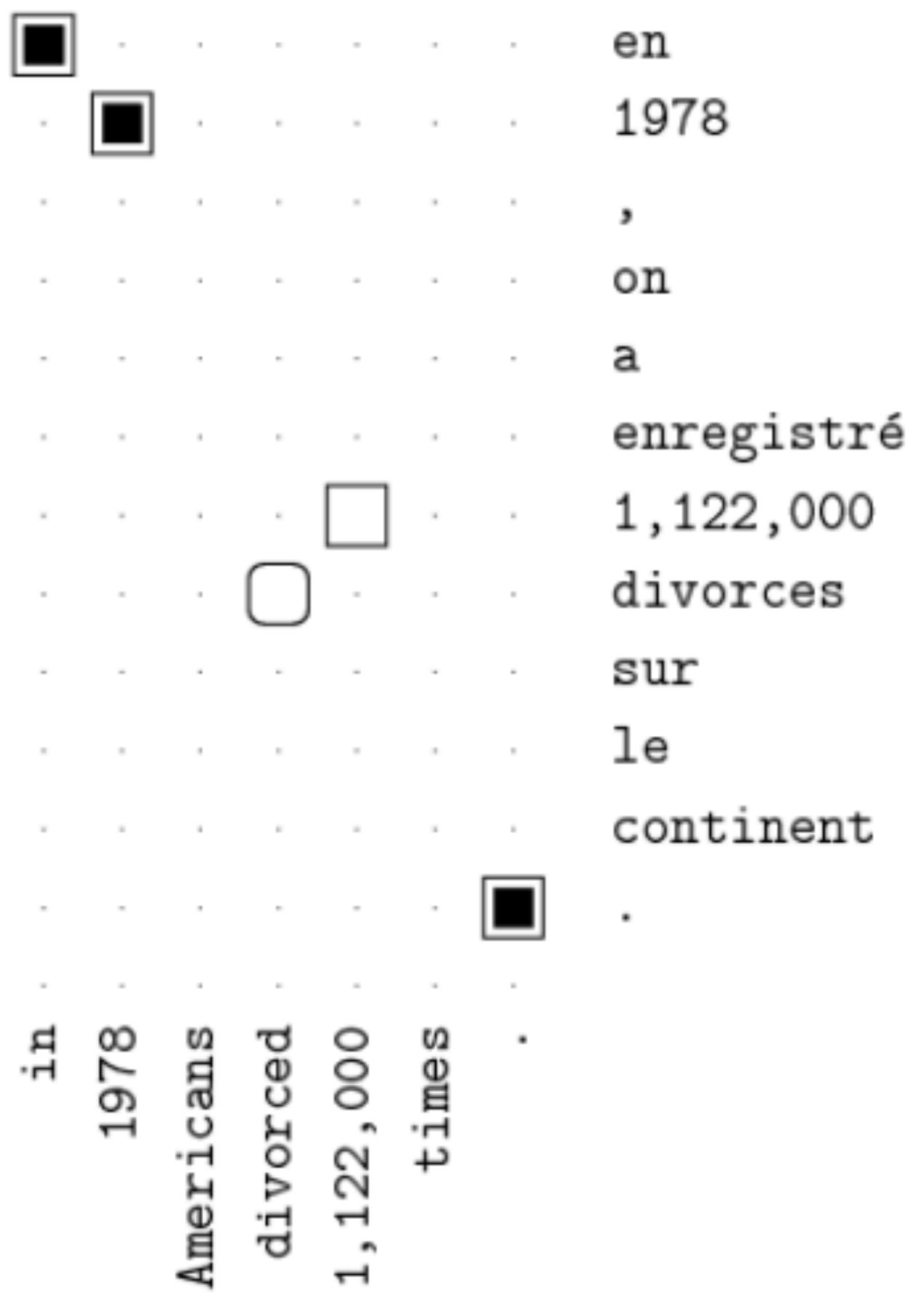
Alignment Error Rate (AER)

