Lexical Translation Models 1

January 30, 2014

Lexical Translation

- How do we translate a word? Look it up in the dictionary

$$Haus : house, home, shell, household$$

- Multiple translations
  - Different word senses, different registers, different inflections (?)
  - *house, home* are common
  - *shell* is specialized (the Haus of a snail is a shell)

How common is each?

Translation	Count
house	5000
home	2000
shell	100
household	80

MLE

$$\hat{p}_{\mathrm{MLE}}(e \mid \texttt{Haus}) = \begin{cases} 0.696 & \text{if } e = \texttt{house} \\ 0.279 & \text{if } e = \texttt{home} \\ 0.014 & \text{if } e = \texttt{shell} \\ 0.011 & \text{if } e = \texttt{household} \\ 0 & \text{otherwise} \end{cases}$$

Lexical Translation

- Goal: a model $p(\mathbf{e} \mid \mathbf{f}, m)$
- where $\mathbf{e}$ and $\mathbf{f}$ are complete English and Foreign sentences

$$\mathbf{e} = \langle e_1, e_2, \ldots, e_m \rangle \qquad \mathbf{f} = \langle f_1, f_2, \ldots, f_n \rangle$$

Lexical Translation

- Goal: a model $p(\mathbf{e} \mid \mathbf{f}, m)$
- where $\mathbf{e}$ and $\mathbf{f}$ are complete English and Foreign sentences
- Lexical translation makes the following ***assumptions***:
  - Each word in e_i in $\mathbf{e}$ is generated from exactly one word in $\mathbf{f}$
  - Thus, we have an *alignment* a_i that indicates which word e_i "came from", specifically it came from f_{a_i}
  - Given the alignments $\mathbf{a}$, translation decisions are conditionally independent of each other and depend *only* on the aligned source word f_{a_i}

Lexical Translation

- Putting our assumptions together, we have:

$$p(\mathbf{e} \mid \mathbf{f}, m) = \sum_{\mathbf{a} \in [0,n]^m} p(\mathbf{a} \mid \mathbf{f}, m) \times \prod_{i=1}^{m} p(e_i \mid f_{a_i})$$

Alignment $\times$ Translation | Alignment

Lexical Translation

$$p(e_i \mid f_{a_i})$$

$p(\texttt{house} \mid \texttt{Haus})$ $p(\texttt{shell} \mid \texttt{Haus})$

$p(\texttt{declaration} \mid \texttt{Unabhaenigkeitserkaerung})$

Remember bigram models...

Lexical Translation

- Putting our assumptions together, we have:

$$p(\mathbf{e} \mid \mathbf{f}, m) = \sum_{\mathbf{a} \in [0,n]^m} p(\mathbf{a} \mid \mathbf{f}, m) \times \prod_{i=1}^{m} p(e_i \mid f_{a_i})$$

Alignment $\times$ Translation | Alignment

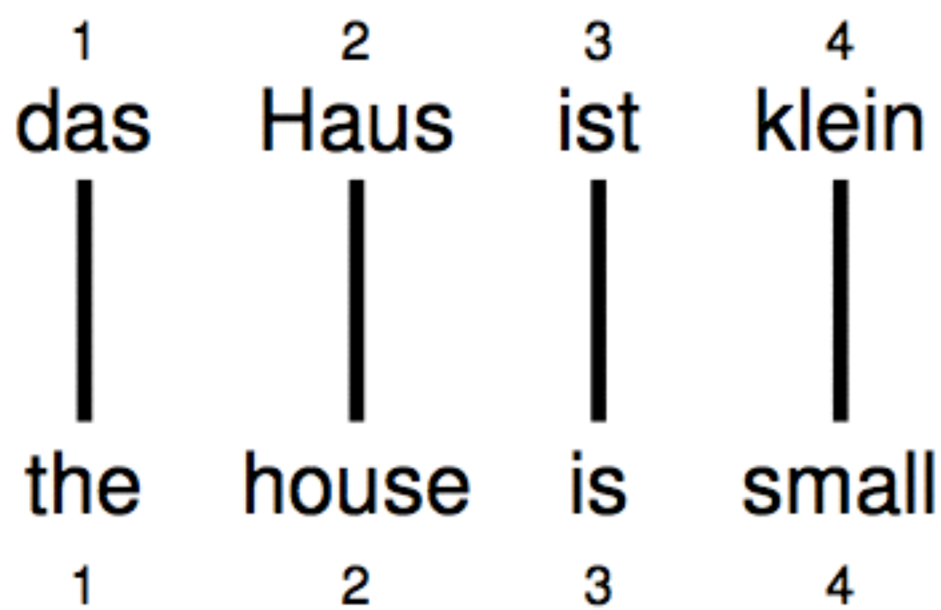
Alignment

$$p(\mathbf{a} \mid \mathbf{f}, m)$$

Most of the action for the first 10 years of MT was here. Words weren't the problem, word *order* was hard.

Alignment

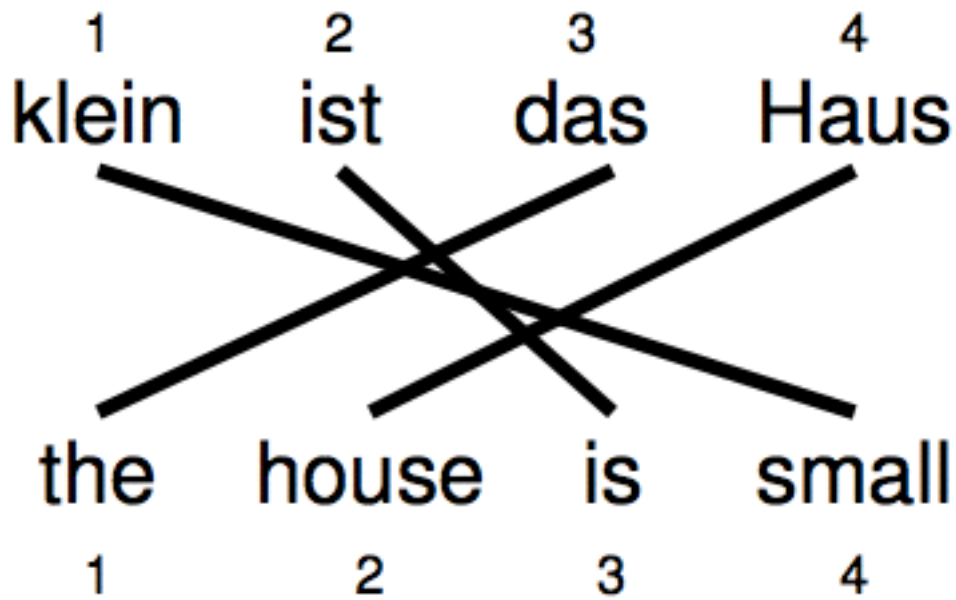
- Alignments can be visualized in by drawing links between two sentences, and they are represented as vectors of positions:



$$\mathbf{a} = (1, 2, 3, 4)^\top$$

Reordering

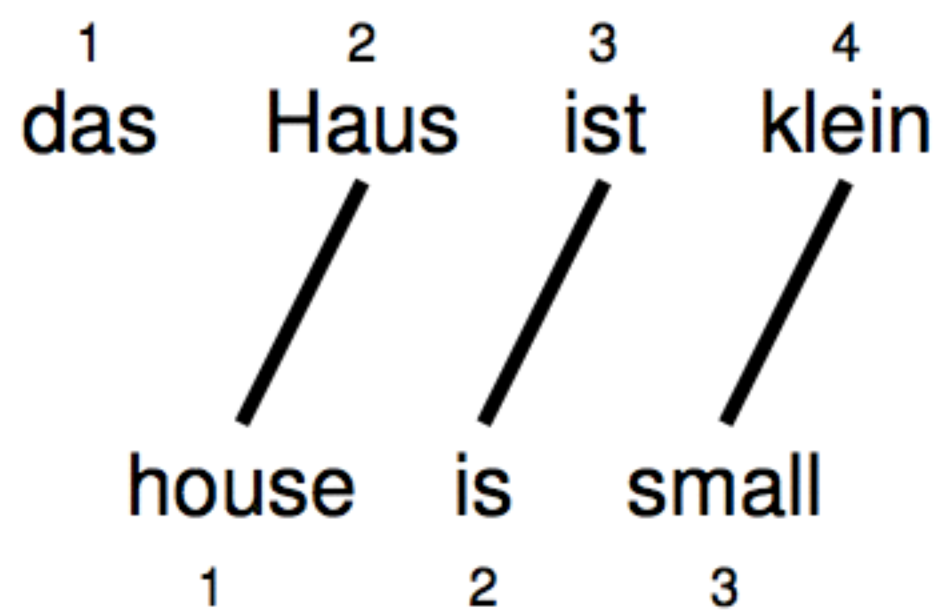
- Words may be reordered during translation.



