Lexical Translation Models 1

January 24, 2013

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

Lexical Translation

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

$$\mathbf{e} = \langle e_1, e_2, \ldots, e_m \rangle$$

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

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

Lexical Translation

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

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

Lexical Translation

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

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

Lexical Translation

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

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

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

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

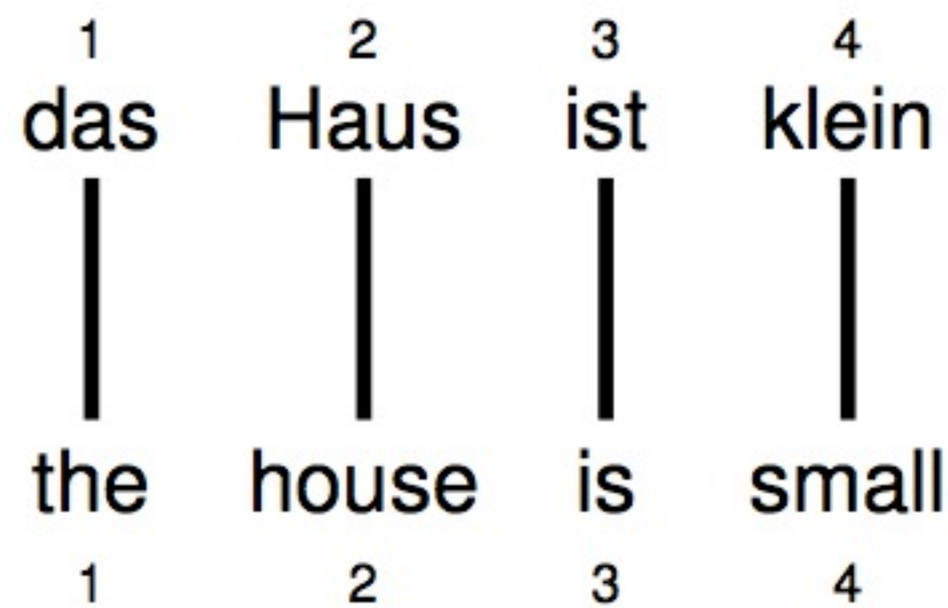
Alignment

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

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

Alignment

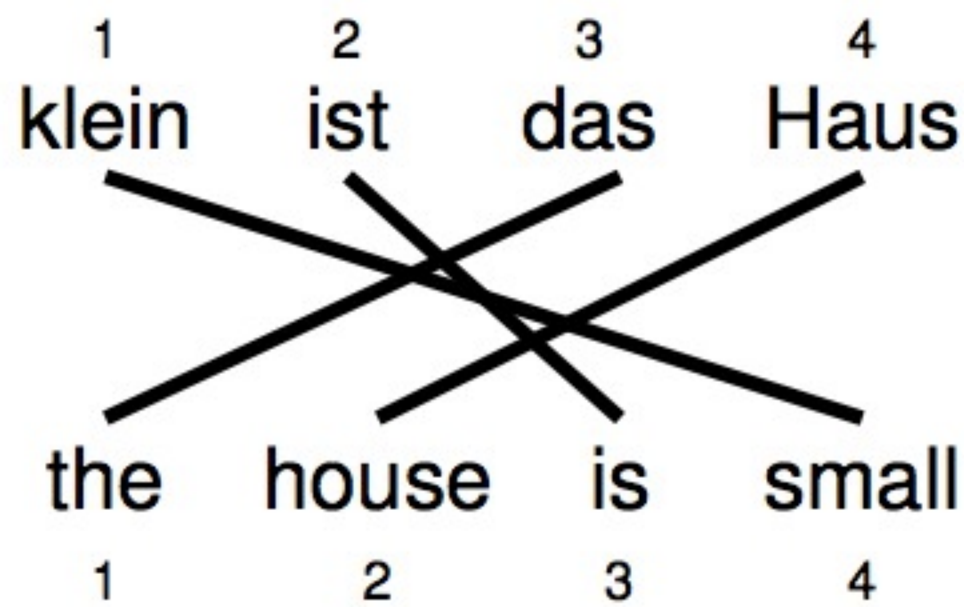
- Alignments can be visualized in by drawing links between two sentences, and they are represented as vectors of positions:



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

Reordering

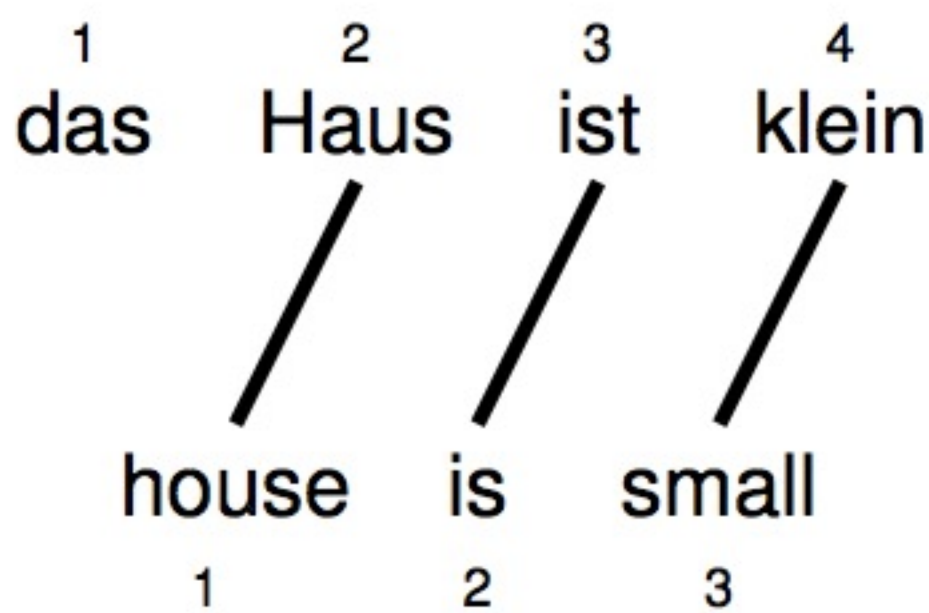
- Words may be reordered during translation.



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

Word Dropping

- A source word may not be translated at all



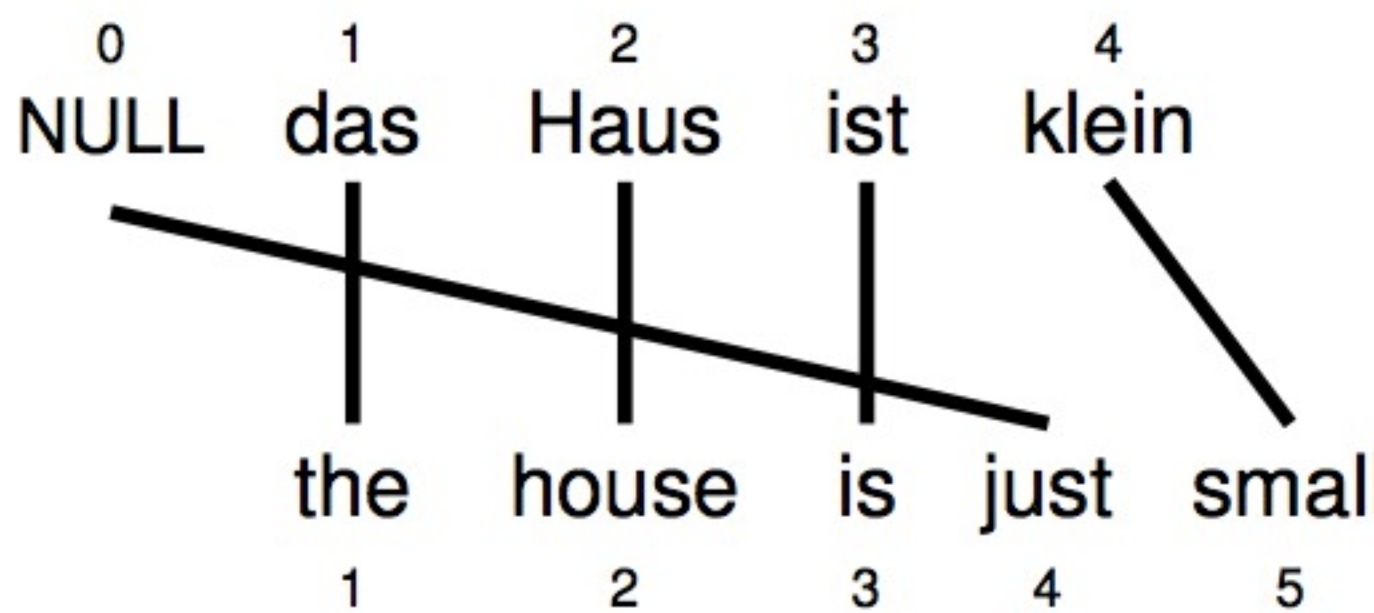
$$\mathbf{a} = (2, 3, 4)^\top$$

Word Insertion

- Words may be inserted during translation

English just does not have an equivalent

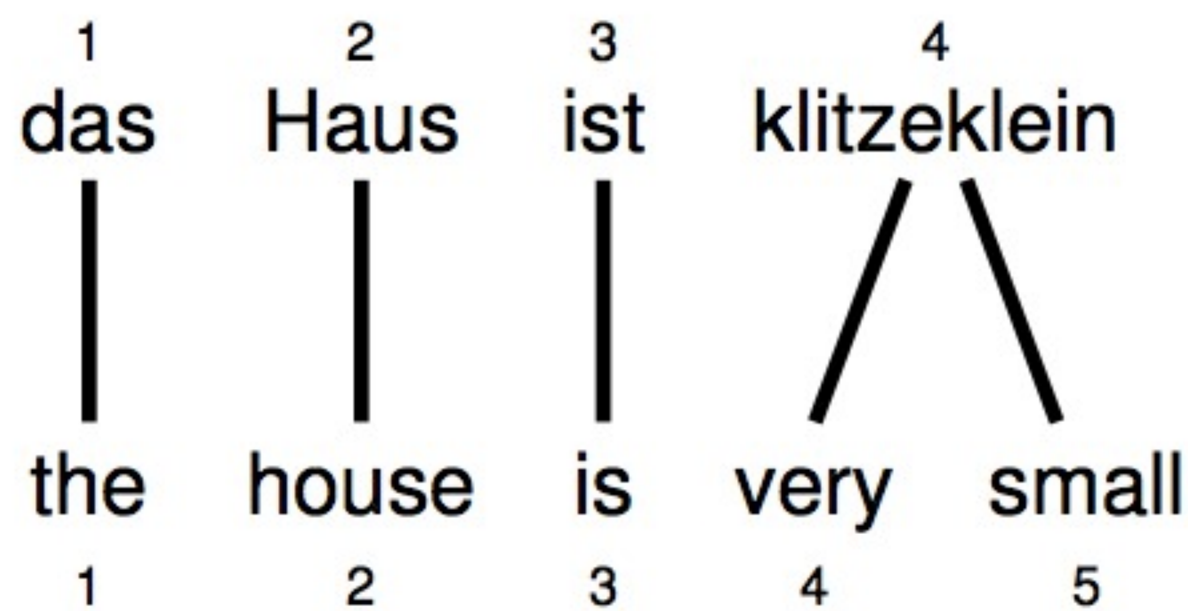
But it must be explained - we typically assume every source sentence contains a NULL token



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

One-to-many Translation

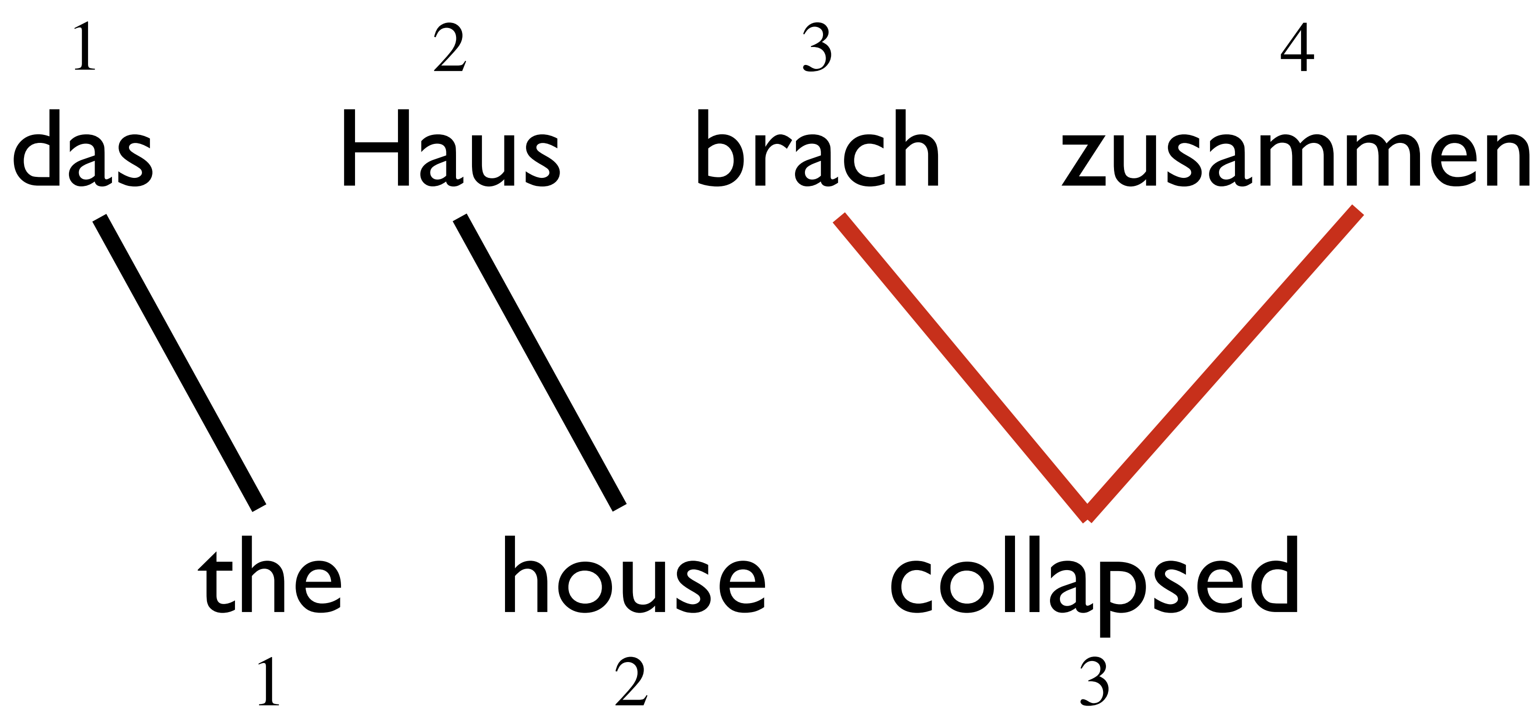
- A source word may translate into **more than one** target word