- How can we measure the alignment quality?
- AER - Och and Ney, 2000

= Sure
 = Possible
 = Predicted

$$AER(A, S, P) = \left(1 - \frac{|A \cap S| + |A \cap P|}{|A| + |S|}\right)$$

$$= \left(1 - \frac{3+3}{3+4}\right) = \frac{1}{7}$$



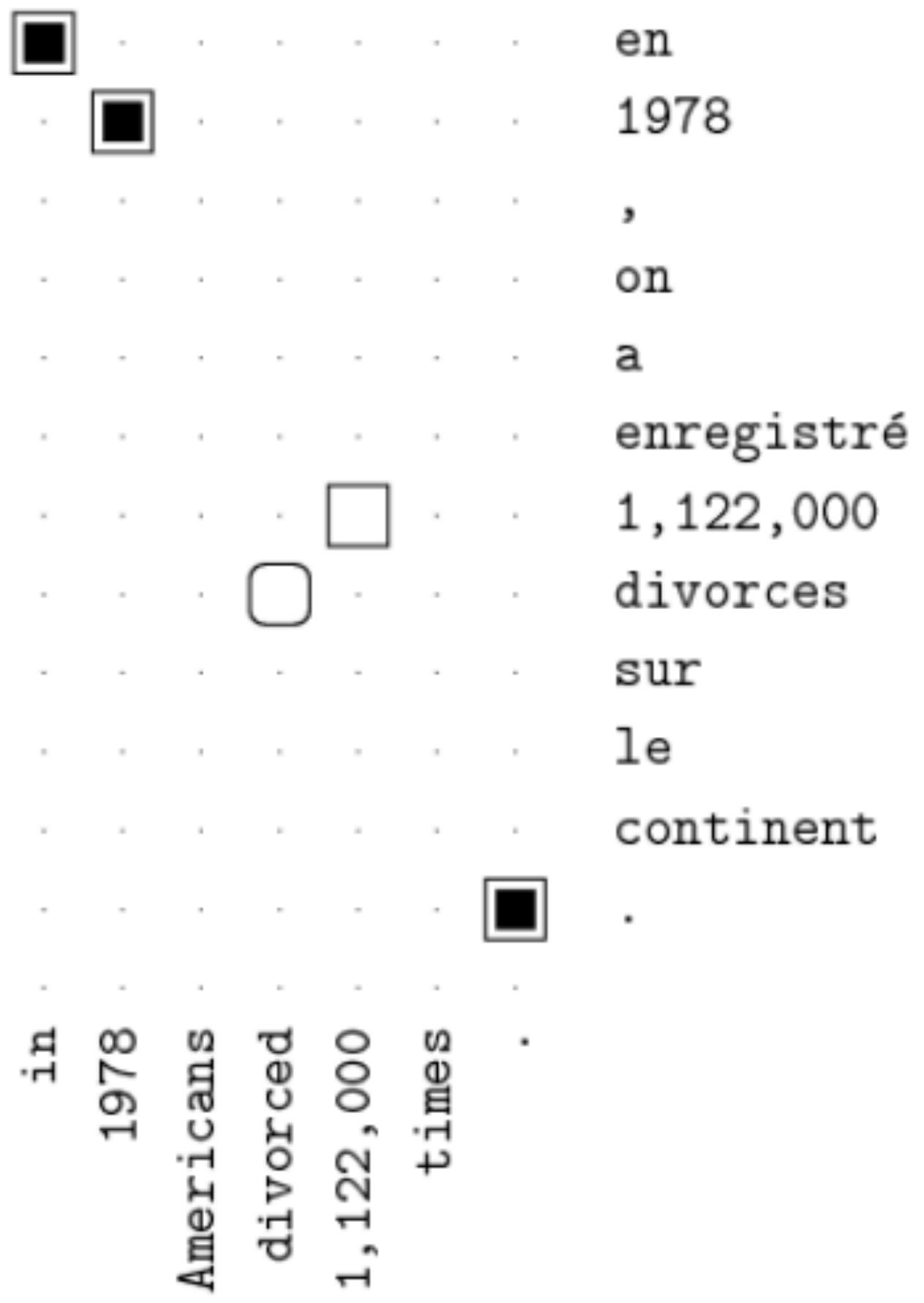
Alignment Error Rate (AER)

- How can we measure the alignment quality?
- AER - Och and Ney, 2000
- Possible contains Sure

= Sure
 = Possible
 = Predicted

$$AER(A, S, P) = \left(1 - \frac{|A \cap S| + |A \cap P|}{|A| + |S|}\right)$$

$$= \left(1 - \frac{3+3}{3+4}\right) = \frac{1}{7}$$



in 1978 Americans divorced 1,122,000 times continent .

en 1978 , on a enregistré 1,122,000 divorces sur le continent .