$$\mathbf{a} = (3, 4, 2, 1)^\top$$

Word Dropping

- A source word may not be translated at all



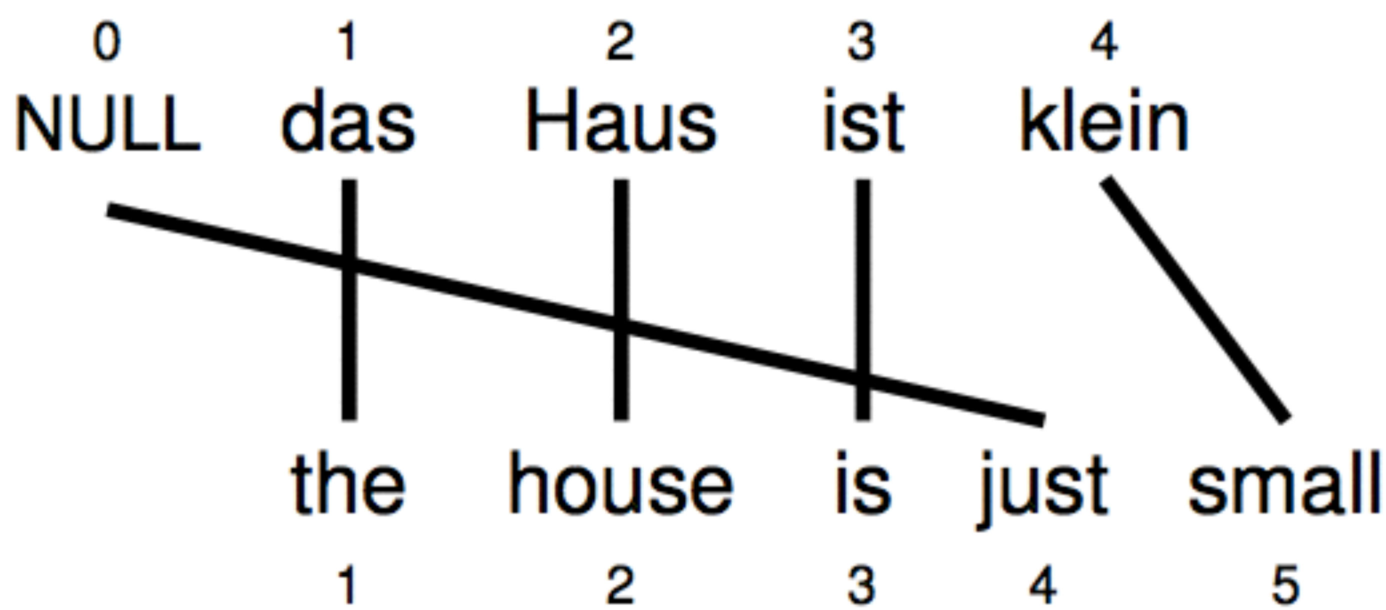
$$\mathbf{a} = (2, 3, 4)^\top$$

Word Insertion

- Words may be inserted during translation

  English just does not have an equivalent

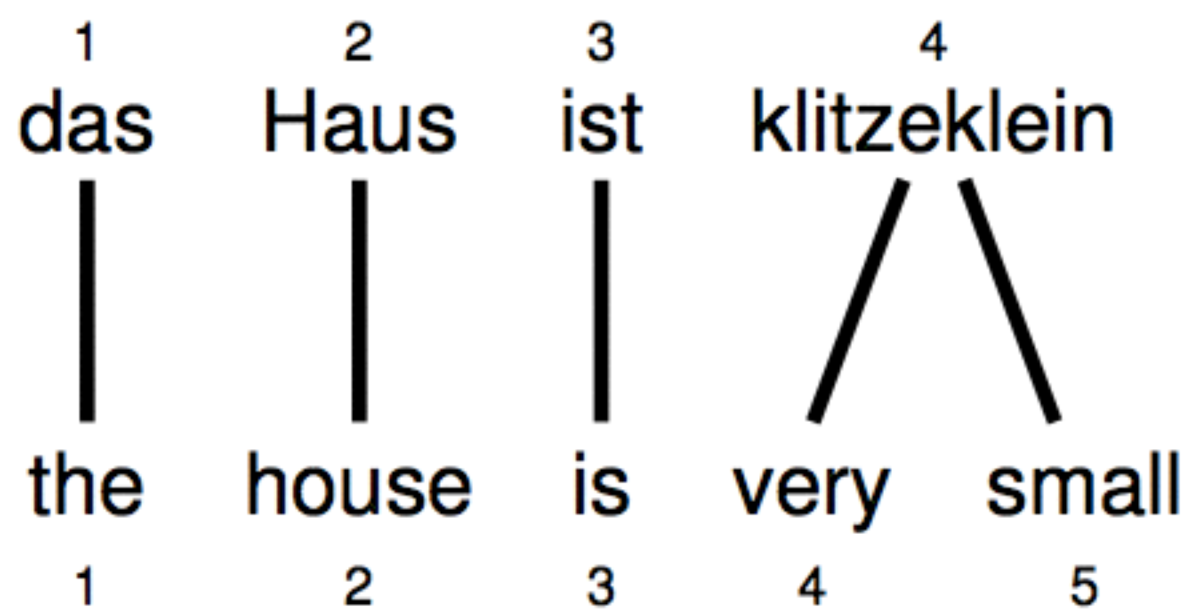
  But it must be explained - we typically assume every source sentence contains a NULL token



$$\mathbf{a} = (1, 2, 3, 0, 4)^\top$$

One-to-many Translation

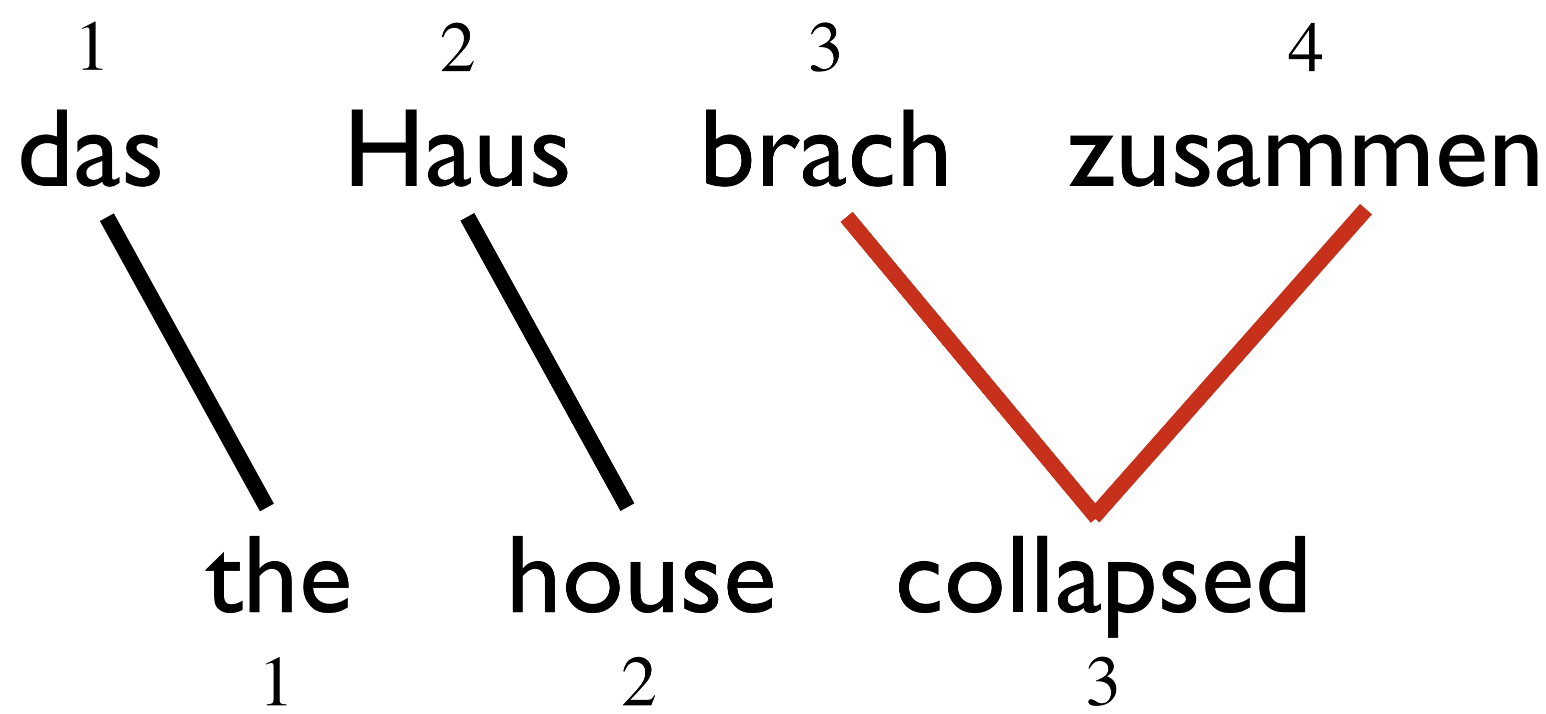
- A source word may translate into **more than one** target word



$$\mathbf{a} = (1, 2, 3, 4, 4)^\top$$

Many-to-one Translation

- **More than one source word** may **not** translate as a unit in lexical translation