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

Many-to-one Translation

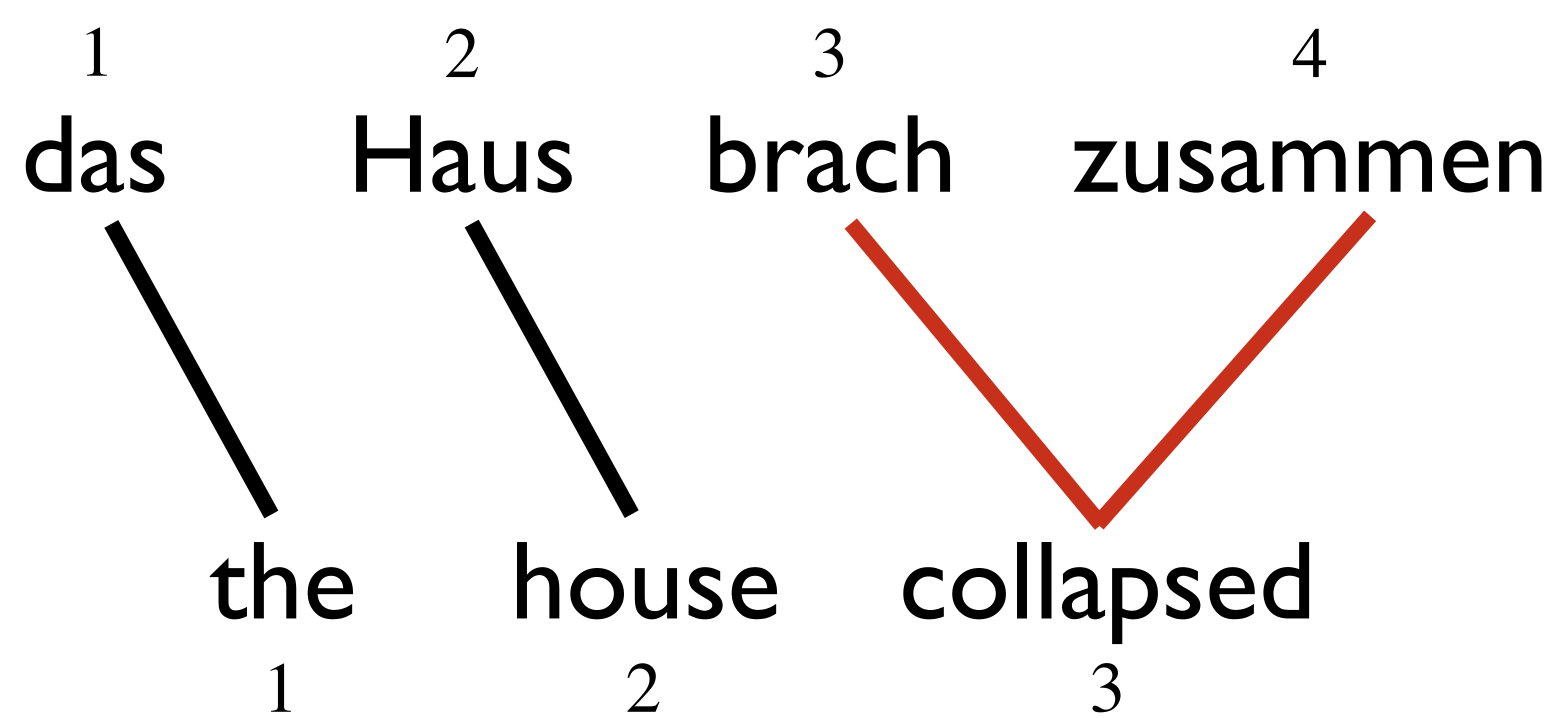
- **More than one source word** may **not** translate as a unit in lexical translation



$$\mathbf{a} = ???$$

Many-to-one Translation

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



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

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

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

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

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

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

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

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

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

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 alignments all translation decisions are independent of each other, so **all translation decisions are independent of each other**.

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 alignments all translation decisions are independent of each other, so **all translation decisions are independent of each other**.

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

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 alignments all translation decisions are independent of each other, so **all translation decisions are independent of each other**.

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 alignments all translation decisions are independent of each other, so **all translation decisions are independent of each other**.