Alignment Error Rate (AER)

- How can we measure the alignment quality?
- AER - Och and Ney, 2000
- Possible contains Sure
- Must hit all Sure to be perfect, ok to not cover all probable

= Sure
 = Possible
 = Predicted

$$\begin{aligned} AER(A, S, P) &= \left(1 - \frac{|A \cap S| + |A \cap P|}{|A| + |S|}\right) \\ &= \left(1 - \frac{3+3}{3+4}\right) = \frac{1}{7} \end{aligned}$$

en 1978 , on a enregistré 1,122,000 divorces sur le continent .

Summary

Summary

- IBM model 1

Summary

- IBM model 1
 - Generative model, based on:

Summary

- IBM model 1
 - Generative model, based on:
 - Alignments (latent variables)



Summary

- IBM model 1
 - Generative model, based on:
 - Alignments (latent variables)
 - Which are used to calculate translation parameters



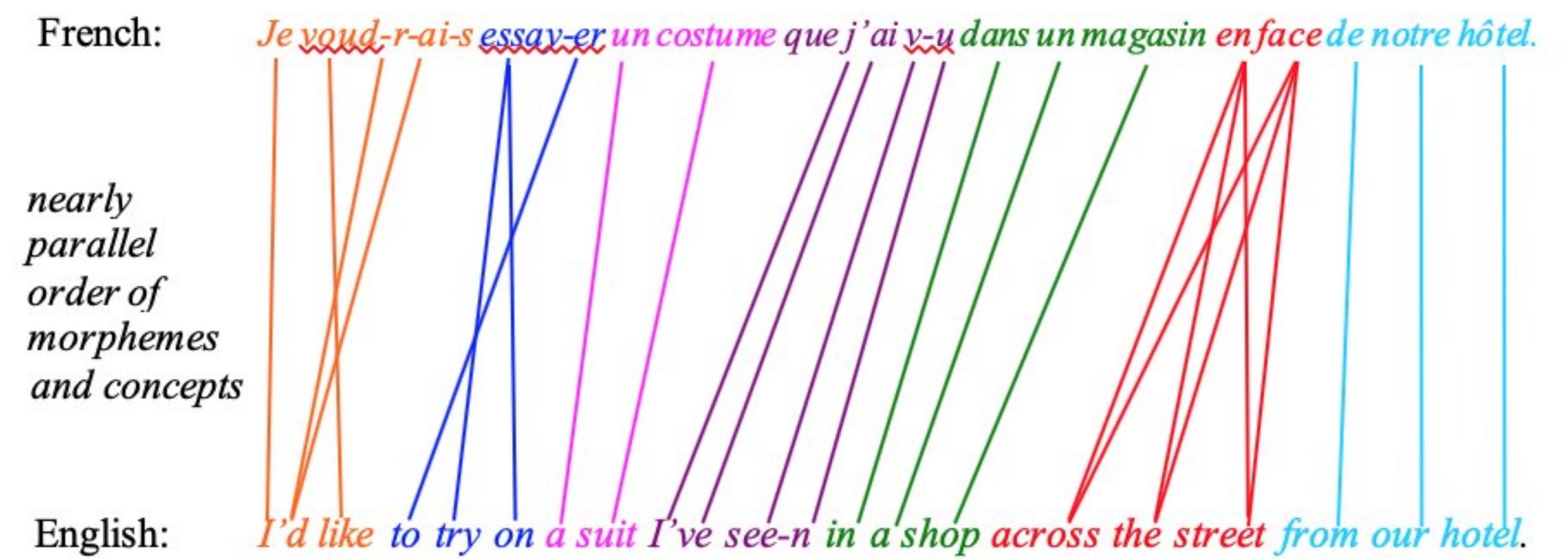
Summary

- IBM model 1
 - Generative model, based on:
 - Alignments (latent variables)
 - Which are used to calculate translation parameters
 - Using Expectation Maximization



Summary

- IBM model 1
 - Generative model, based on:
 - Alignments (latent variables)
 - Which are used to calculate translation parameters
 - Using Expectation Maximization
 - Exercise 1 - May 18th



Summary

- IBM model 1
 - Generative model, based on:
 - Alignments (latent variables)
 - Which are used to calculate translation parameters
 - Using Expectation Maximization
 - Exercise 1 - May 18th

French:
Je voud-r-ai-s essay-er un costume que j'ai vu dans un magasin en face de notre hôtel.