$\mathbf{a} = ???$ $\quad\quad$ $\mathbf{a} = (1, 2, (3, 4)^\top)^\top$ **?**

IBM Model 1

- Simplest possible lexical translation model
- Additional assumptions
  - The m alignment decisions are independent
  - The alignment distribution for each $a_i$is uniform over all source words and NULL

$$
\begin{aligned}
&\text{for each } i \in [1, 2, \ldots, m] \\
&\qquad\qquad a_i \sim \mathrm{Uniform}(0, 1, 2, \ldots, n) \\
&\qquad\qquad e_i \sim \mathrm{Categorical}(\boldsymbol{\theta}_{f_{a_i}})
\end{aligned}
$$

Historical Note

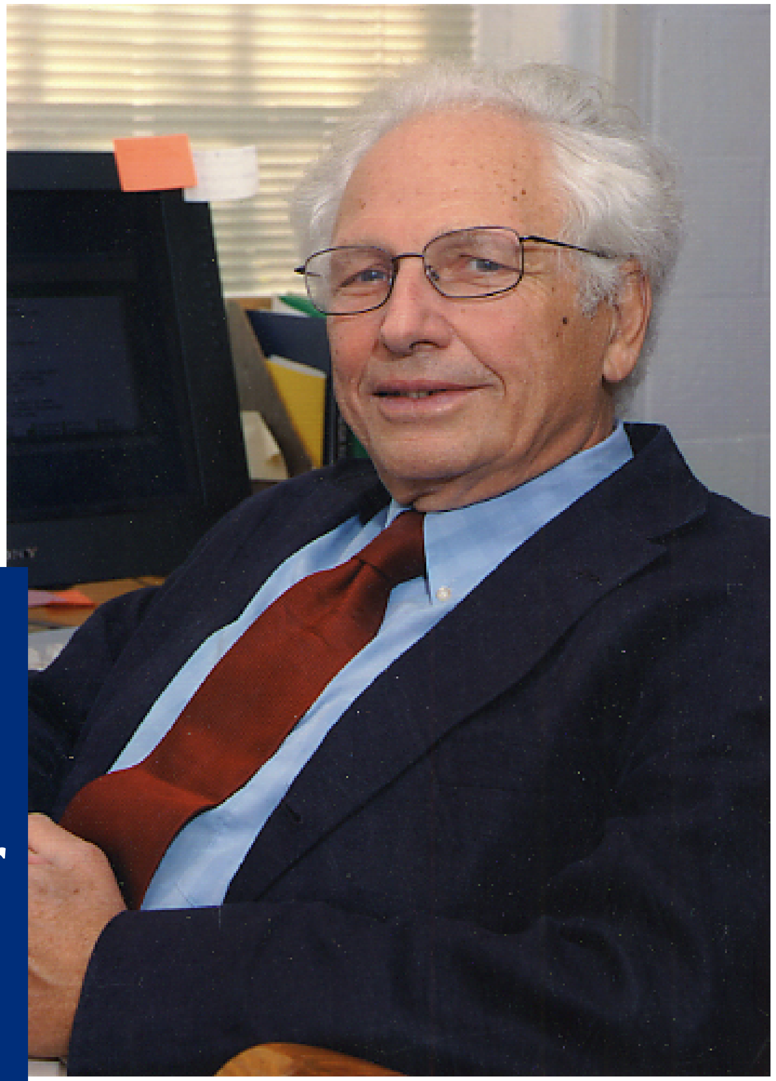
IBM Models

Renaissance

"The validity of a statistical (information theoretic) approach to MT has indeed been recognized, as the authors mention, by Weaver as early as 1949. And was universally recognized as mistaken by 1950 (cf. Hutchins, MT – Past, Present, Future, Ellis Horwood, 1986, p. 30ff and references therein). The crude force of computers is not science. The paper is simply beyond the scope of COLING."

1. Introduction

The growing availability of bilingual, machine-readable texts has stimulated interest



Fred Jelinek (1932-2010)

The Center For Language and Speech Processing

at the Johns Hopkins University

IBM Model 1

$$\boxed{\text{for each } i \in [1, 2, \ldots, m]}$$

$$a_i \sim \text{Uniform}(0, 1, 2, \ldots, n)$$

$$e_i \sim \text{Categorical}(\boldsymbol{\theta}_{f_{a_i}})$$

$$p(\mathbf{e}, \mathbf{a} \mid \mathbf{f}, m) = \prod_{i=1}^{m}$$

IBM Model 1

$$\text{for each } i \in [1, 2, \ldots, m]$$
$$\boxed{a_i \sim \text{Uniform}(0, 1, 2, \ldots, n)}$$
$$e_i \sim \text{Categorical}(\boldsymbol{\theta}_{f_{a_i}})$$

$$p(\mathbf{e}, \mathbf{a} \mid \mathbf{f}, m) = \prod_{i=1}^{m} \frac{1}{1+n}$$

IBM Model 1

$$\text{for each } i \in [1, 2, \ldots, m]$$

$$a_i \sim \text{Uniform}(0, 1, 2, \ldots, n)$$

$$\boxed{e_i \sim \text{Categorical}(\boldsymbol{\theta}_{f_{a_i}})}$$

$$p(\mathbf{e}, \mathbf{a} \mid \mathbf{f}, m) = \prod_{i=1}^{m} \frac{1}{1+n}\, p(e_i \mid f_{a_i})$$

IBM Model 1

$$\text{for each } i \in [1, 2, \ldots, m]$$

$$a_i \sim \text{Uniform}(0, 1, 2, \ldots, n)$$

$$\boxed{e_i \sim \text{Categorical}(\boldsymbol{\theta}_{f_{a_i}})}$$

$$p(\mathbf{e}, \mathbf{a} \mid \mathbf{f}, m) = \prod_{i=1}^{m} \frac{1}{1+n} \, p(e_i \mid f_{a_i})$$

$$p(e_i, a_i \mid \mathbf{f}, m) = \frac{1}{1+n} p(e_i \mid f_{a_i})$$

$$p(\mathbf{e}, \mathbf{a} \mid \mathbf{f}, m) = \prod_{i=1}^{m} p(e_i, a_i \mid \mathbf{f}, m)$$

Marginal probability

$$p(e_i, a_i \mid \mathbf{f}, m) = \frac{1}{1+n} p(e_i \mid f_{a_i})$$

$$p(e_i \mid \mathbf{f}, m) = \sum_{a_i=0}^{n} \frac{1}{1+n} p(e_i \mid f_{a_i})$$

Recall our independence assumption: all alignment decisions are independent of each other, and given the alignments then all translation decisions are independent of each other, so **all translation decisions are independent of each other**.

$$p(a, b, c, d) = p(a)p(b)p(c)p(d)$$

$$p(\mathbf{e} \mid \mathbf{f}, m) = \prod_{i=1}^{m} p(e_i \mid \mathbf{f}, m)$$

Marginal probability

$$p(e_i, a_i \mid \mathbf{f}, m) = \frac{1}{1+n} p(e_i \mid f_{a_i})$$

$$p(e_i \mid \mathbf{f}, m) = \sum_{a_i=0}^{n} \frac{1}{1+n} p(e_i \mid f_{a_i})$$

$$p(\mathbf{e} \mid \mathbf{f}, m) = \prod_{i=1}^{m} p(e_i \mid \mathbf{f}, m)$$

Marginal probability

$$p(e_i, a_i \mid \mathbf{f}, m) = \frac{1}{1+n} p(e_i \mid f_{a_i})$$

$$p(e_i \mid \mathbf{f}, m) = \sum_{a_i=0}^{n} \frac{1}{1+n} p(e_i \mid f_{a_i})$$

$$\begin{aligned} p(\mathbf{e} \mid \mathbf{f}, m) &= \prod_{i=1}^{m} p(e_i \mid \mathbf{f}, m) \\ &= \prod_{i=1}^{m} \sum_{a_i=0}^{n} \frac{1}{1+n} p(e_i \mid f_{a_i}) \end{aligned}$$

Marginal probability

$$p(e_i, a_i \mid \mathbf{f}, m) = \frac{1}{1+n} p(e_i \mid f_{a_i})$$

$$p(e_i \mid \mathbf{f}, m) = \sum_{a_i=0}^{n} \frac{1}{1+n} p(e_i \mid f_{a_i})$$

$$\begin{aligned} p(\mathbf{e} \mid \mathbf{f}, m) &= \prod_{i=1}^{m} p(e_i \mid \mathbf{f}, m) \\ &= \prod_{i=1}^{m} \sum_{a_i=0}^{n} \frac{1}{1+n} p(e_i \mid f_{a_i}) \\ &= \frac{1}{(1+n)^m} \prod_{i=1}^{m} \sum_{a_i=0}^{n} p(e_i \mid f_{a_i}) \end{aligned}$$