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

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

0	1	2	3	4
NULL	das	Haus	ist	klein

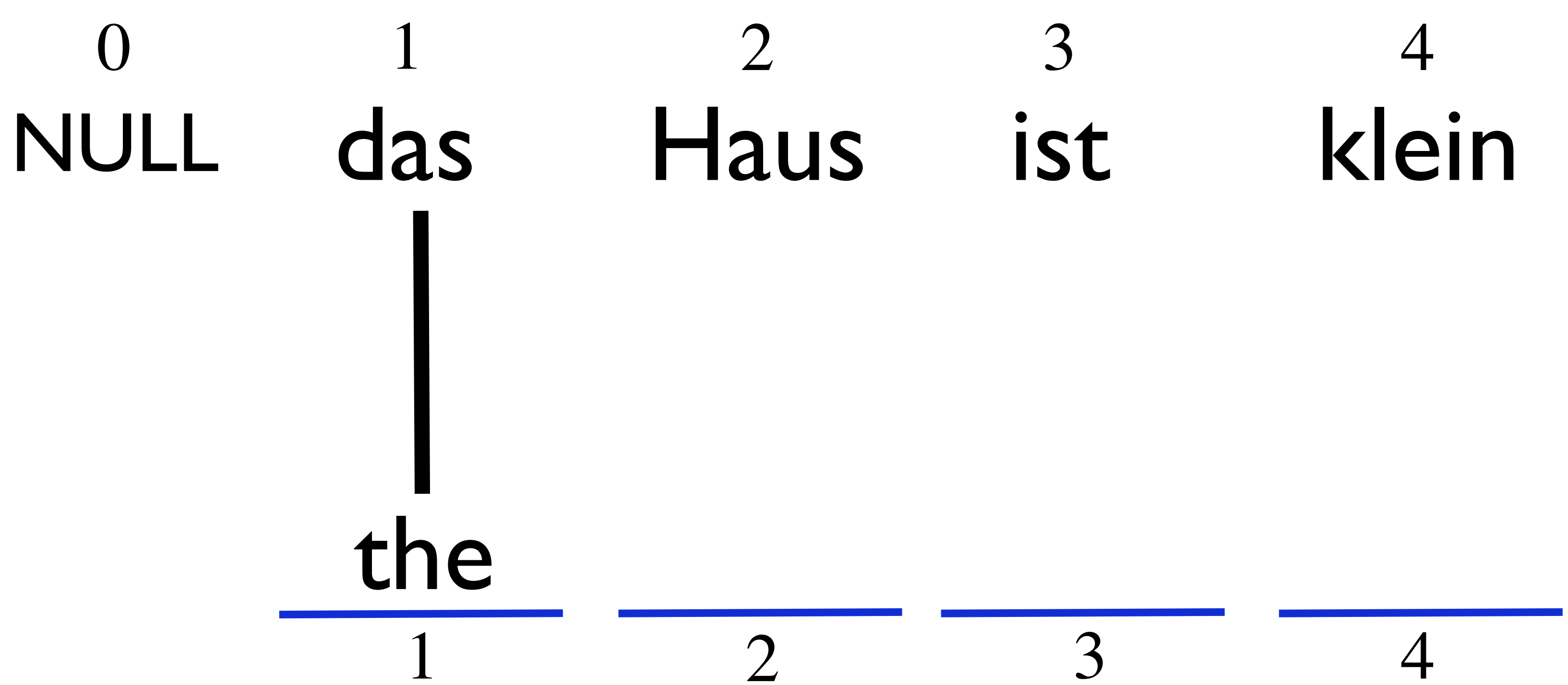
____	____	____	____
1	2	3	4

Example

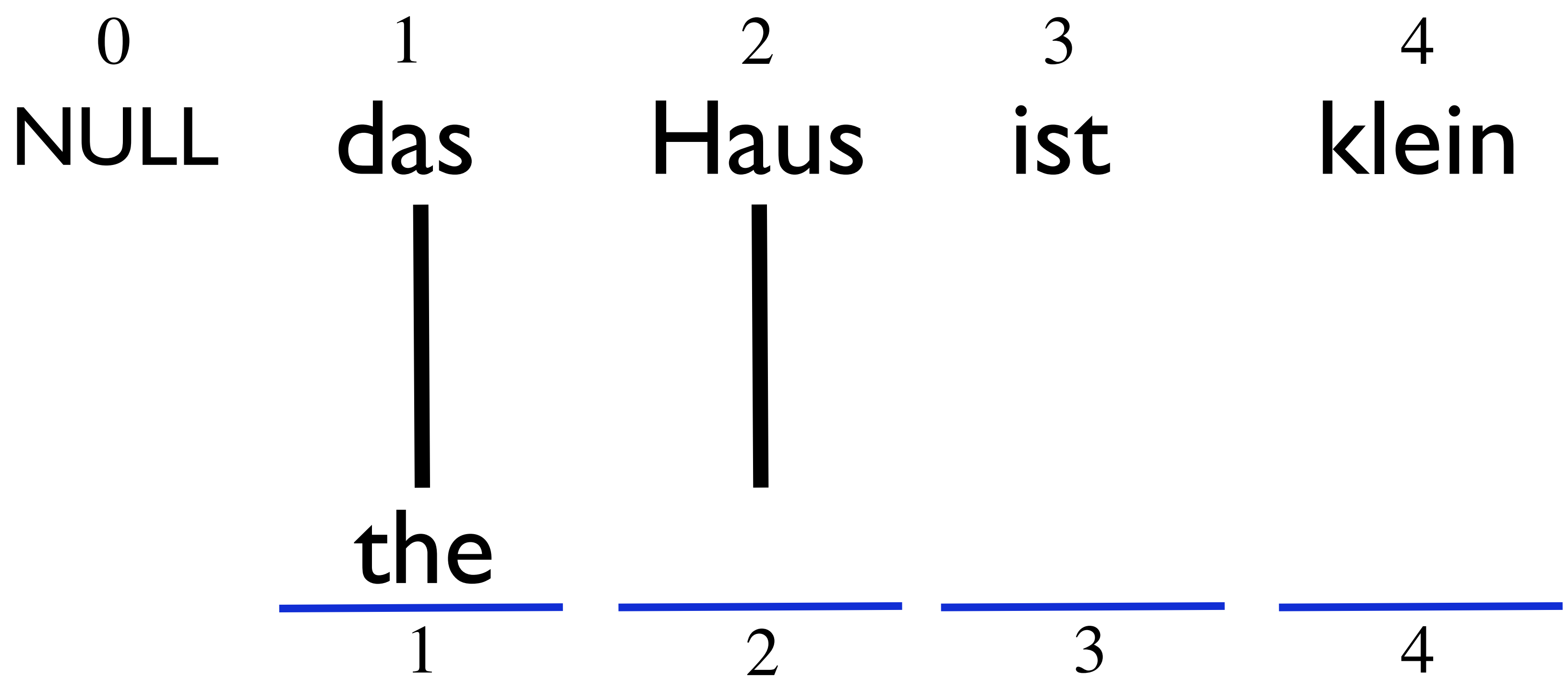
0	1	2	3	4
NULL	das	Haus	ist	klein

1	2	3	4

Example



Example



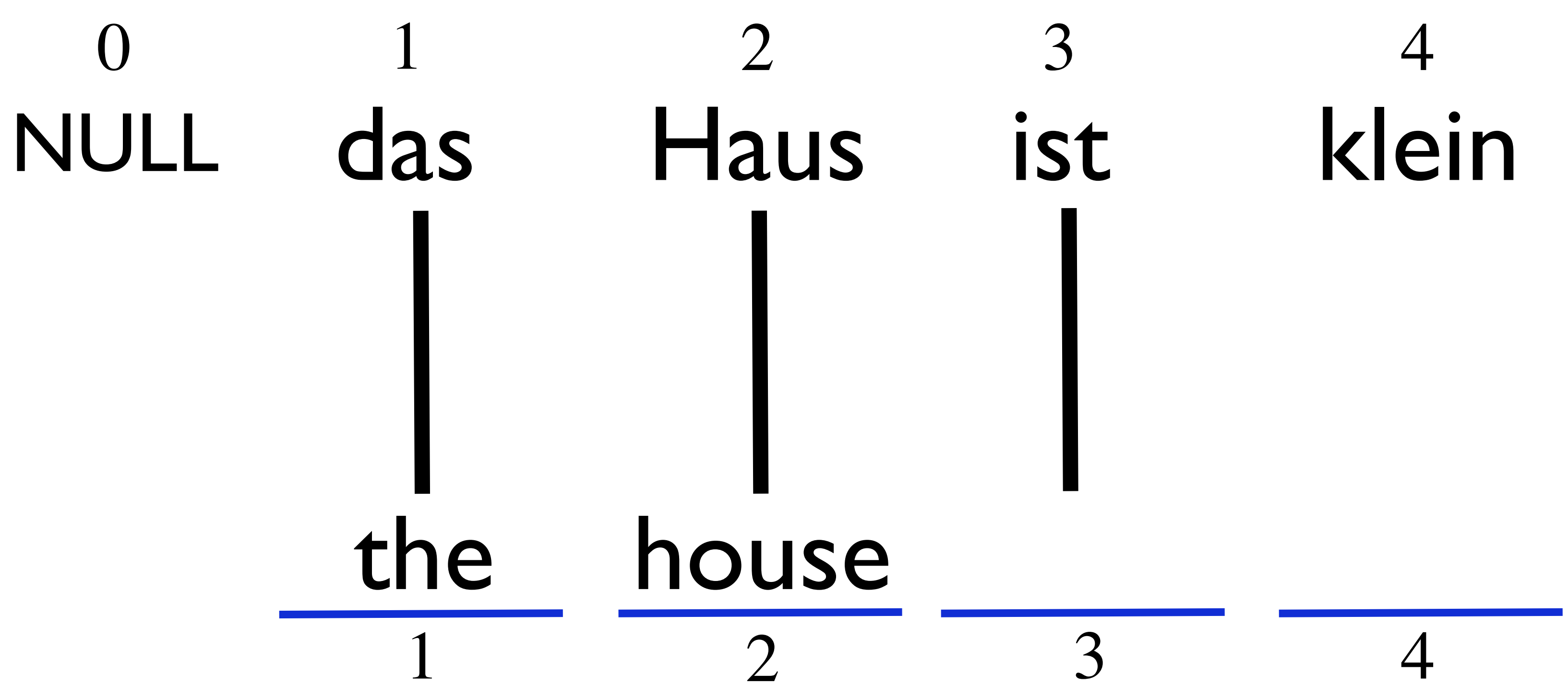
Example

0 NULL | 1 das | 2 Haus | 3 ist | 4 klein

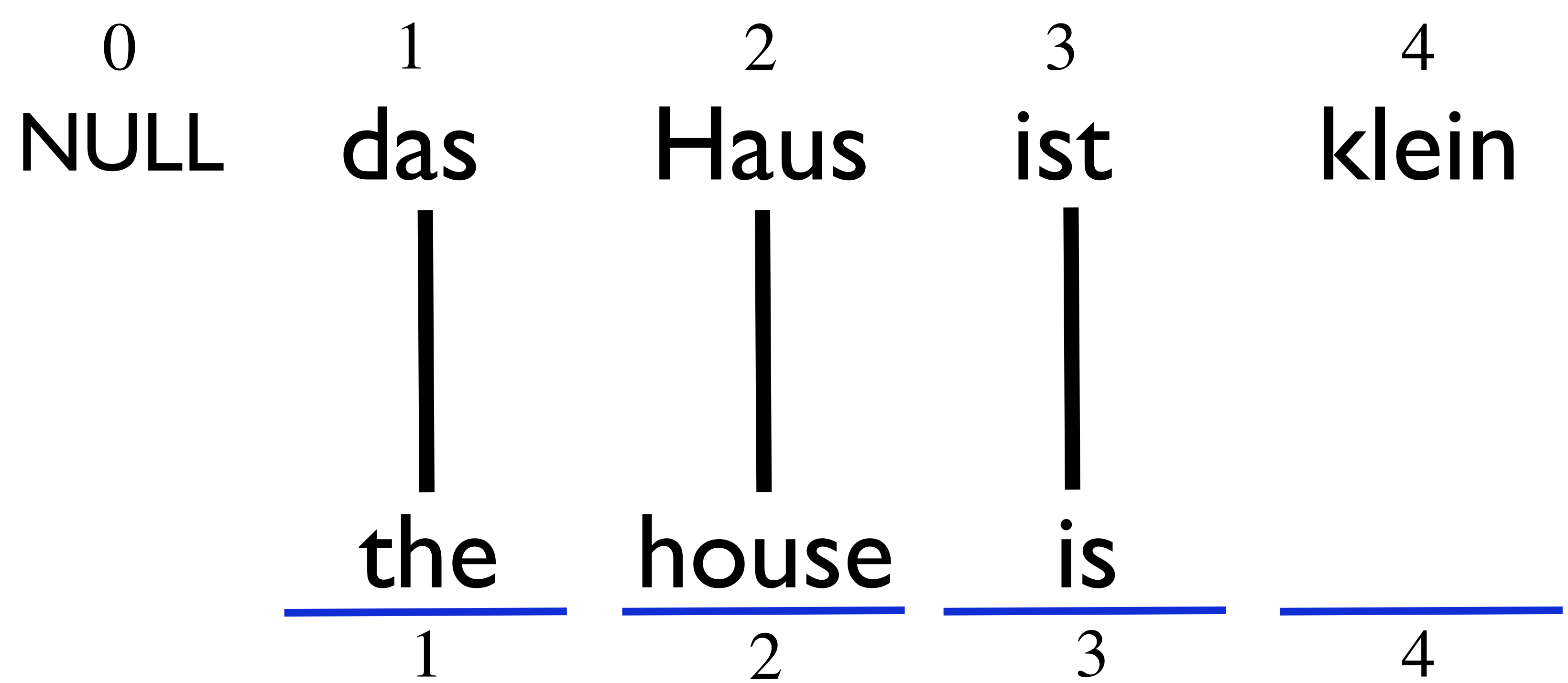
the | house | ___ | ___

1 | 2 | 3 | 4

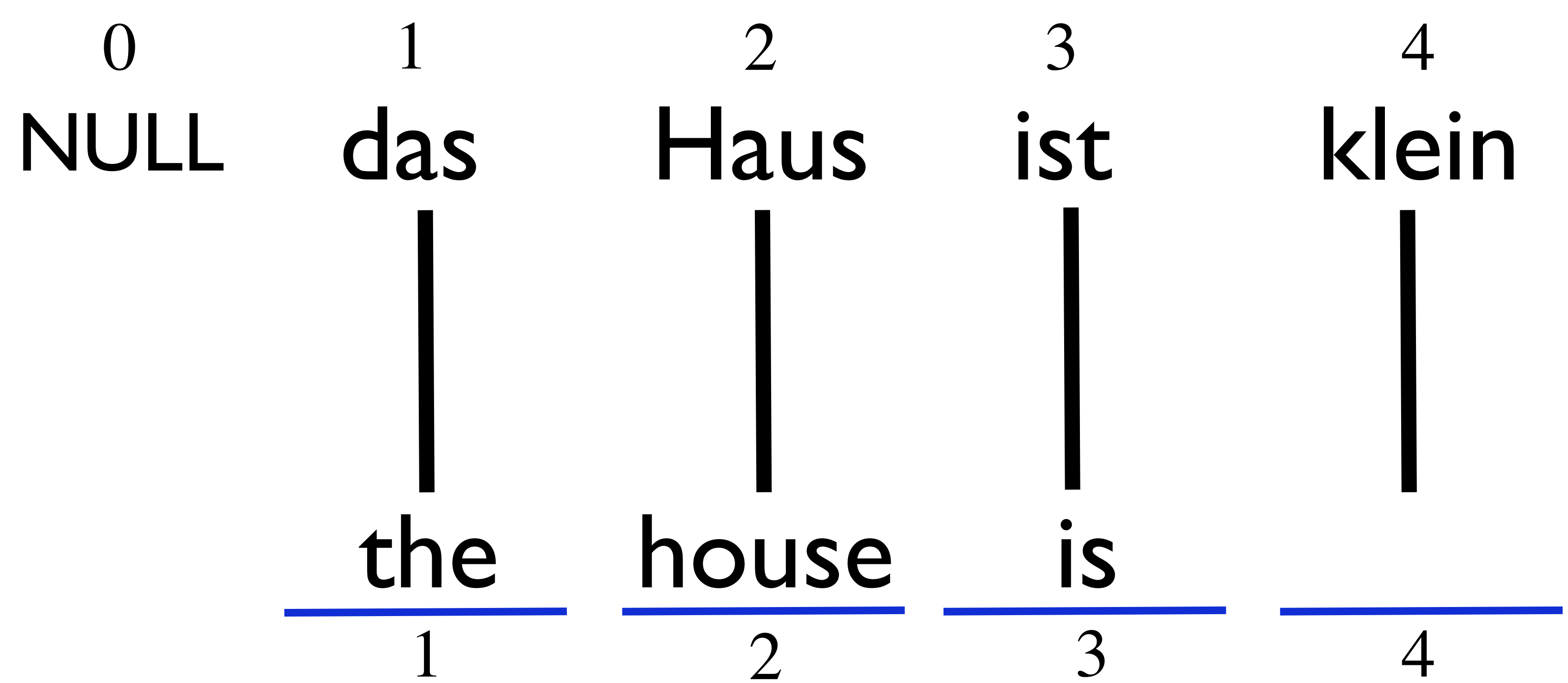
Example



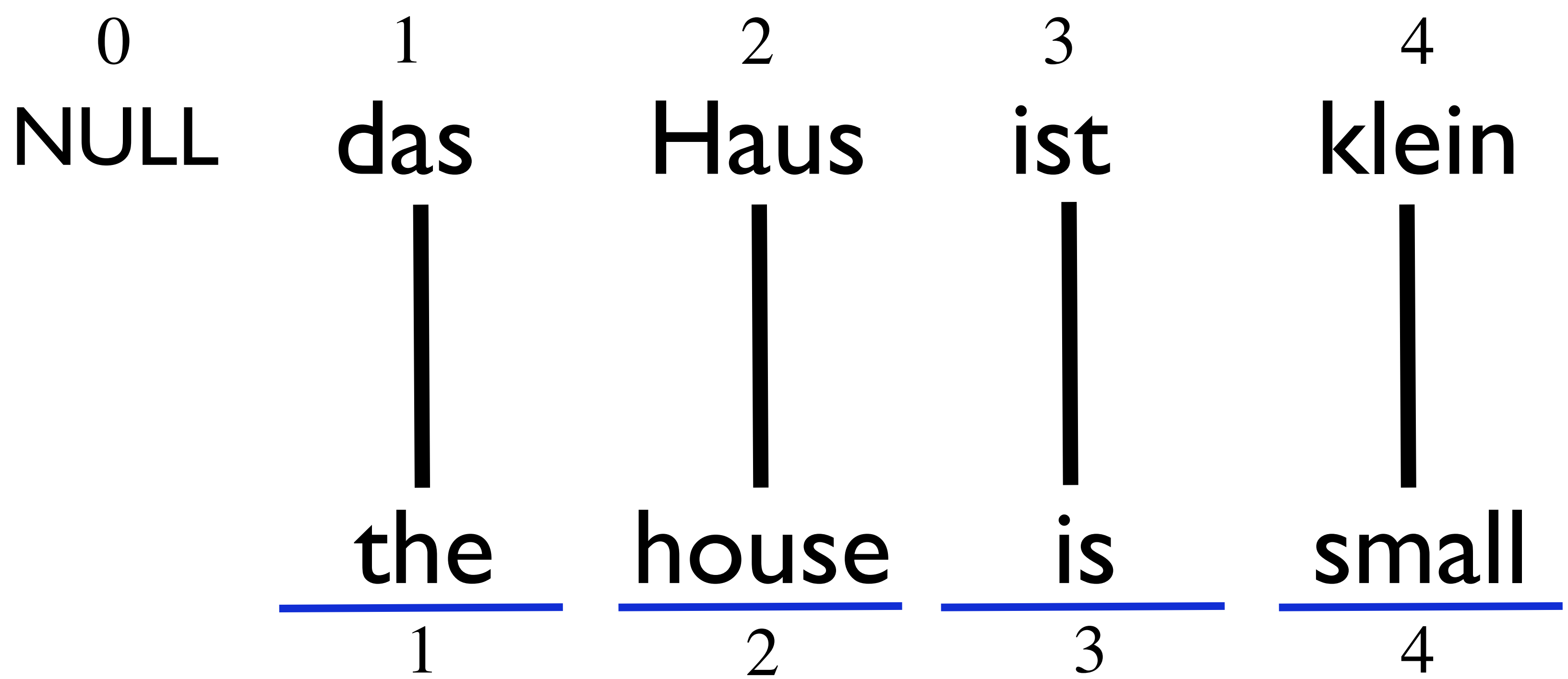