English:
I'd like to try on a suit I've seen in a shop across the street from our hotel.

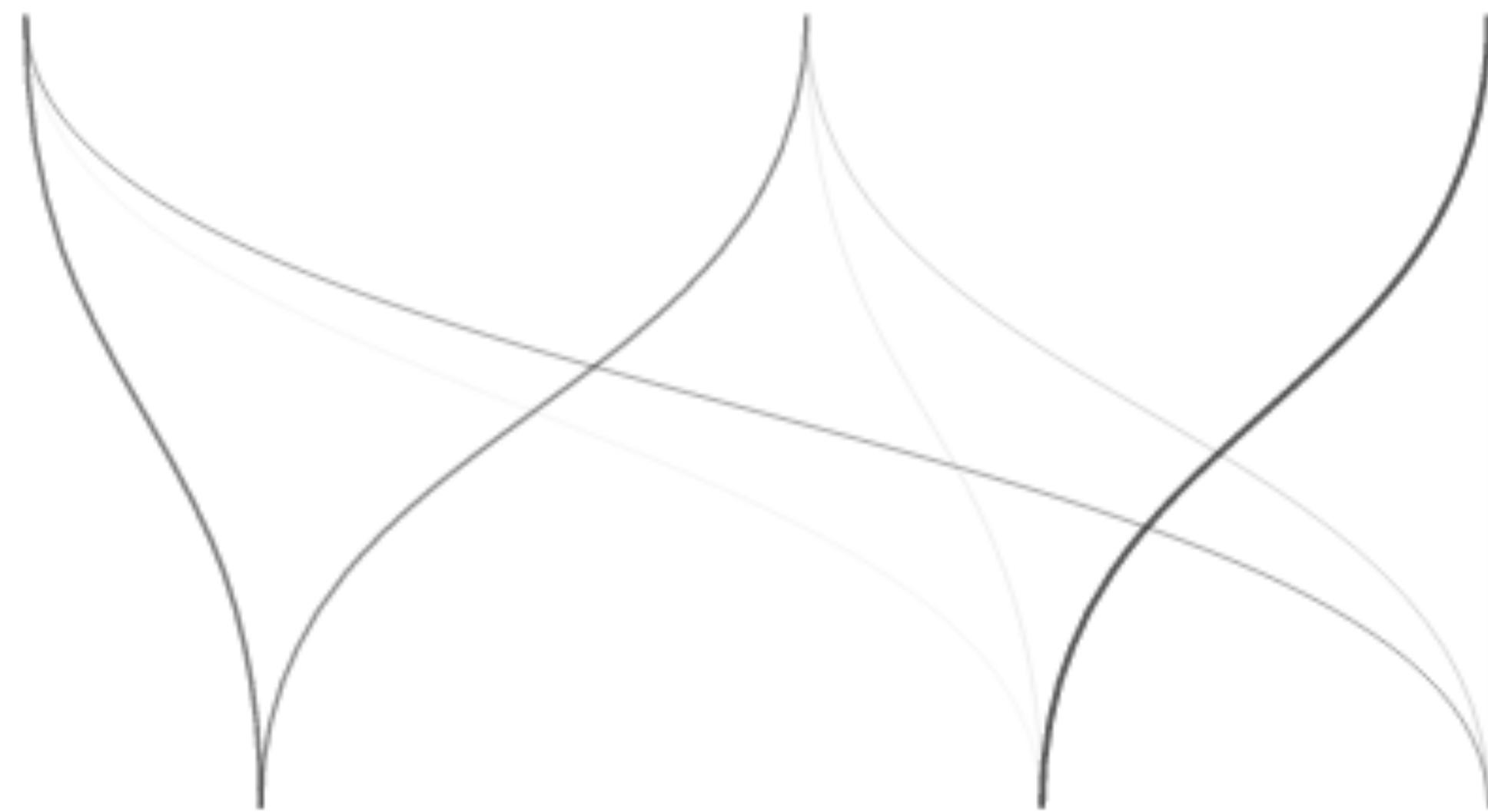
nearby parallel order of morphemes and concepts

Turkish:
Otel-imiz-in karşı-sın-da-ki dükkân-da gör-düğ-üm bir elbise-yi dene-mek iste-r-im.

English:
I'd like to try on a suit I've seen in a shop across the street from our hotel.

inverse order of morphemes and concepts

Any Questions ?



Questions diverses ?