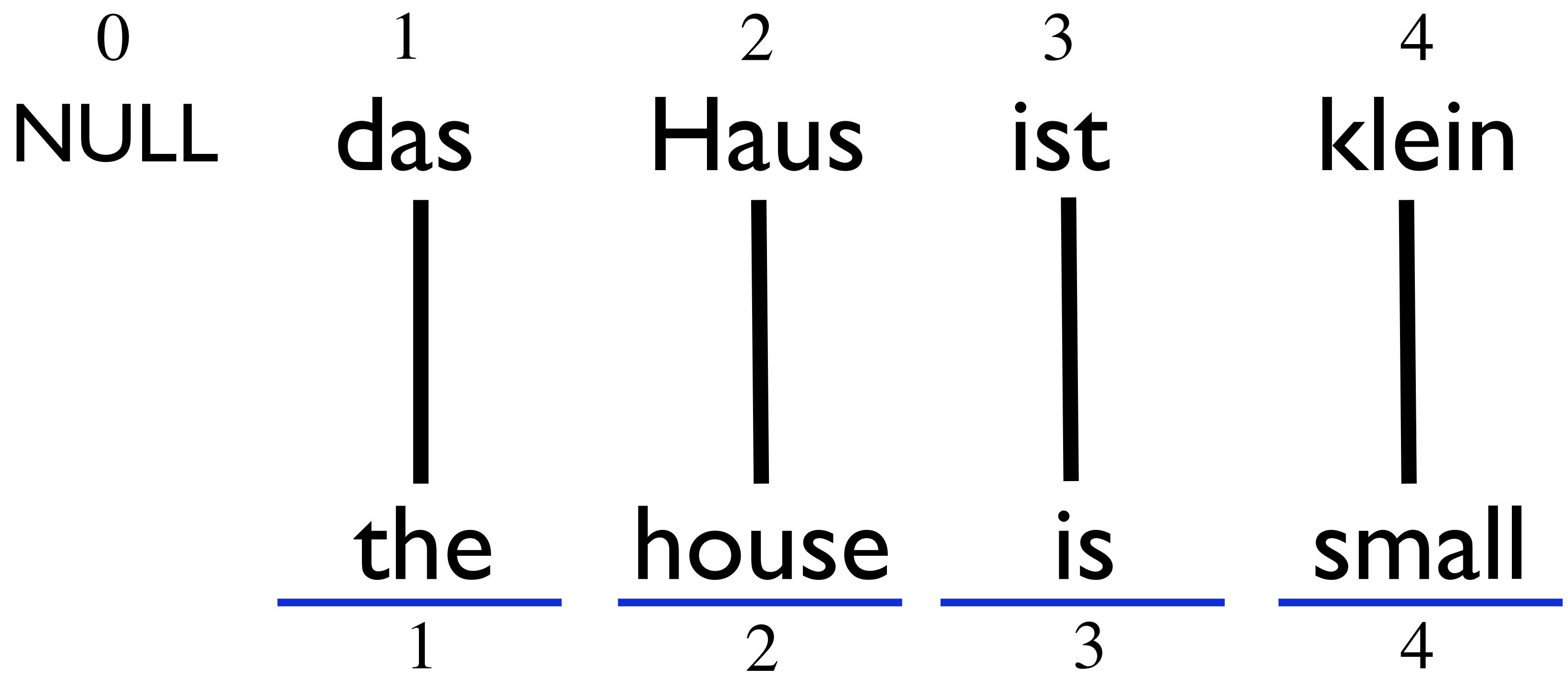
Example

0	1	2	3	4
NULL	das	Haus	ist	klein

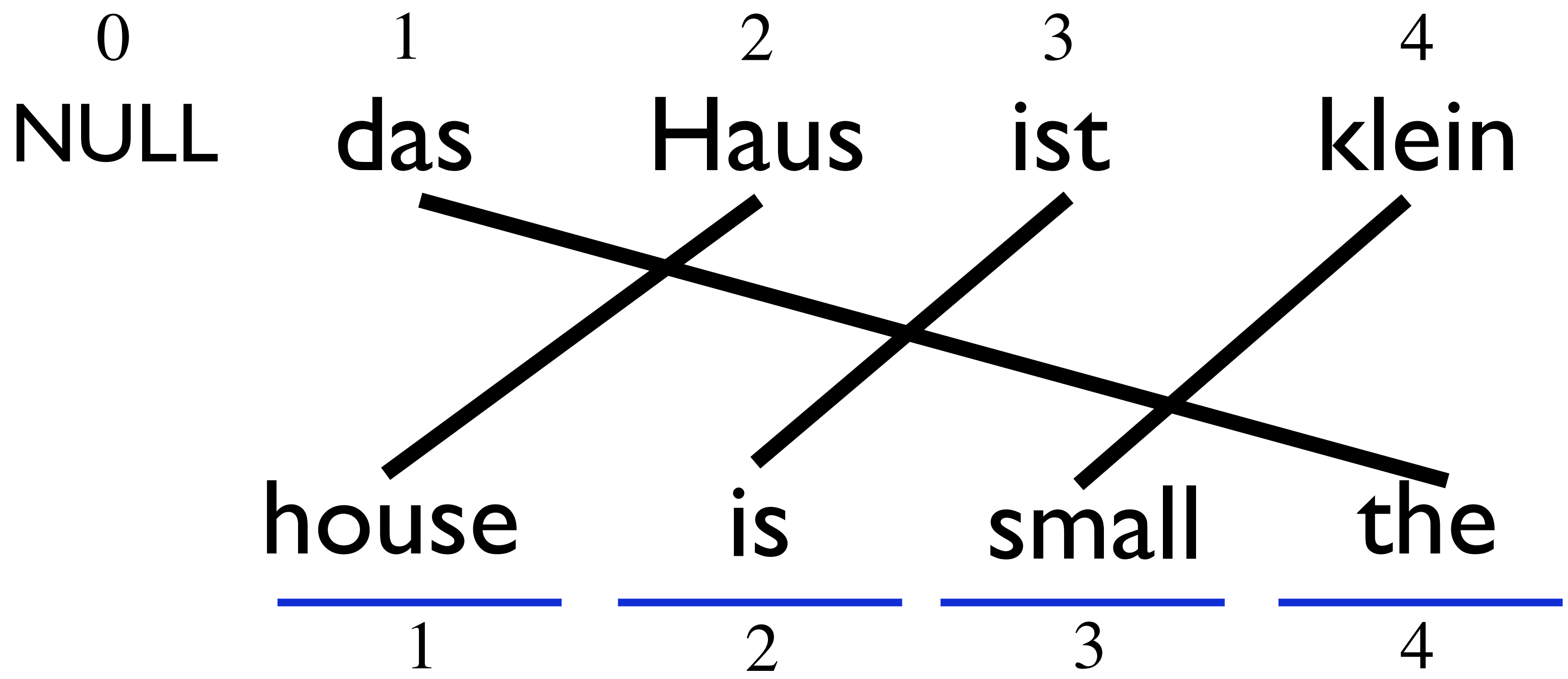
_____	_____	_____	_____
1	2	3	4

Start with a foreign sentence and a target length.

Example



Example

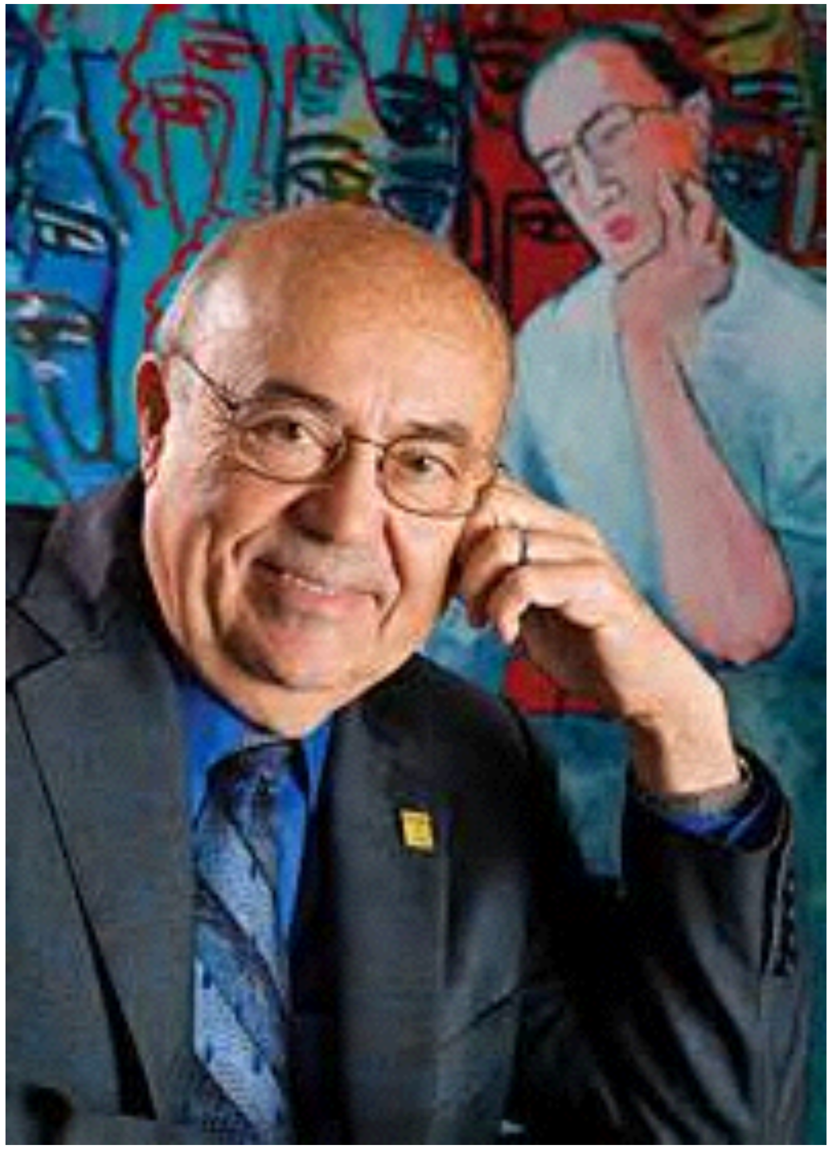


Finding the Viterbi Alignment

$$\mathbf{a}^* = \arg \max_{\mathbf{a} \in [0,1,\ldots,n]^m} p(\mathbf{a} \mid \mathbf{e}, \mathbf{f})$$