Example



Example



Example



Example

0	1	2	3	4
NULL	das	Haus	ist	klein

____ ____ ____ ____

1 2 3 4

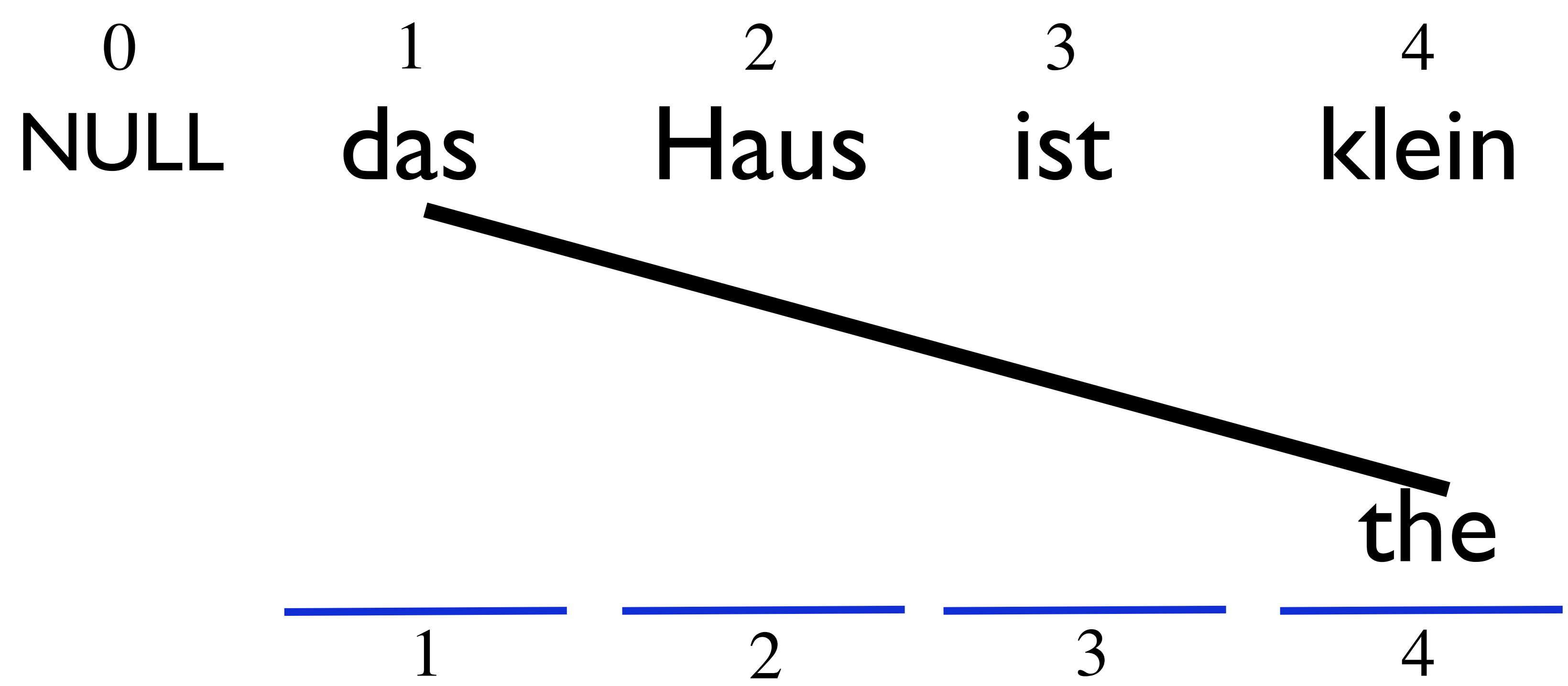
Example

0	1	2	3	4
NULL	das	Haus	ist	klein

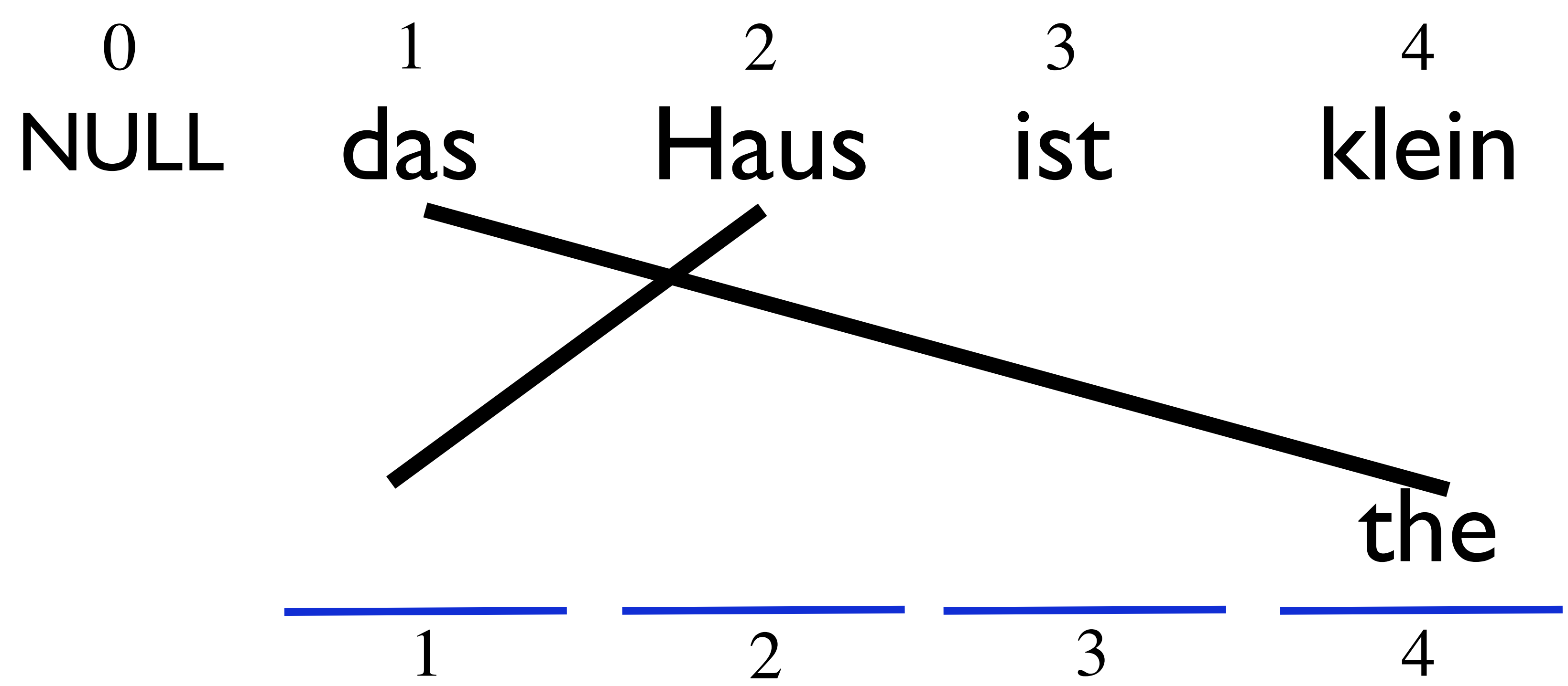
____ ____ ____ ____

1 2 3 4

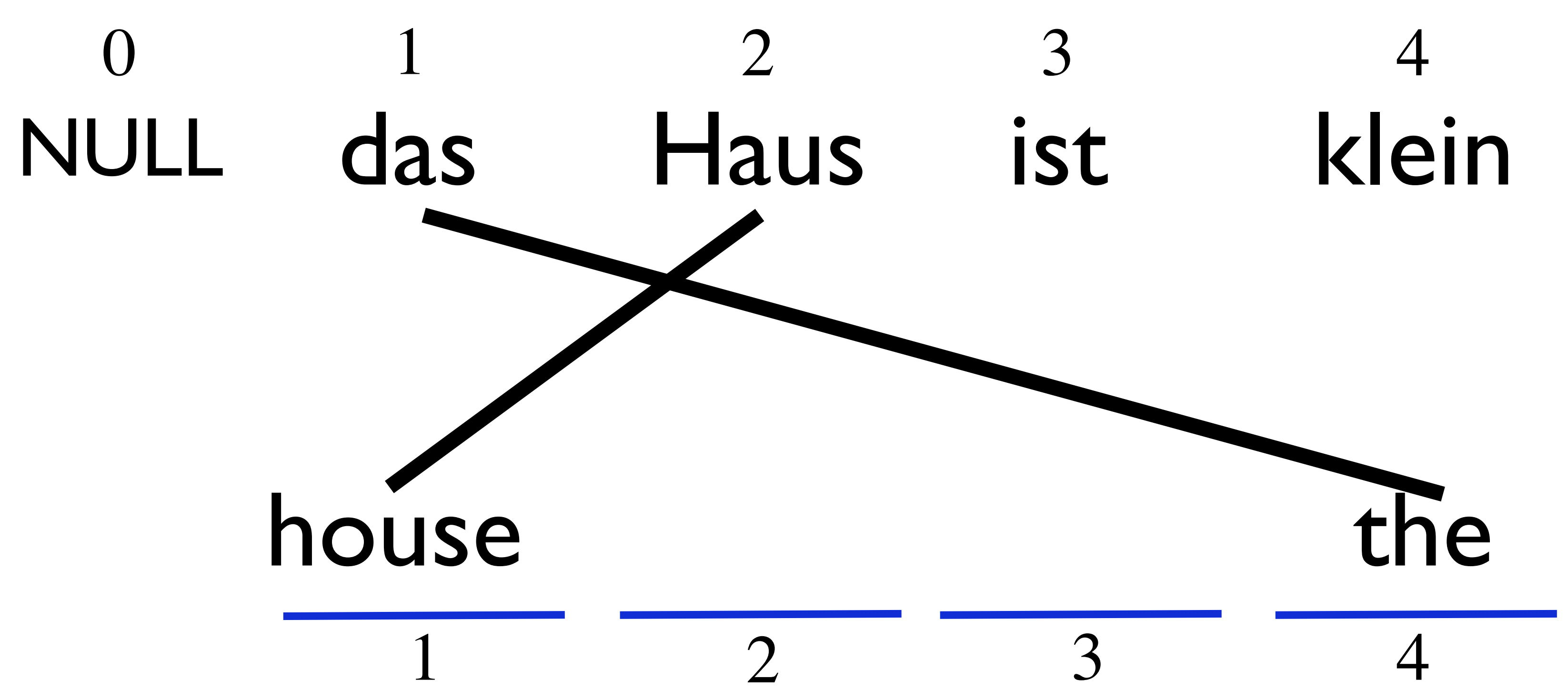
Example



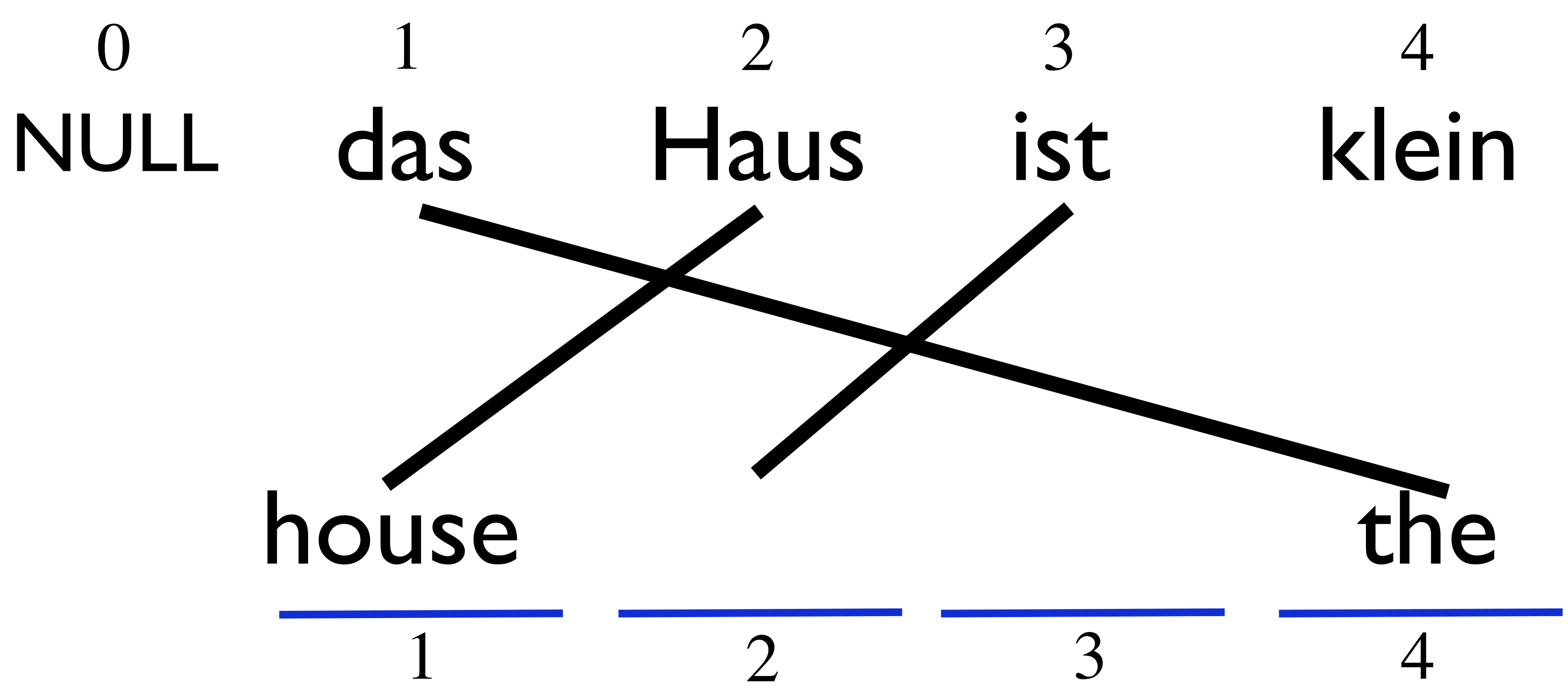
Example



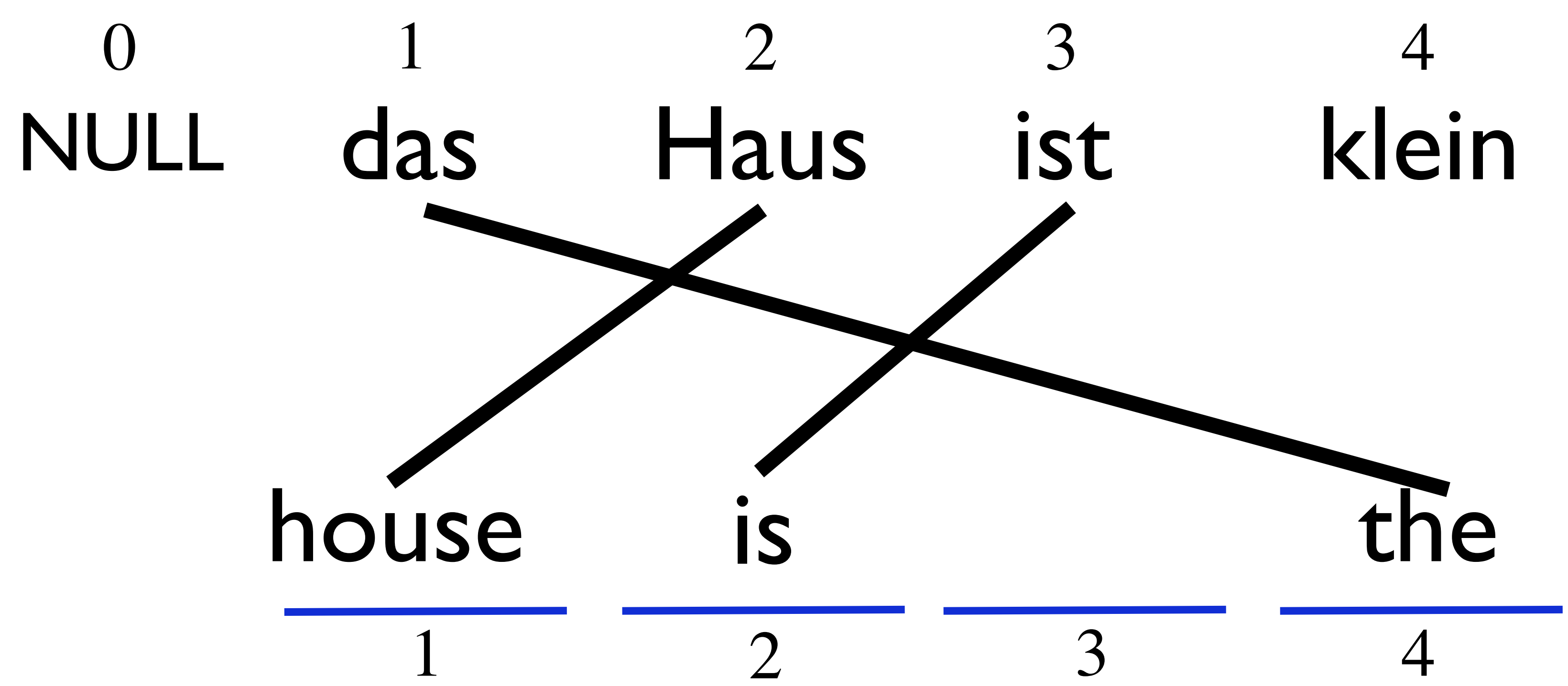
Example



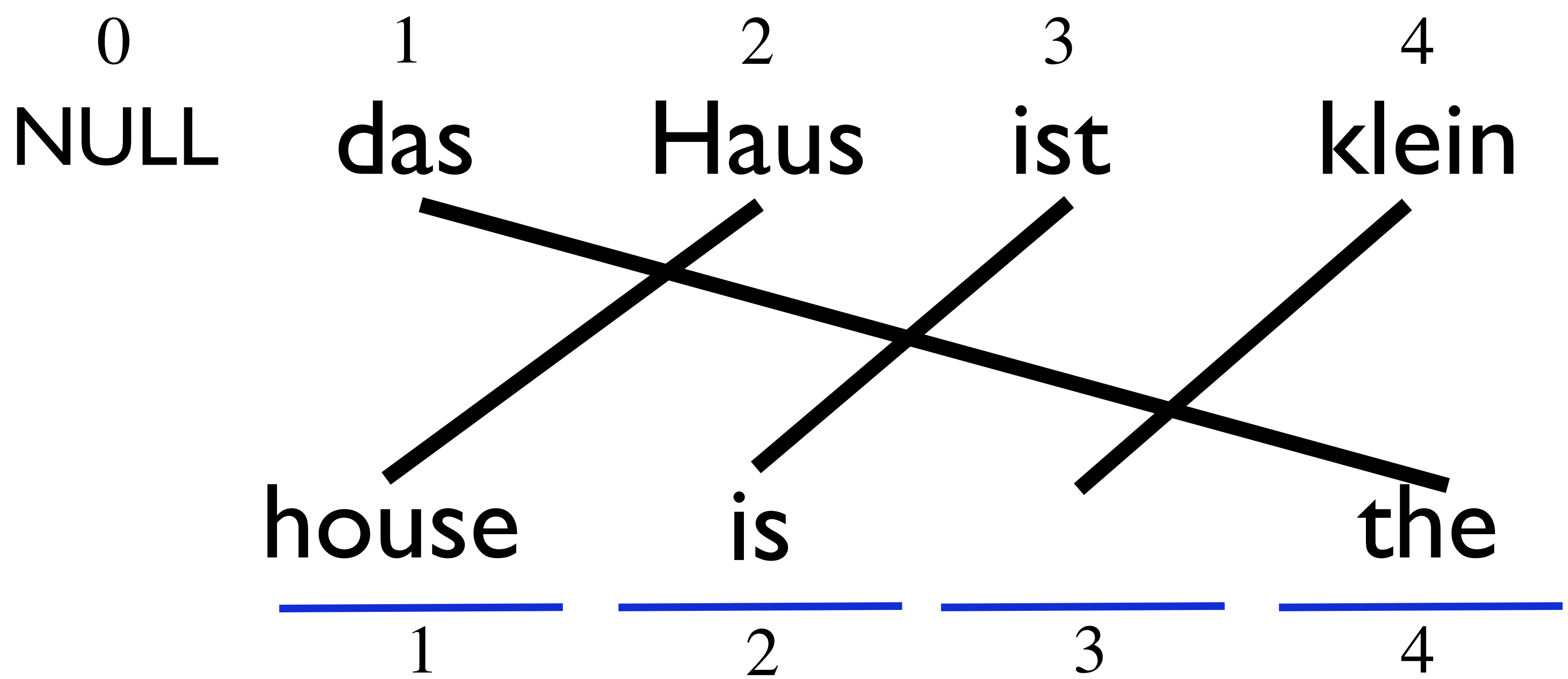
Example



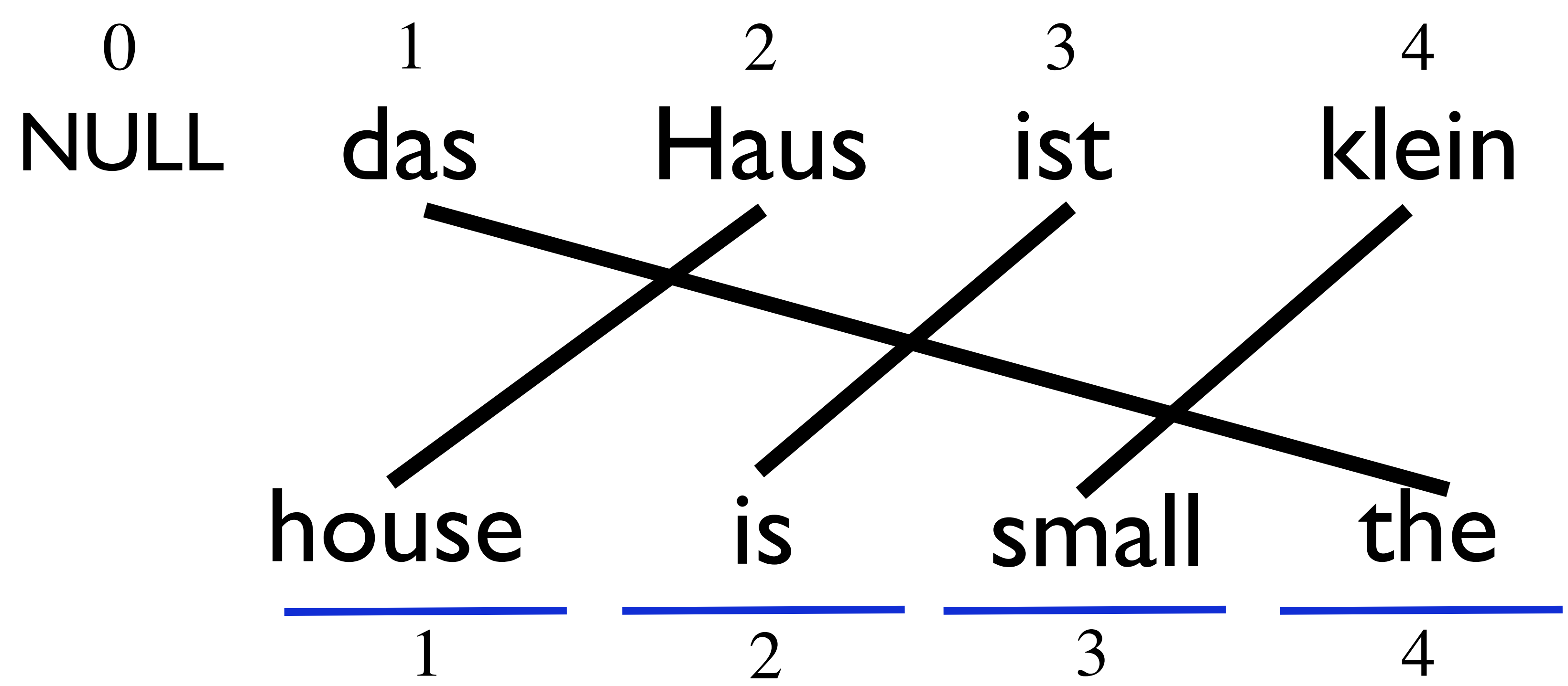
Example



Example



Example



Finding the Viterbi Alignment