Historical Note #2

The **Viterbi algorithm** is a dynamic programming algorithm for finding the most likely sequence of hidden states – called the **Viterbi path** – that results in a sequence of observed events, especially in the context of Markov information sources and hidden Markov models.



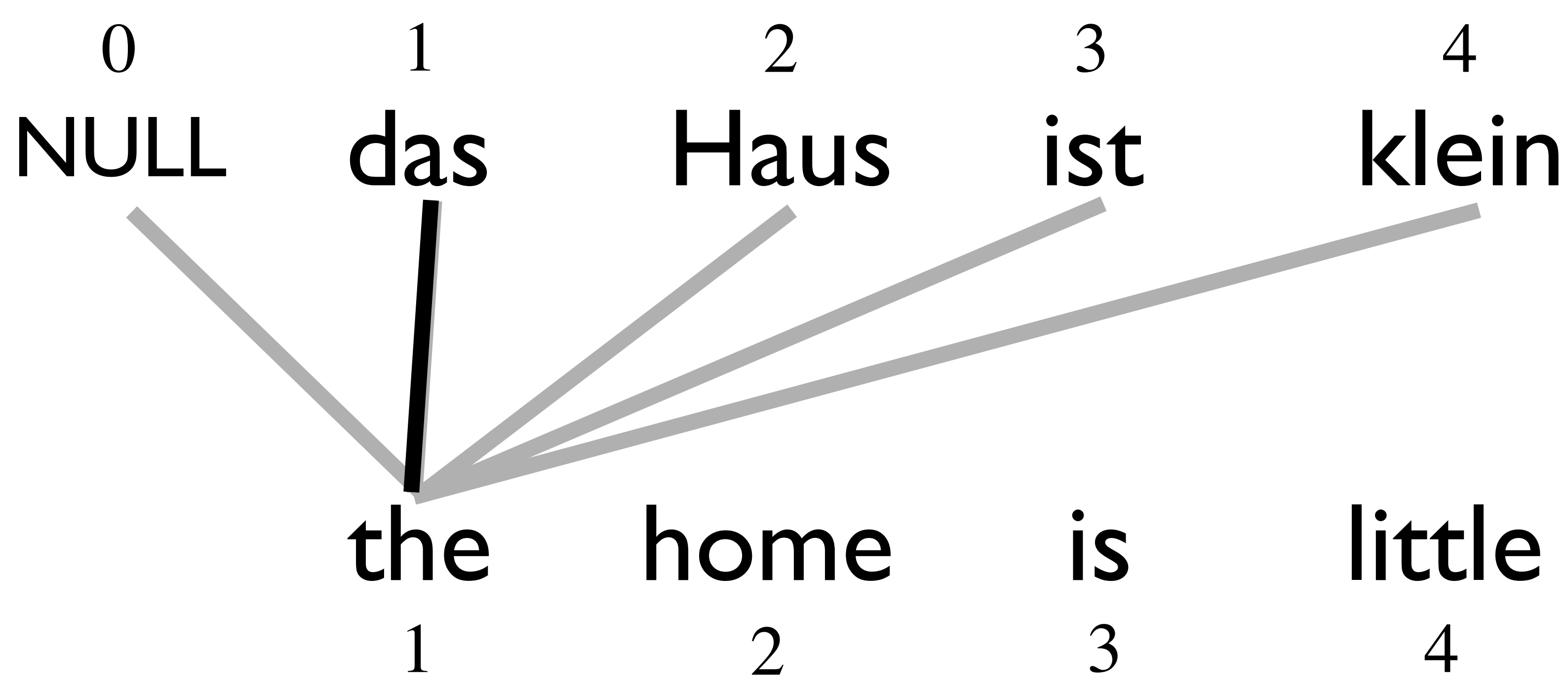
Andrew Viterbi

Professor at USC

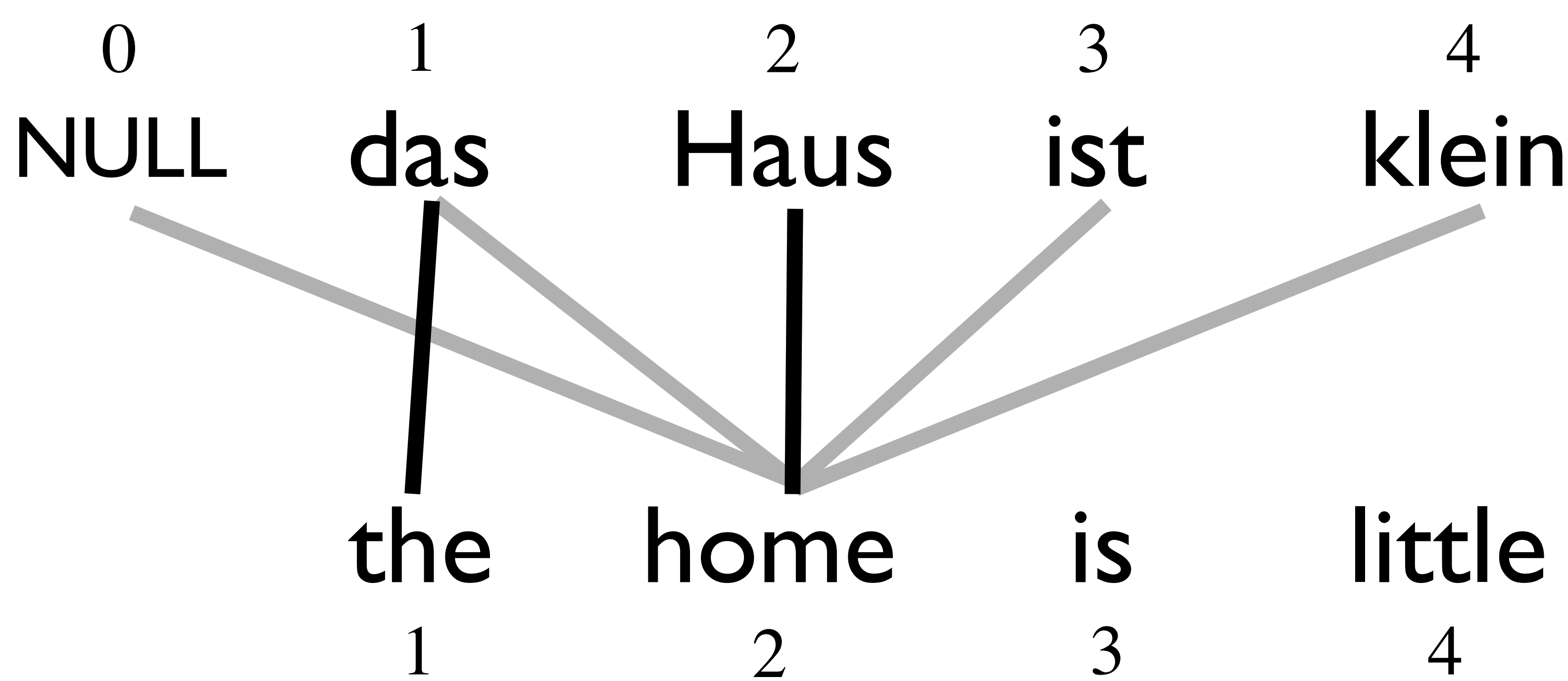
co-founder of Qualcomm

classmates with Fred Jelinek

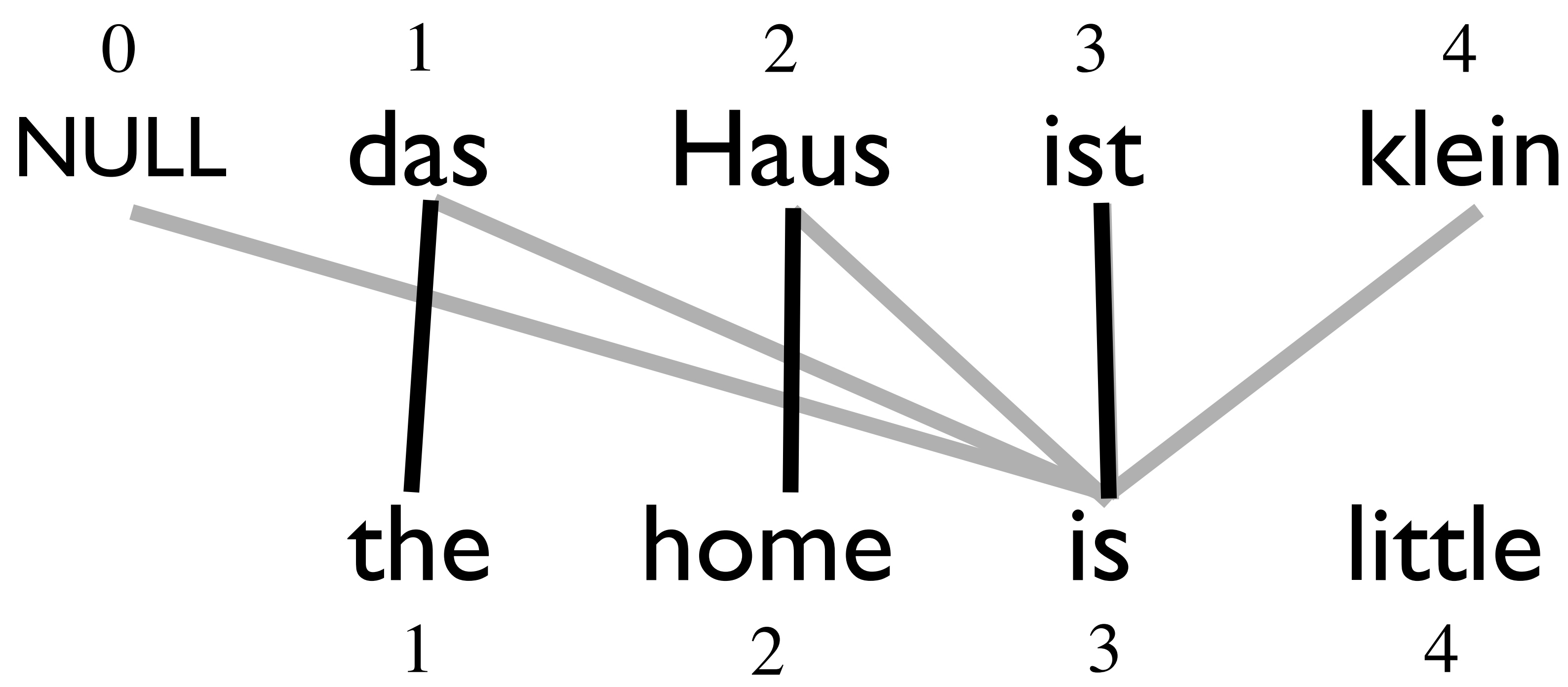
Finding the Viterbi Alignment



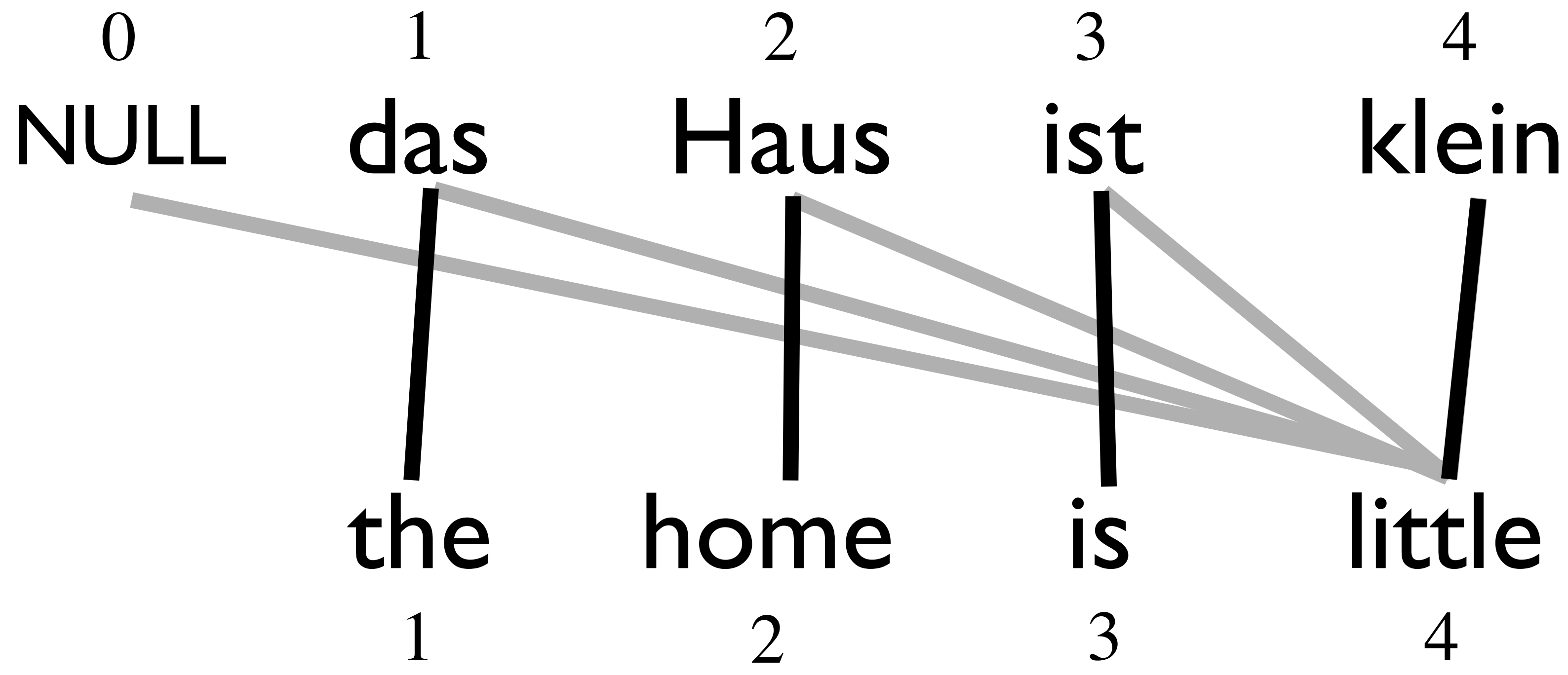
Finding the Viterbi Alignment



Finding the Viterbi Alignment

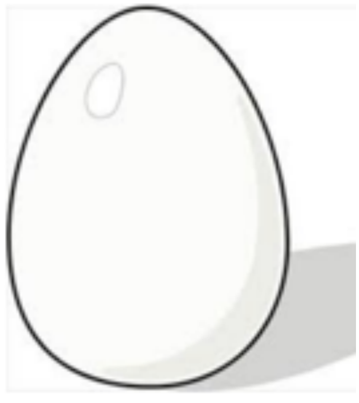


Finding the Viterbi Alignment



Learning Lexical Translation Models

- How do we learn the parameters $p(e \mid f)$
- "Chicken and egg" problem
  - If we had the alignments, we could estimate the parameters (MLE)
  - If we had parameters, we could find the most likely alignments



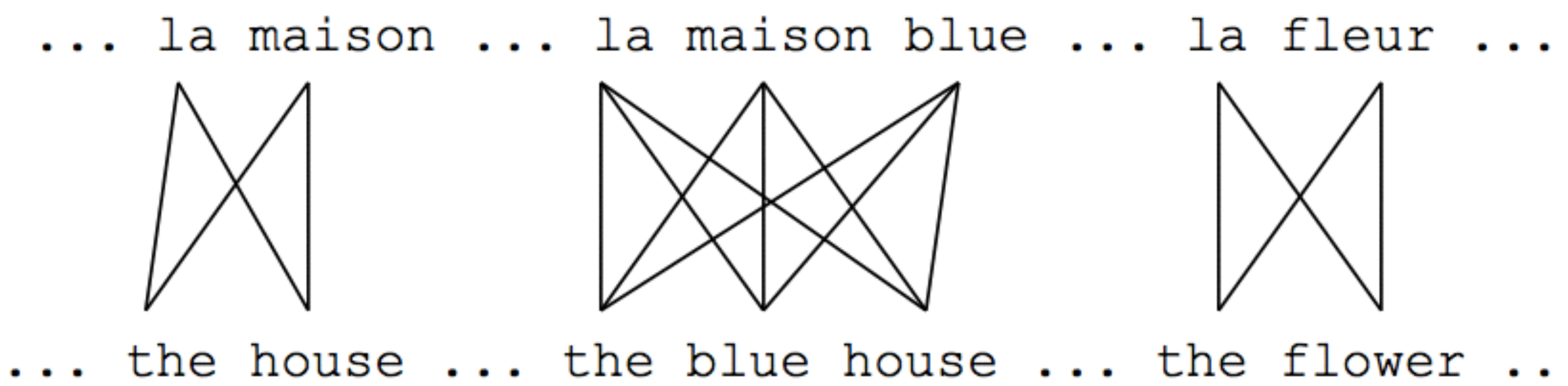
EM Algorithm

- pick some random (or uniform) parameters
- Repeat until you get bored (~ 5 iterations for lexical translation models)
    - using your current parameters, compute "expected" alignments for every target word token in the training data