$$
\begin{aligned}
\mathbf{a}^* &= \arg \max_{\mathbf{a} \in [0,1,\ldots,n]^m} p(\mathbf{a} \mid \mathbf{e}, \mathbf{f}) \\
&= \arg \max_{\mathbf{a} \in [0,1,\ldots,n]^m} \frac{p(\mathbf{e}, \mathbf{a} \mid \mathbf{f})}{\sum_{\mathbf{a}'} p(\mathbf{e}, \mathbf{a}' \mid \mathbf{f})} \\
&= \arg \max_{\mathbf{a} \in [0,1,\ldots,n]^m} p(\mathbf{e}, \mathbf{a} \mid \mathbf{f})
\end{aligned}
$$

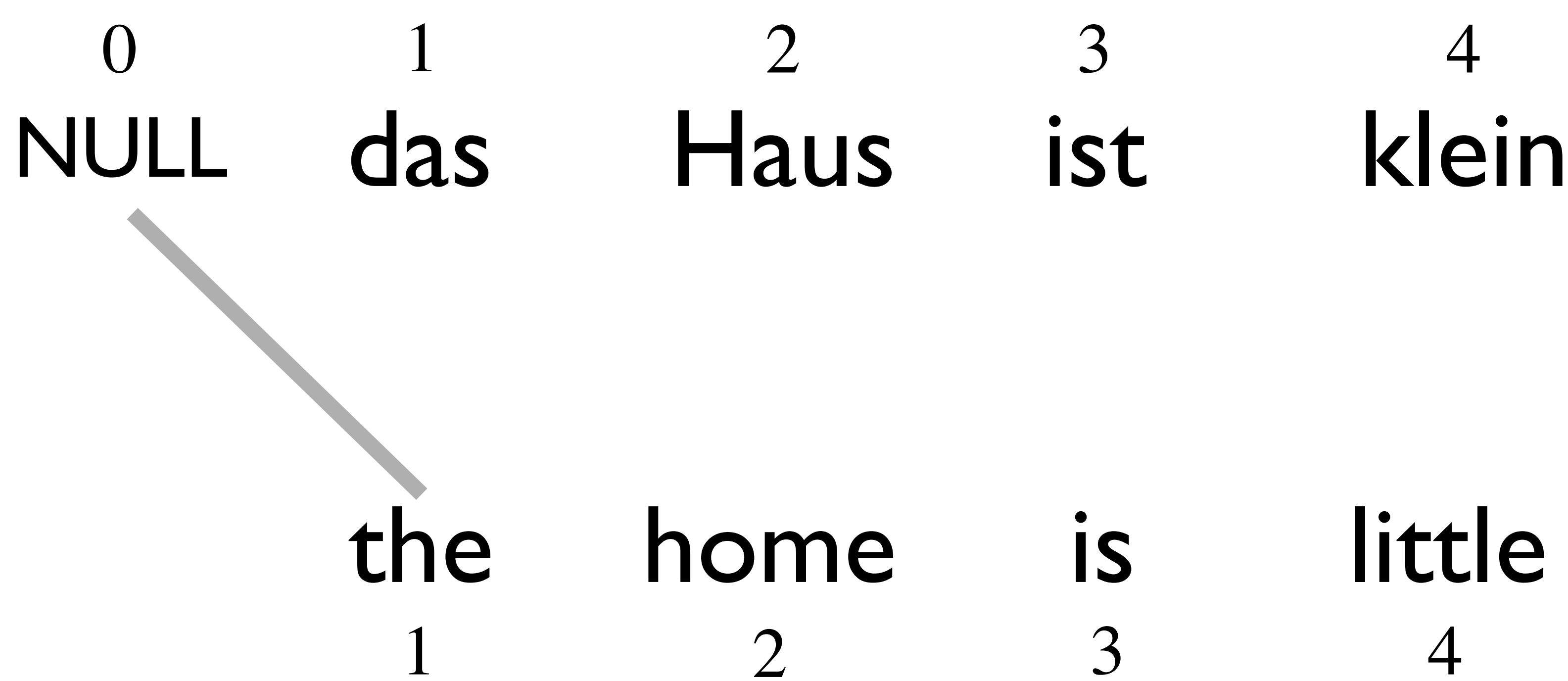
$$
\begin{aligned}
a_i^* &= \arg \max_{a_i = 0}^{n} \frac{1}{1+n} p(e_i \mid f_{a_i}) \\
&= \arg \max_{a_i = 0}^{n} p(e_i \mid f_{a_i})
\end{aligned}
$$

Finding the Viterbi Alignment

0	1	2	3	4
NULL	das	Haus	ist	klein

the	home	is	little
1	2	3	4

Finding the Viterbi Alignment



Finding the Viterbi Alignment

0	1	2	3	4
NULL	das	Haus	ist	klein
	the	home	is	little
	1	2	3	4

Finding the Viterbi Alignment

0	1	2	3	4
NULL	das	Haus	ist	klein

the	home	is	little
1	2	3	4

Finding the Viterbi Alignment

0	1	2	3	4
NULL	das	Haus	ist	klein

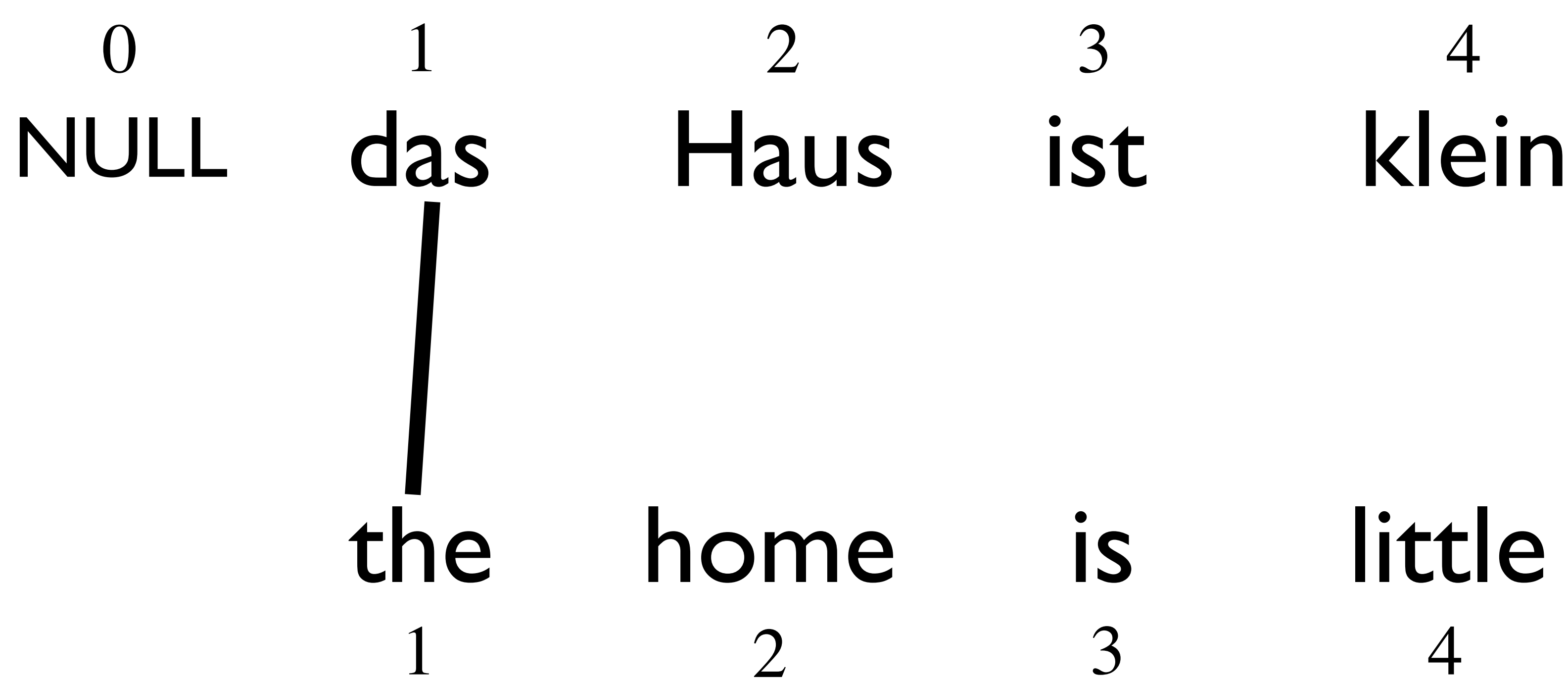
the	home	is	little
1	2	3	4

Finding the Viterbi Alignment

0	1	2	3	4
NULL	das	Haus	ist	klein

the	home	is	little
1	2	3	4

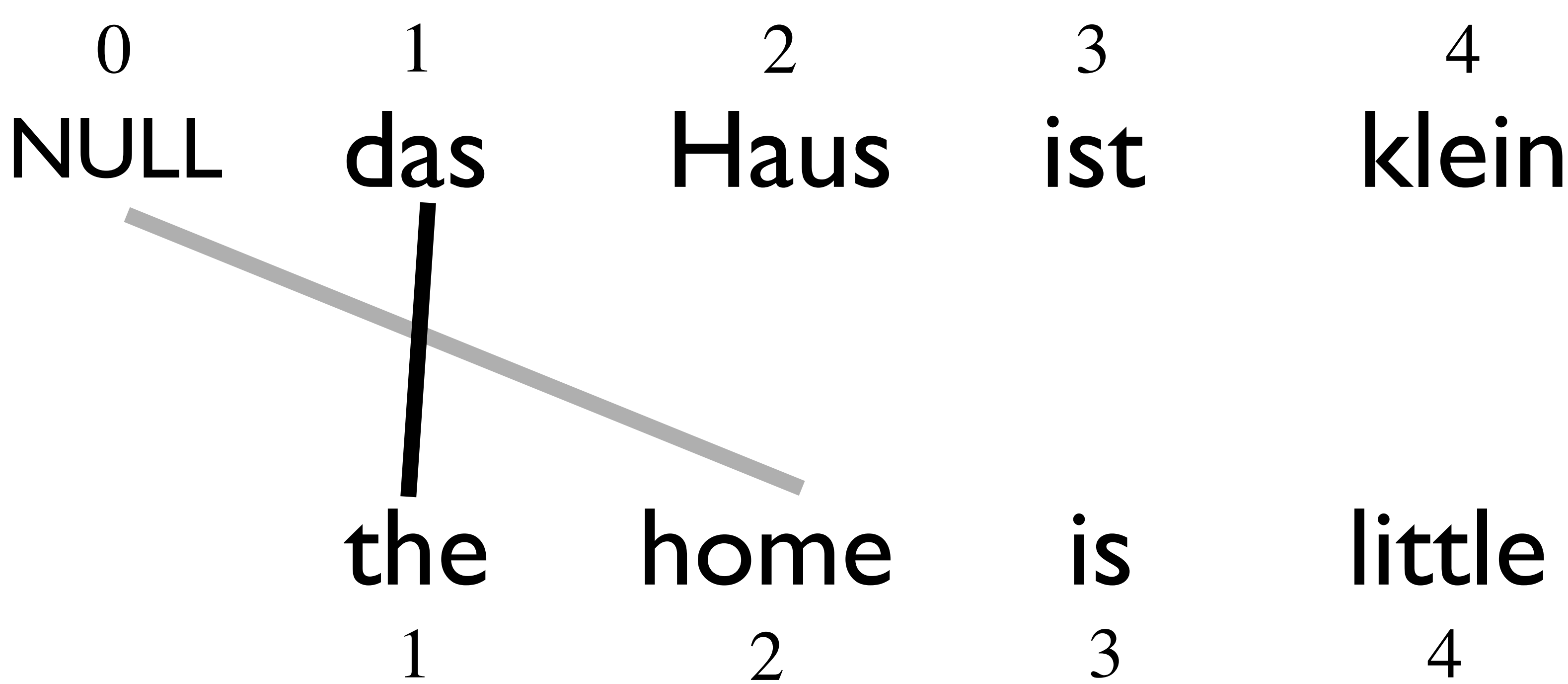