      $$p(a_i \mid \mathbf{e}, \mathbf{f}) \qquad \text{(on board)}$$

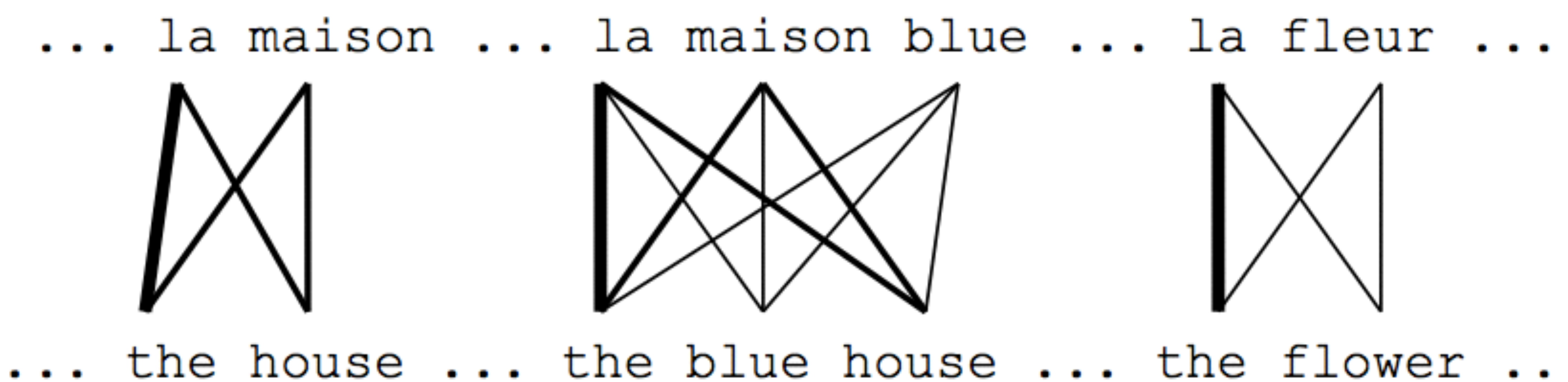
    - keep track of the expected number of times f translates into e throughout the whole corpus
    - keep track of the expected number of times that f is used as the source of any translation
    - use these expected counts as if they were "real" counts in the standard MLE equation

EM for Model 1



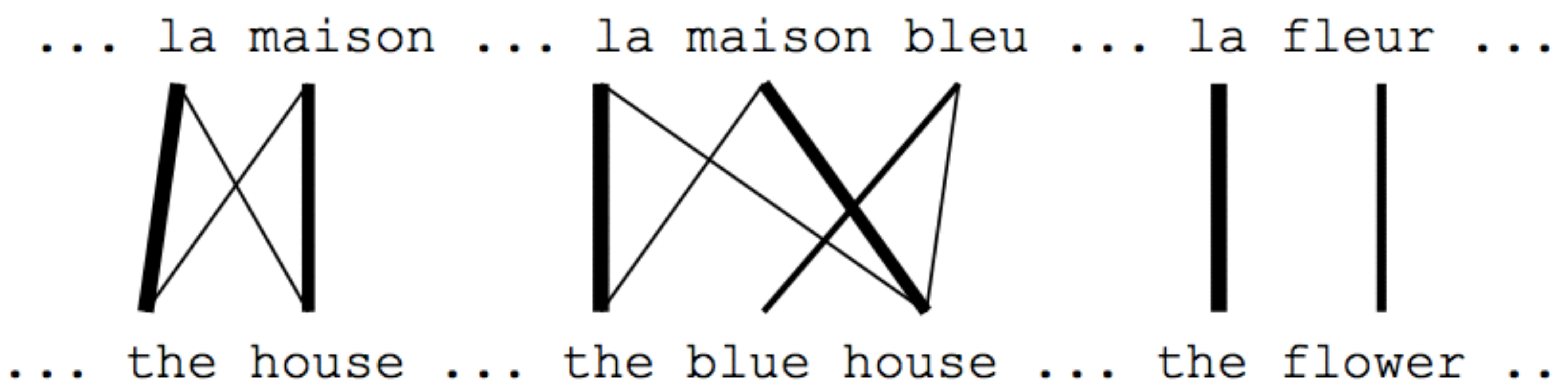
- Initial step: all alignments equally likely
- Model learns that, e.g., la is often aligned with the

EM for Model 1



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

EM for Model 1

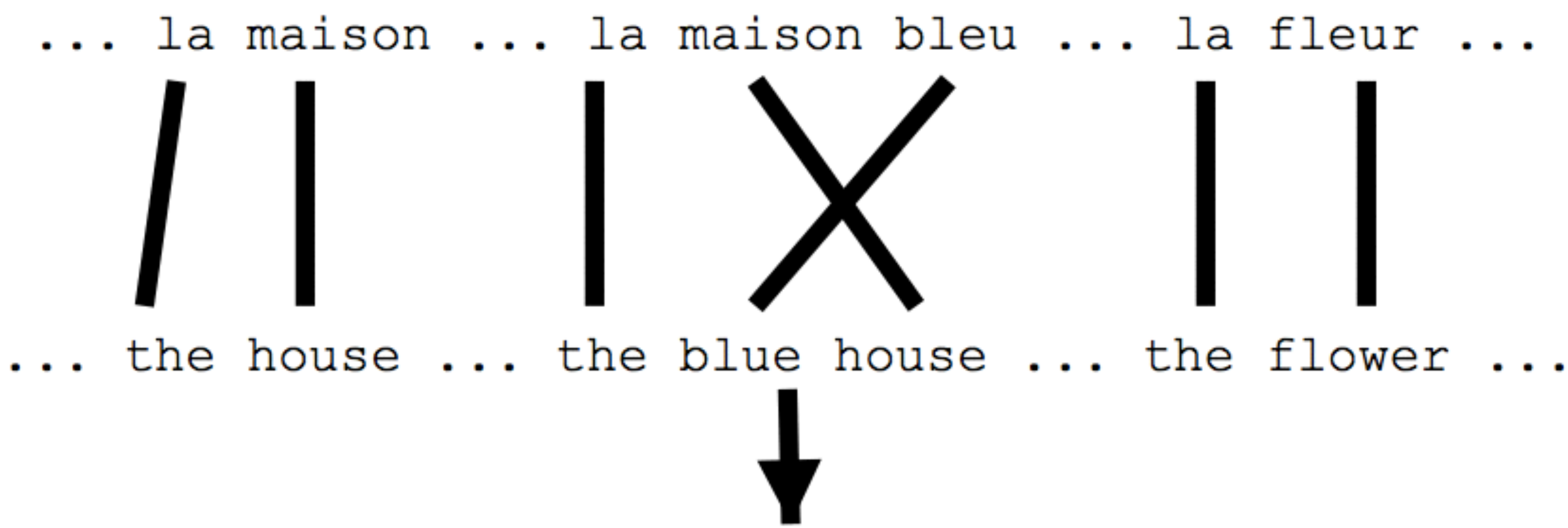


- After another iteration
- It becomes apparent that alignments, e.g., between fleur and flower are more likely (pigeon hole principle)

EM for Model 1

- Convergence
- Inherent hidden structure revealed by EM

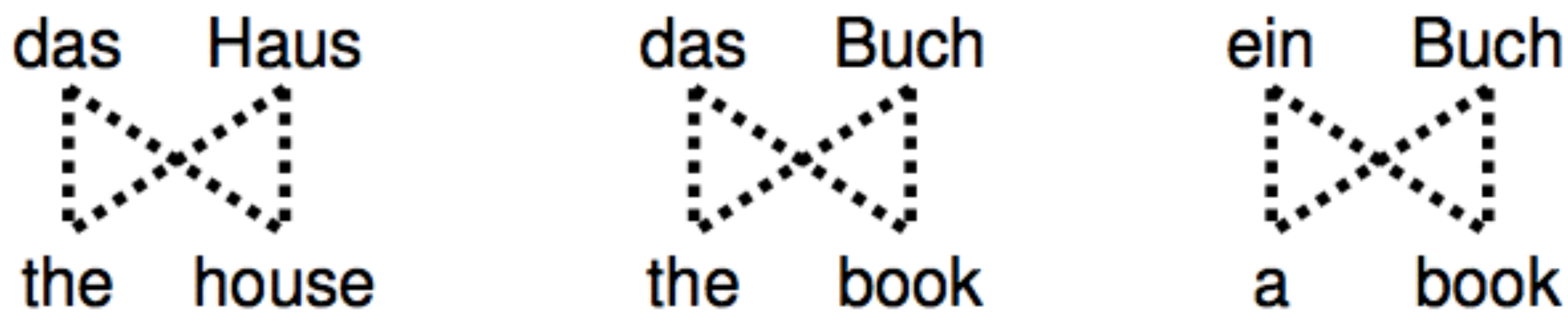
EM for Model 1




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

- Parameter estimation from the aligned corpus

Convergence



e	f	initial	1st it.	2nd it.	3rd it.	...	final
the	das	0.25	0.5	0.6364	0.7479	...	1
book	das	0.25	0.25	0.1818	0.1208	...	0
house	das	0.25	0.25	0.1818	0.1313	...	0
the	buch	0.25	0.25	0.1818	0.1208	...	0
book	buch	0.25	0.5	0.6364	0.7479	...	1
a	buch	0.25	0.25	0.1818	0.1313	...	0
book	ein	0.25	0.5	0.4286	0.3466	...	0
a	ein	0.25	0.5	0.5714	0.6534	...	1
the	haus	0.25	0.5	0.4286	0.3466	...	0
house	haus	0.25	0.5	0.5714	0.6534	...	1