Finding the Viterbi Alignment



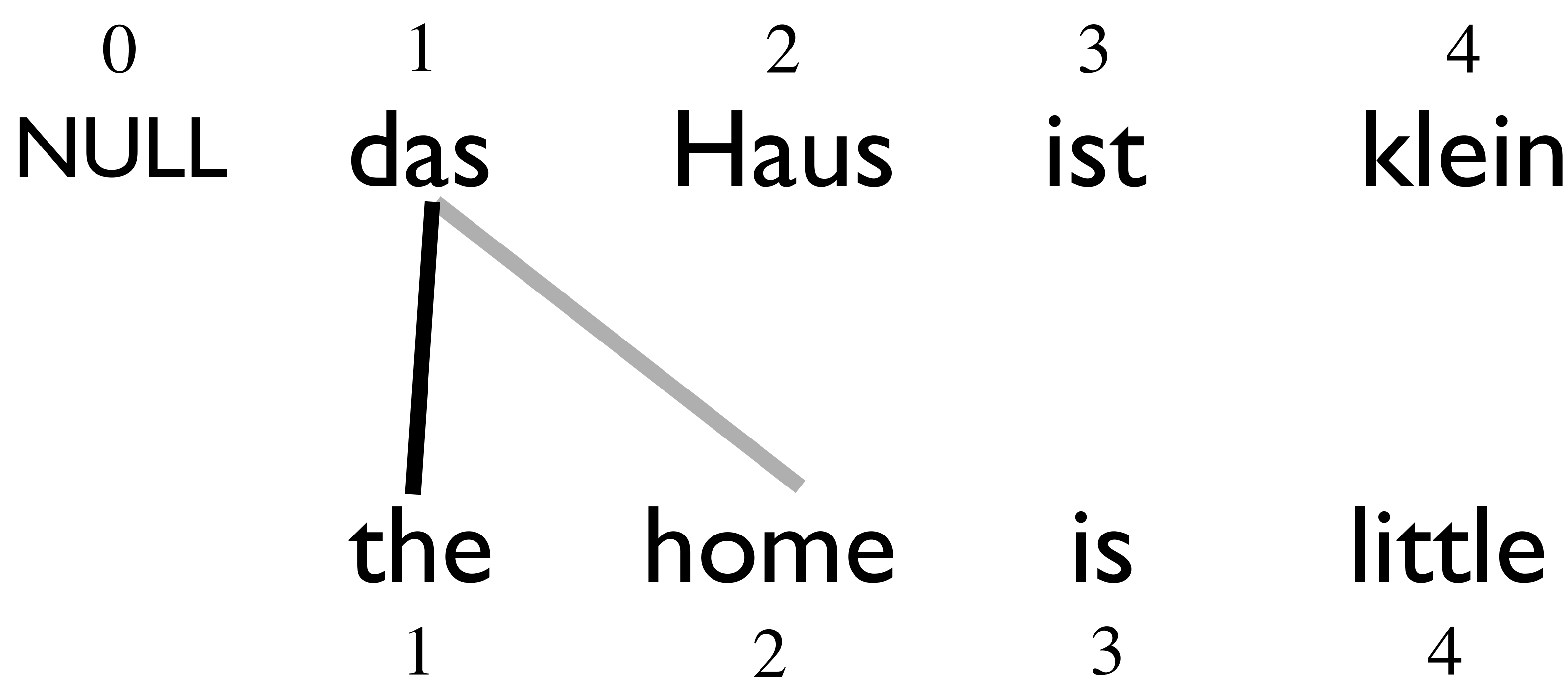
Finding the Viterbi Alignment

0	1	2	3	4
NULL	das	Haus	ist	klein
	the	home	is	little
	1	2	3	4

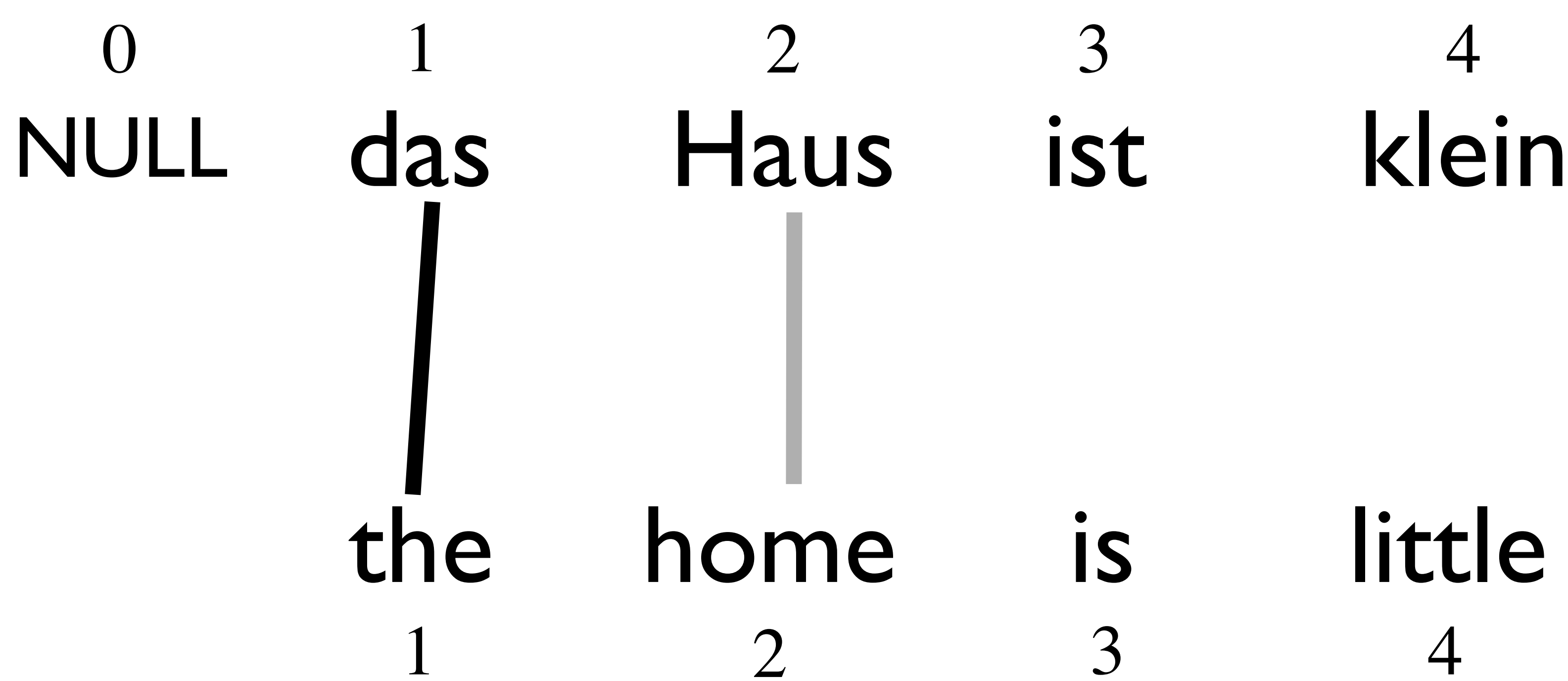
Finding the Viterbi Alignment



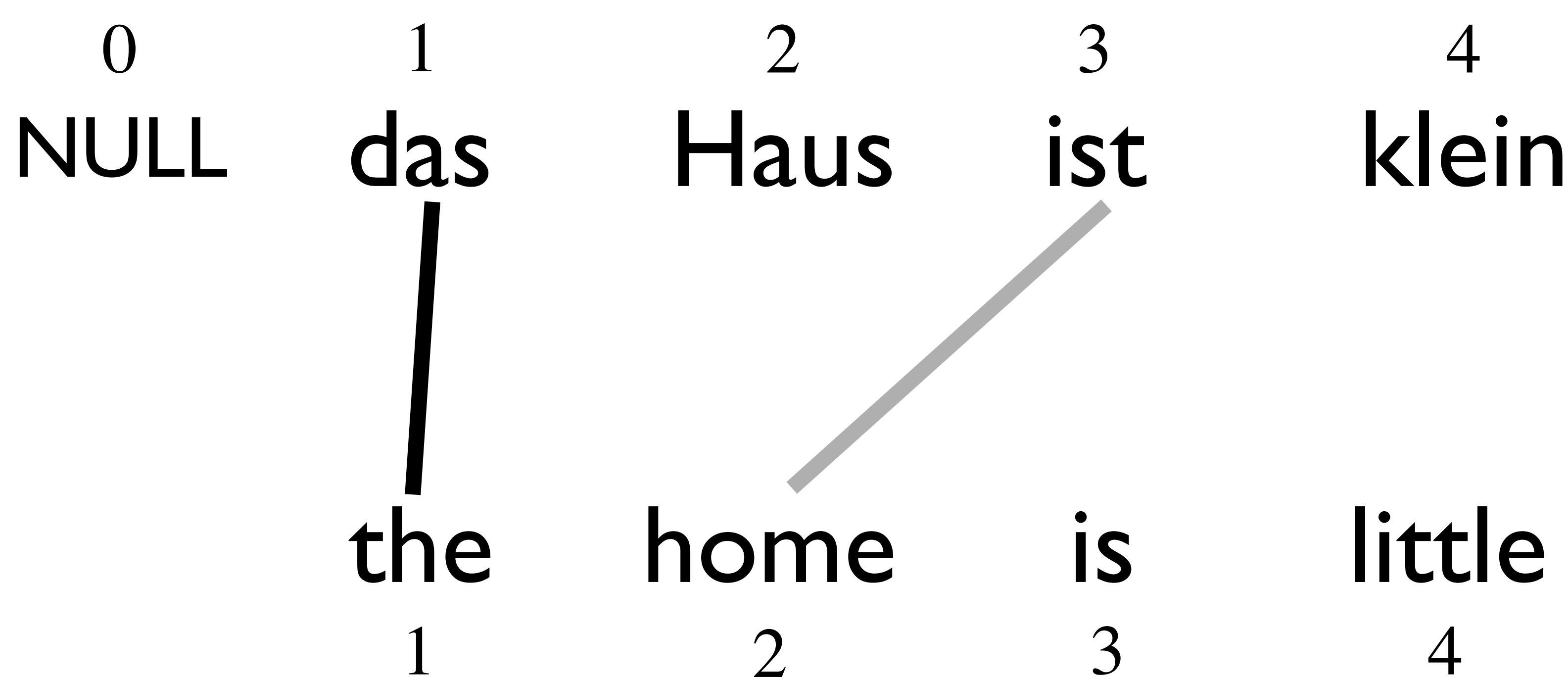
Finding the Viterbi Alignment



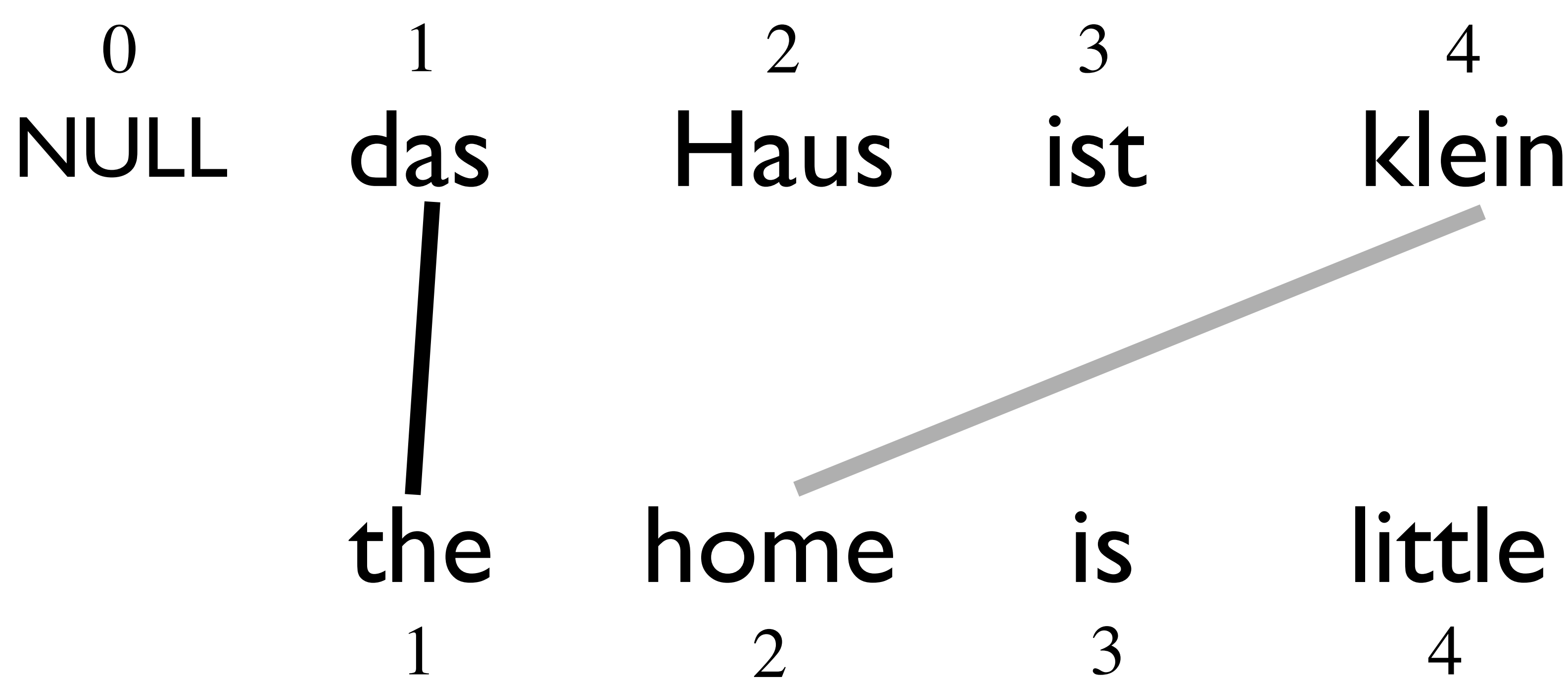
Finding the Viterbi Alignment



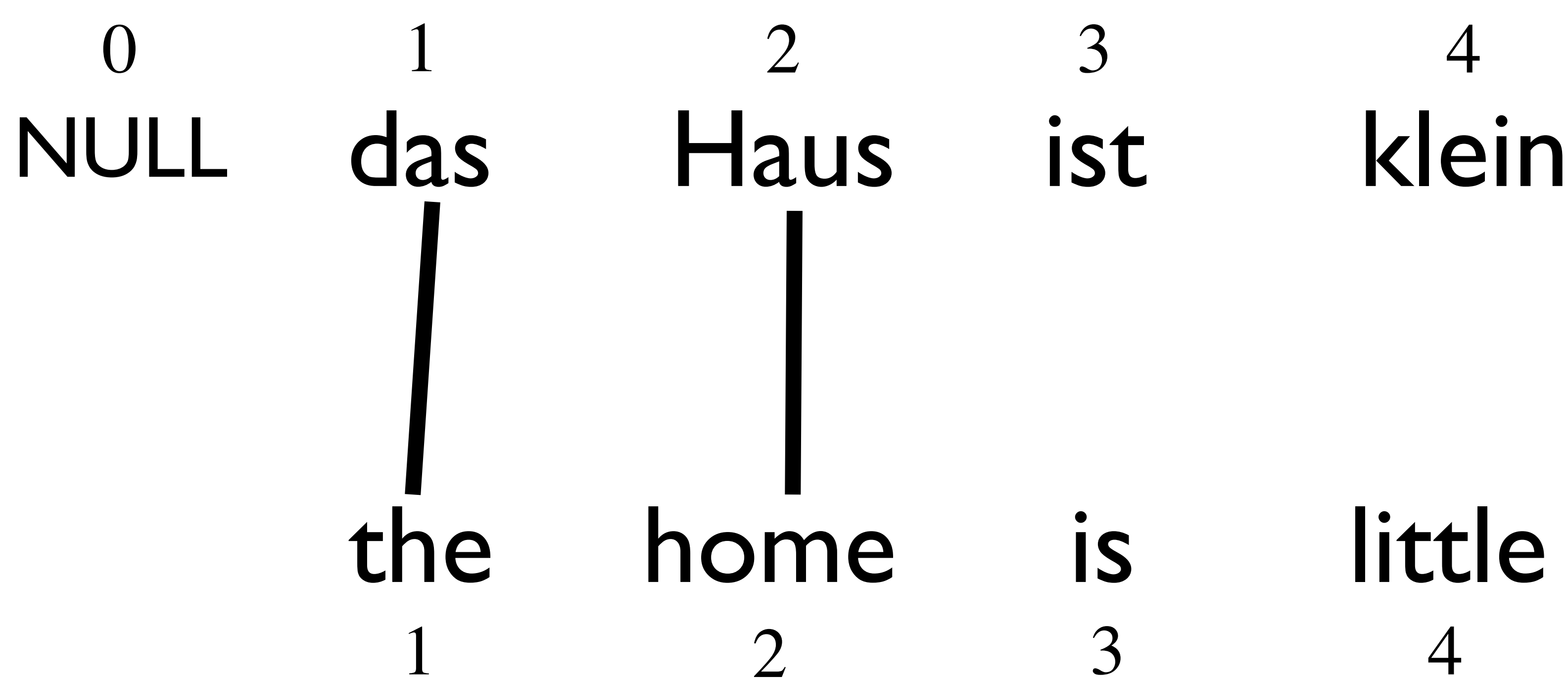
Finding the Viterbi Alignment



Finding the Viterbi Alignment



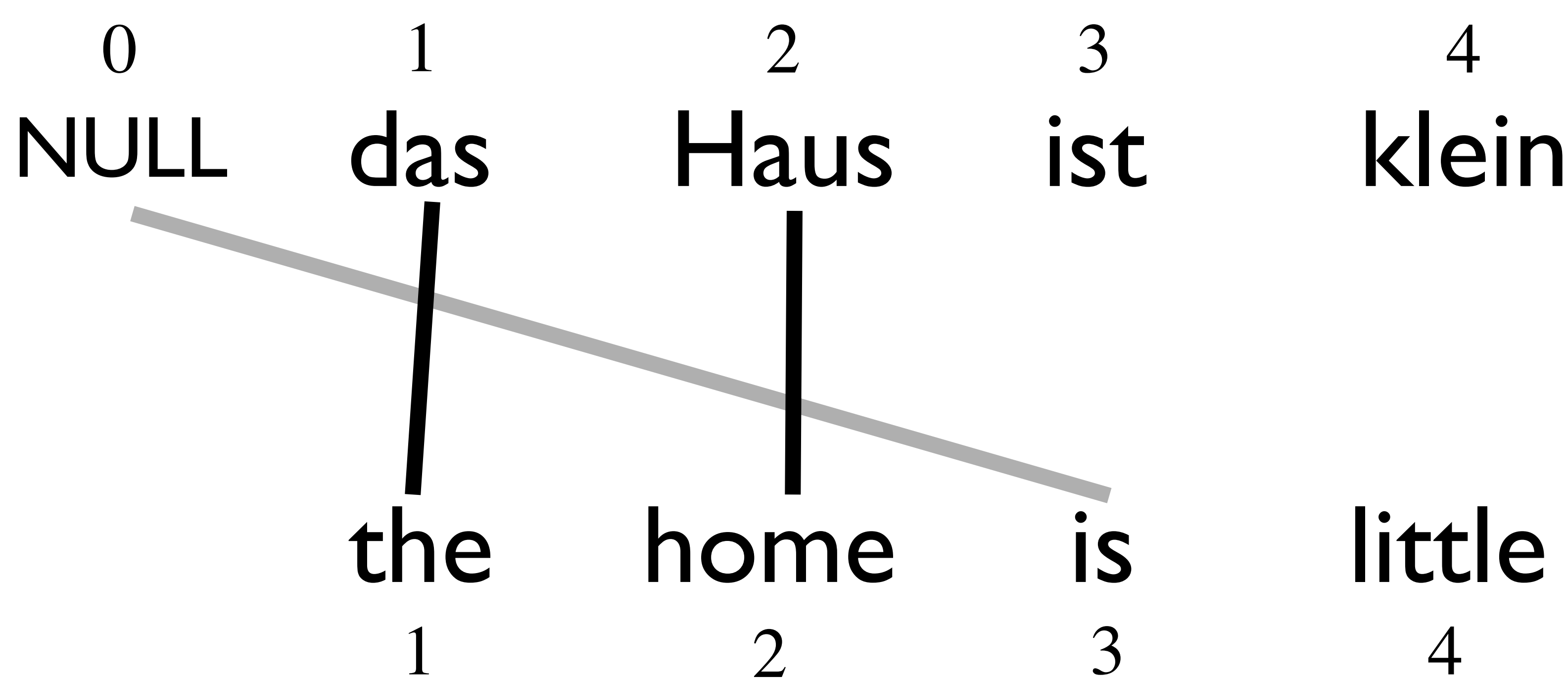
Finding the Viterbi Alignment