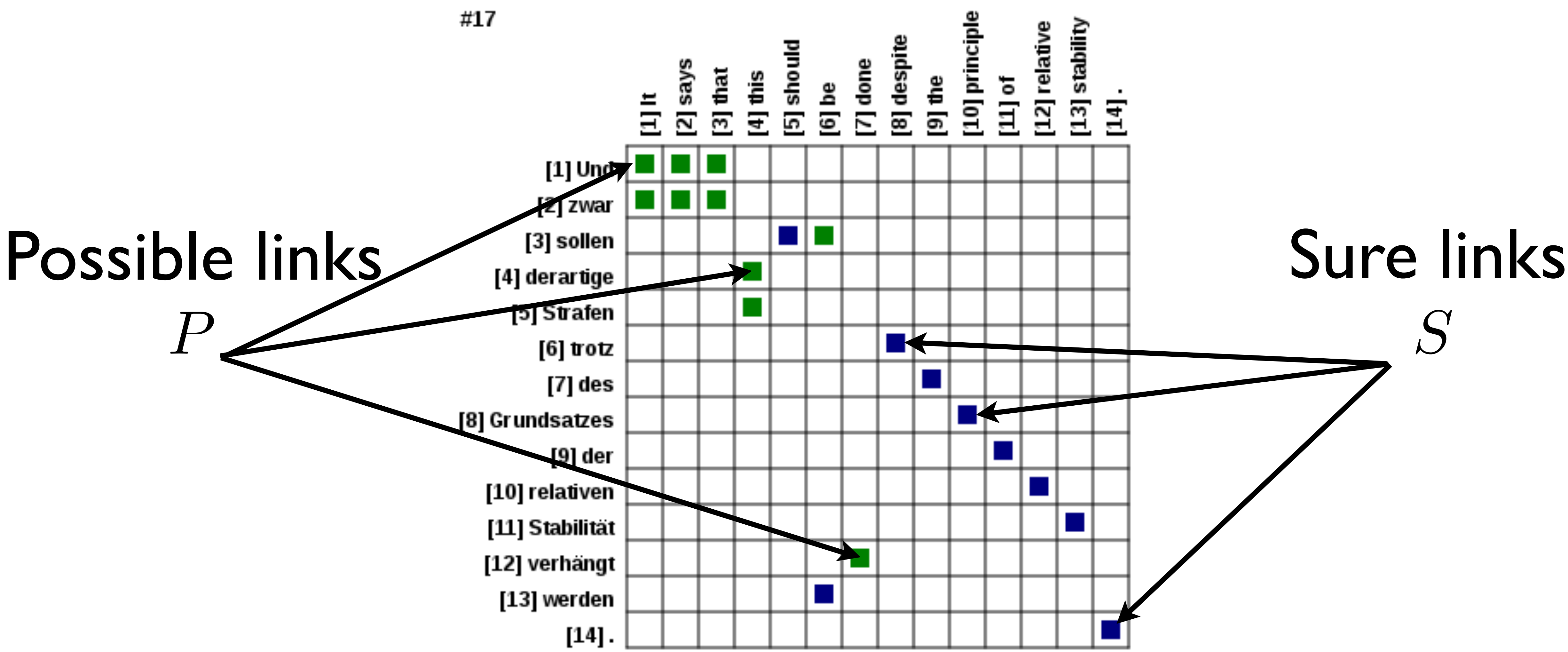
Evaluation

- Since we have a probabilistic model, we can evaluate **perplexity**.

$$\mathrm{PPL} = 2^{-\frac{1}{\sum_{(\mathbf{e},\mathbf{f})\in\mathcal{D}} |\mathbf{e}|} \log \prod_{(\mathbf{e},\mathbf{f})\in\mathcal{D}} p(\mathbf{e}|\mathbf{f})}$$

	Iter 1	Iter 2	Iter 3	Iter 4	...	Iter
-log likelihood	-	7.66	7.21	6.84	...	-6
perplexity	-	2.42	2.3	2.21	...	2

Alignment Error Rate

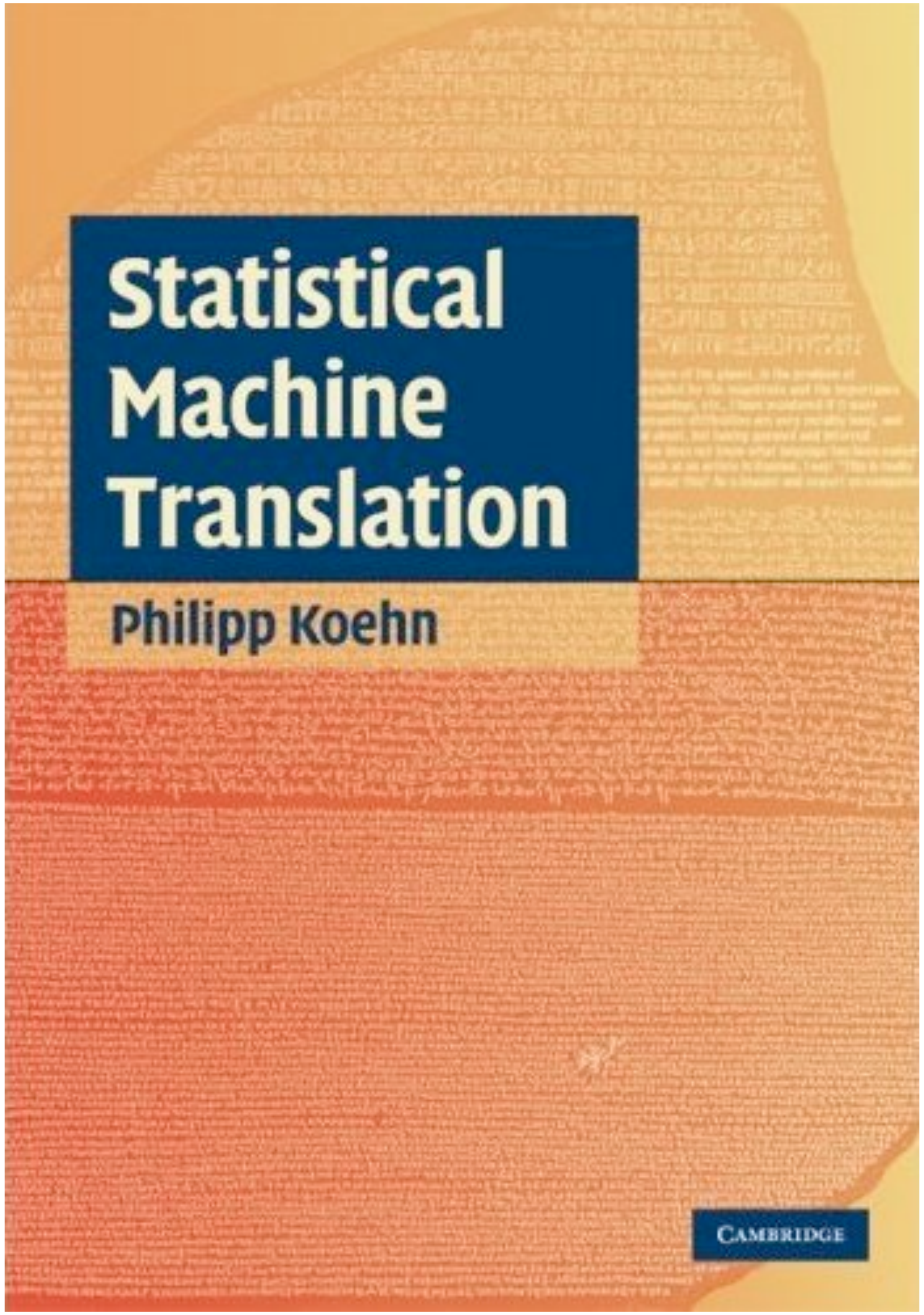


$$\text{Precision}(A, P) = \frac{|P \cap A|}{|A|} \qquad \text{Recall}(A, S) = \frac{|S \cap A|}{|S|}$$

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

Reading

- Read Chapter 4 from the textbook (today we covered 4.1 and 4.2)



Announcements

- First language-in-10 start next week
  - Thursday, Feb 6: Emily - Swedish
- HW 1 is due in 1 week