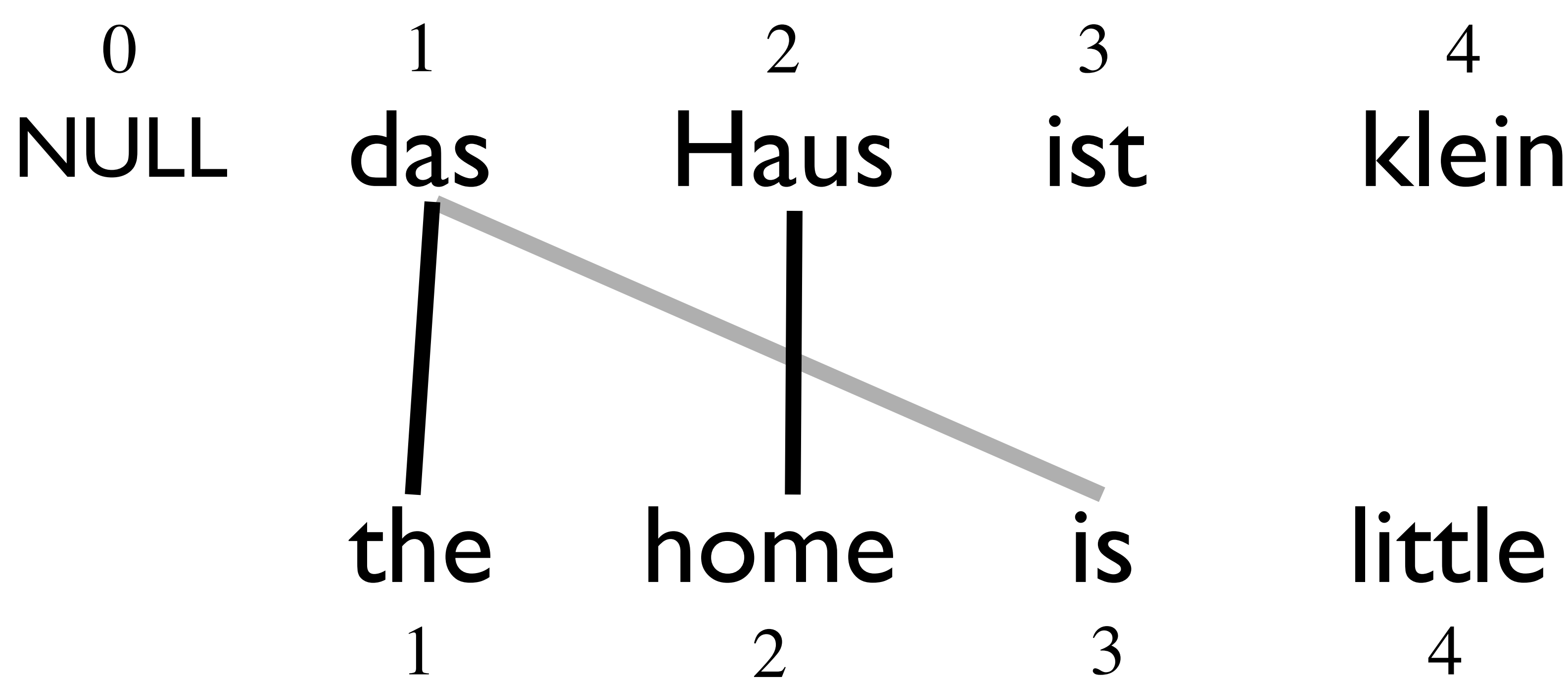
Finding the Viterbi Alignment

0	1	2	3	4
NULL	das	Haus	ist	klein
	the	home	is	little
	1	2	3	4

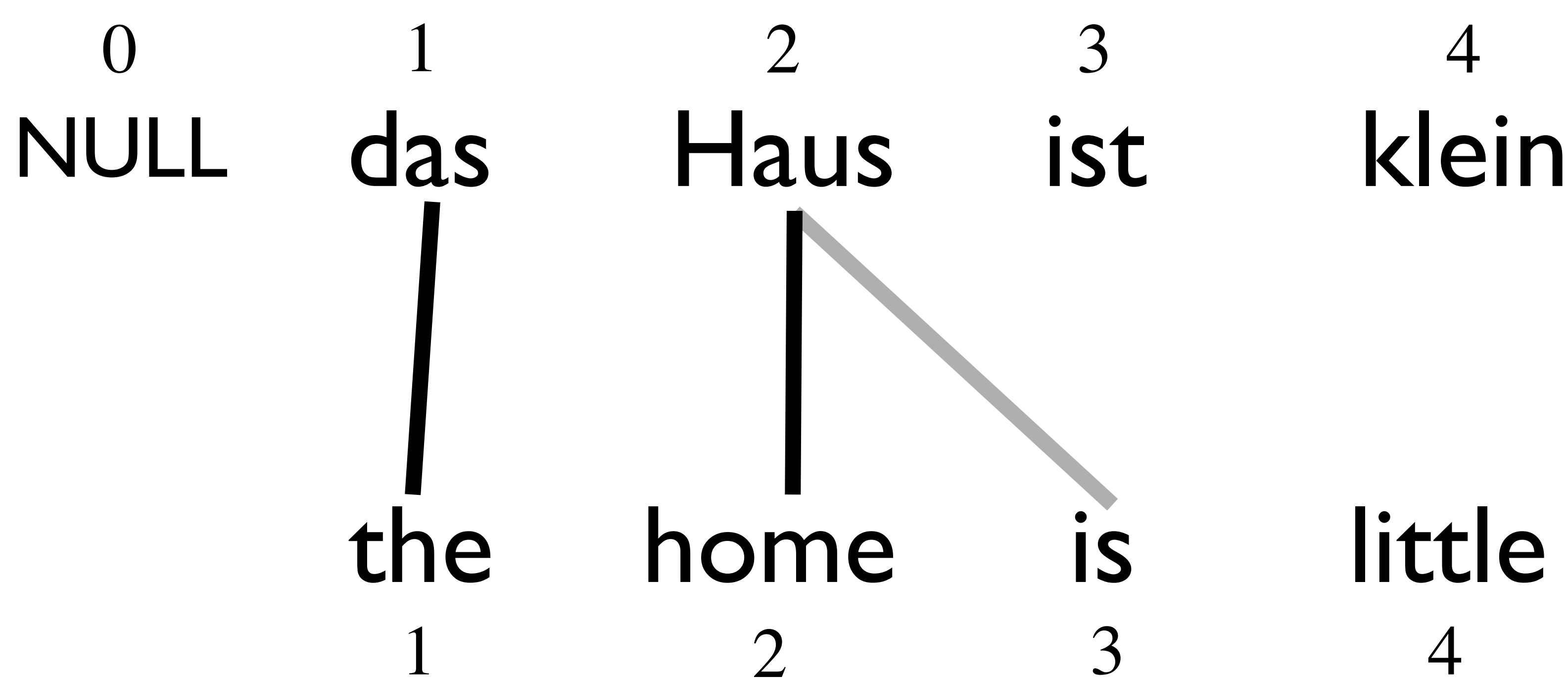
Finding the Viterbi Alignment



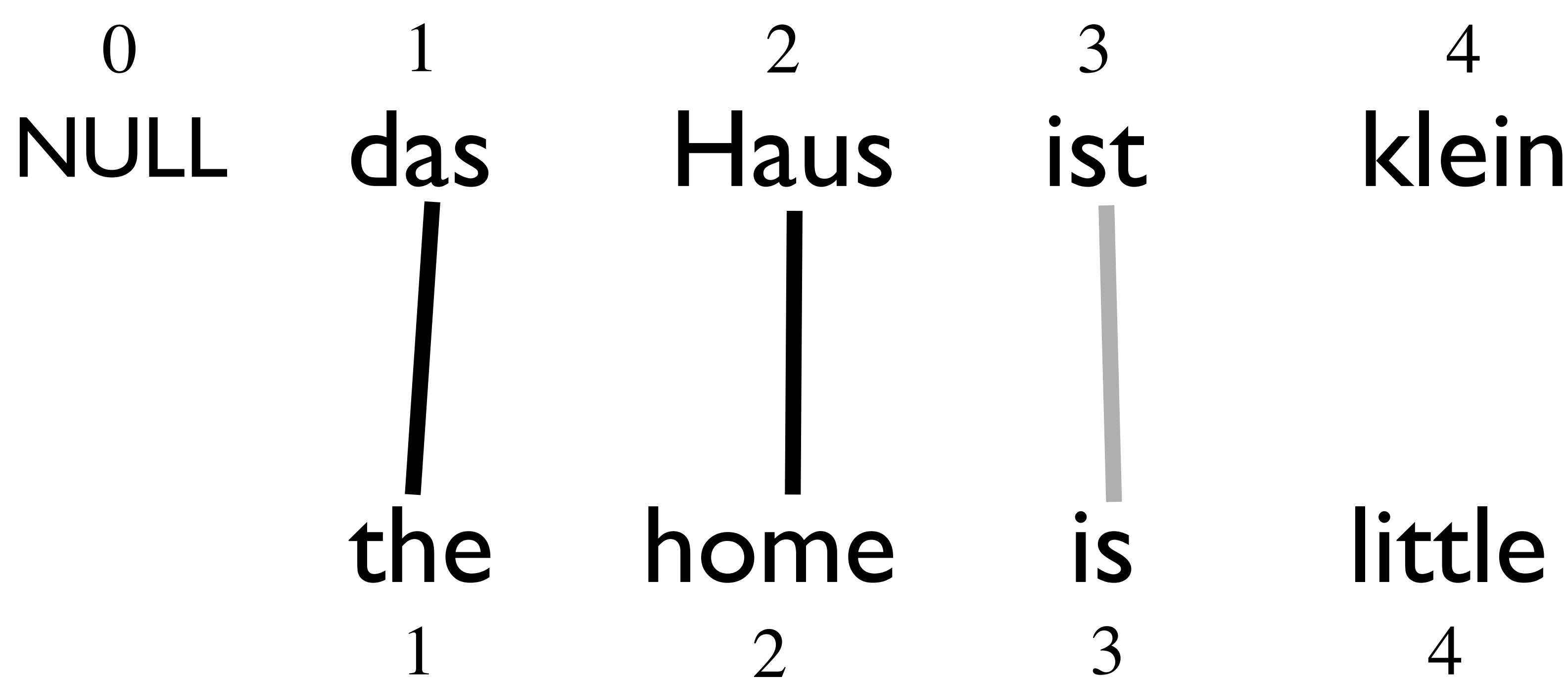
Finding the Viterbi Alignment



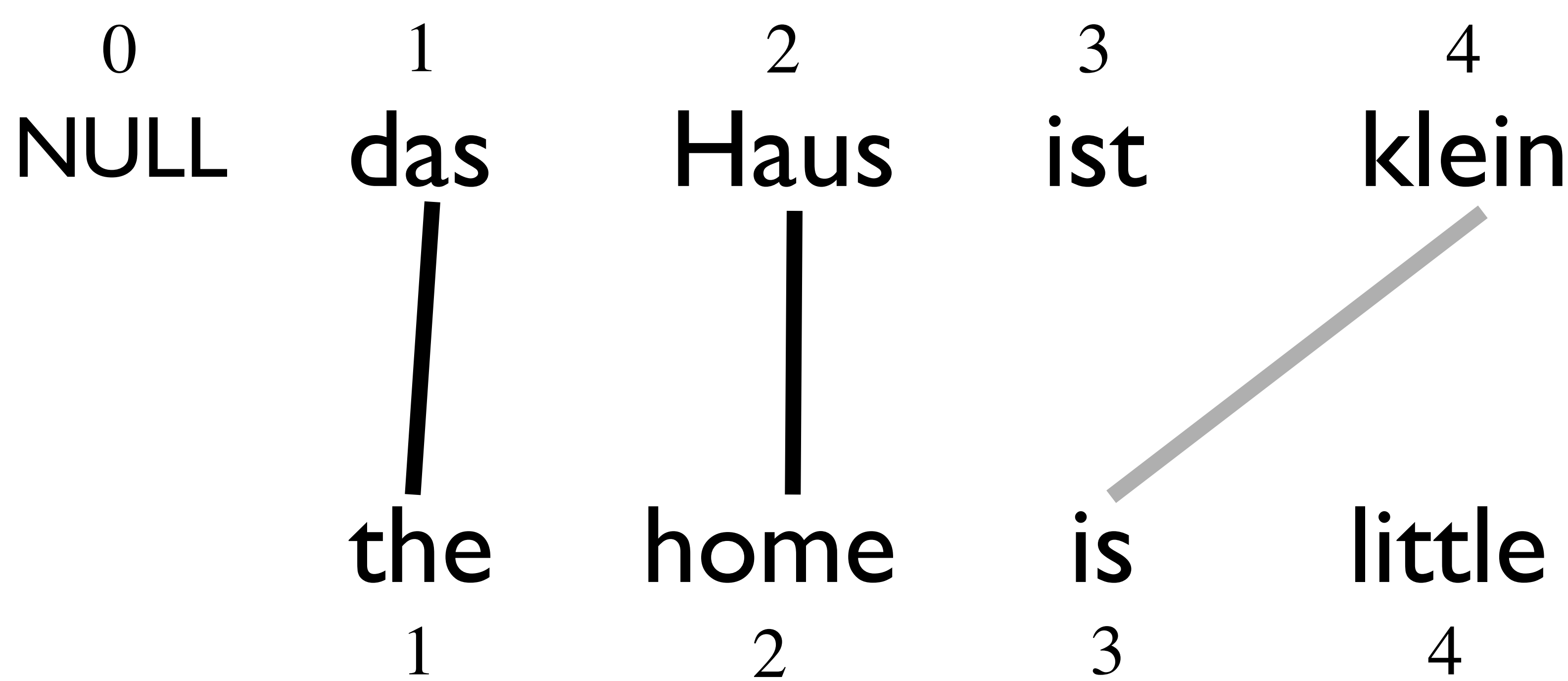
Finding the Viterbi Alignment



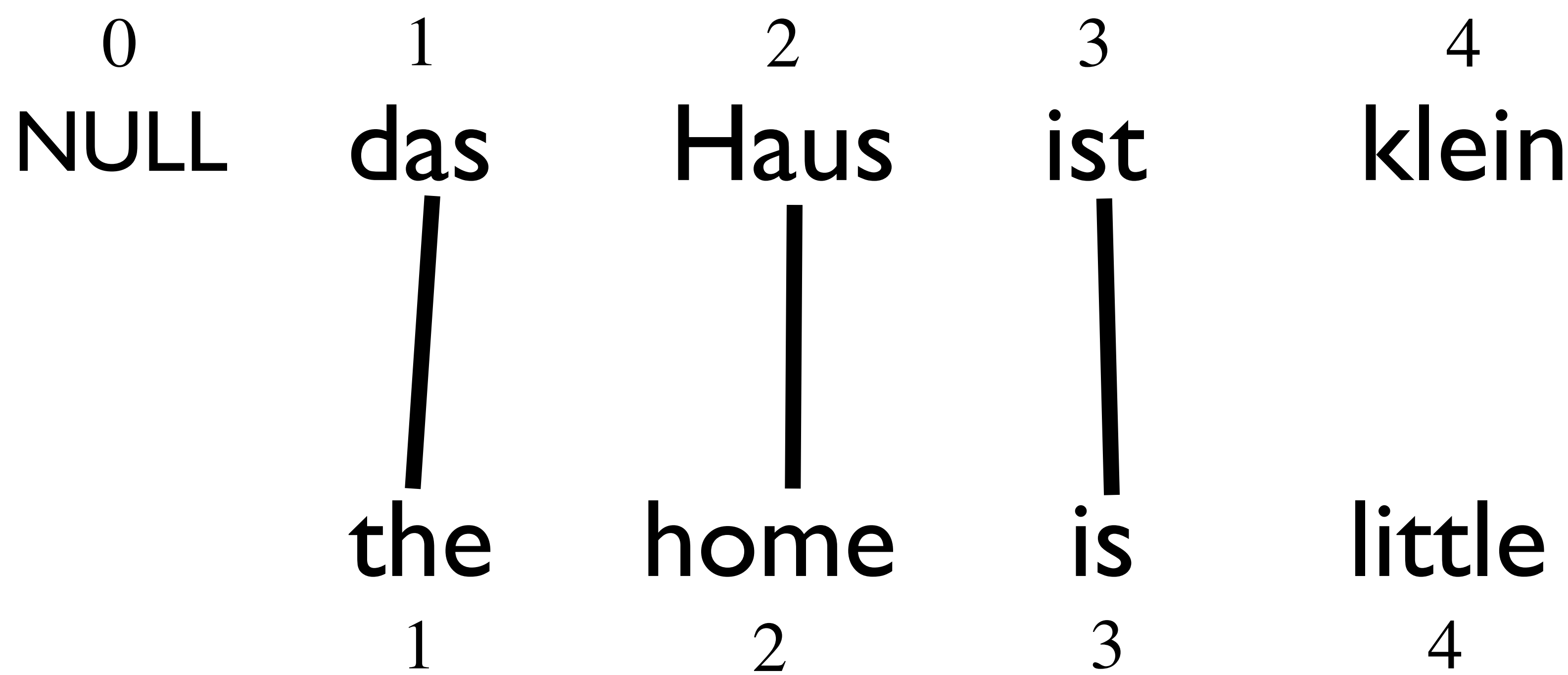
Finding the Viterbi Alignment



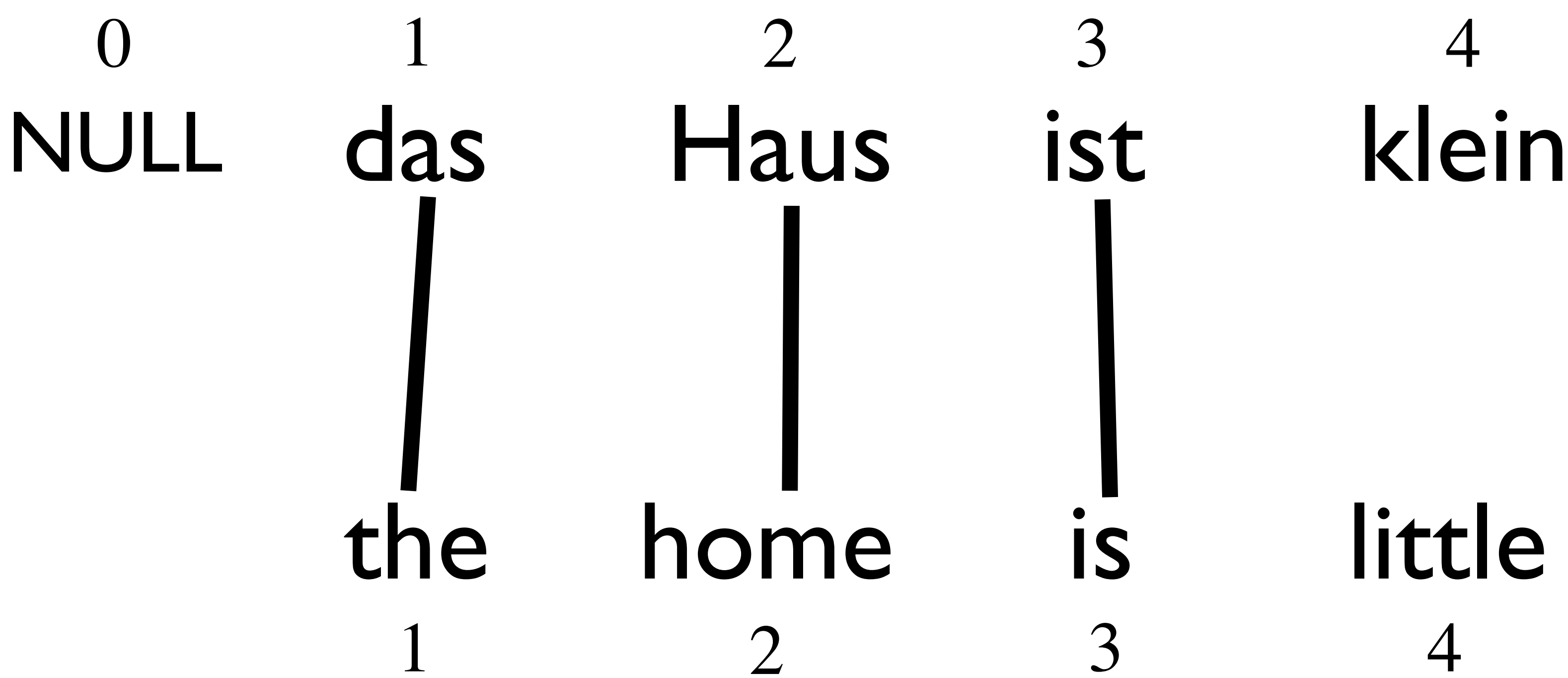
Finding the Viterbi Alignment



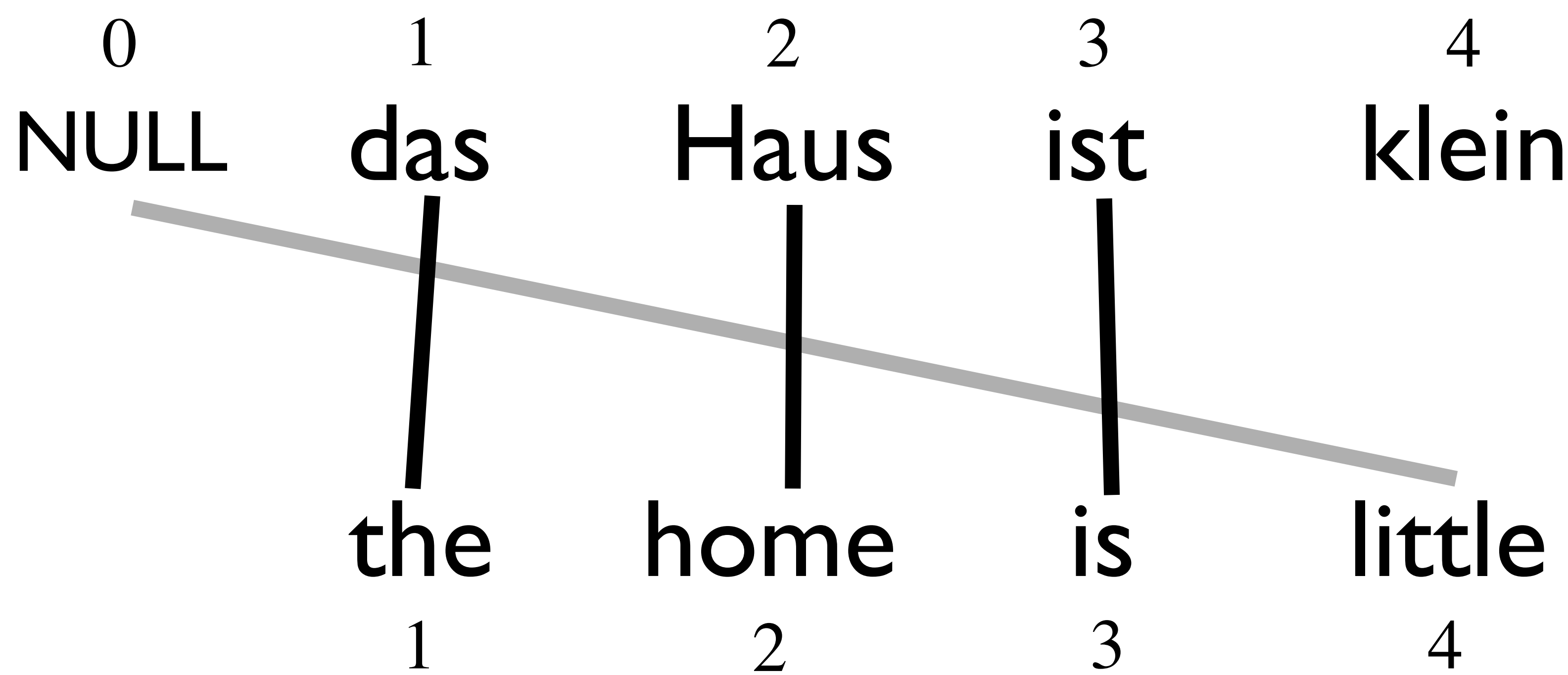
Finding the Viterbi Alignment



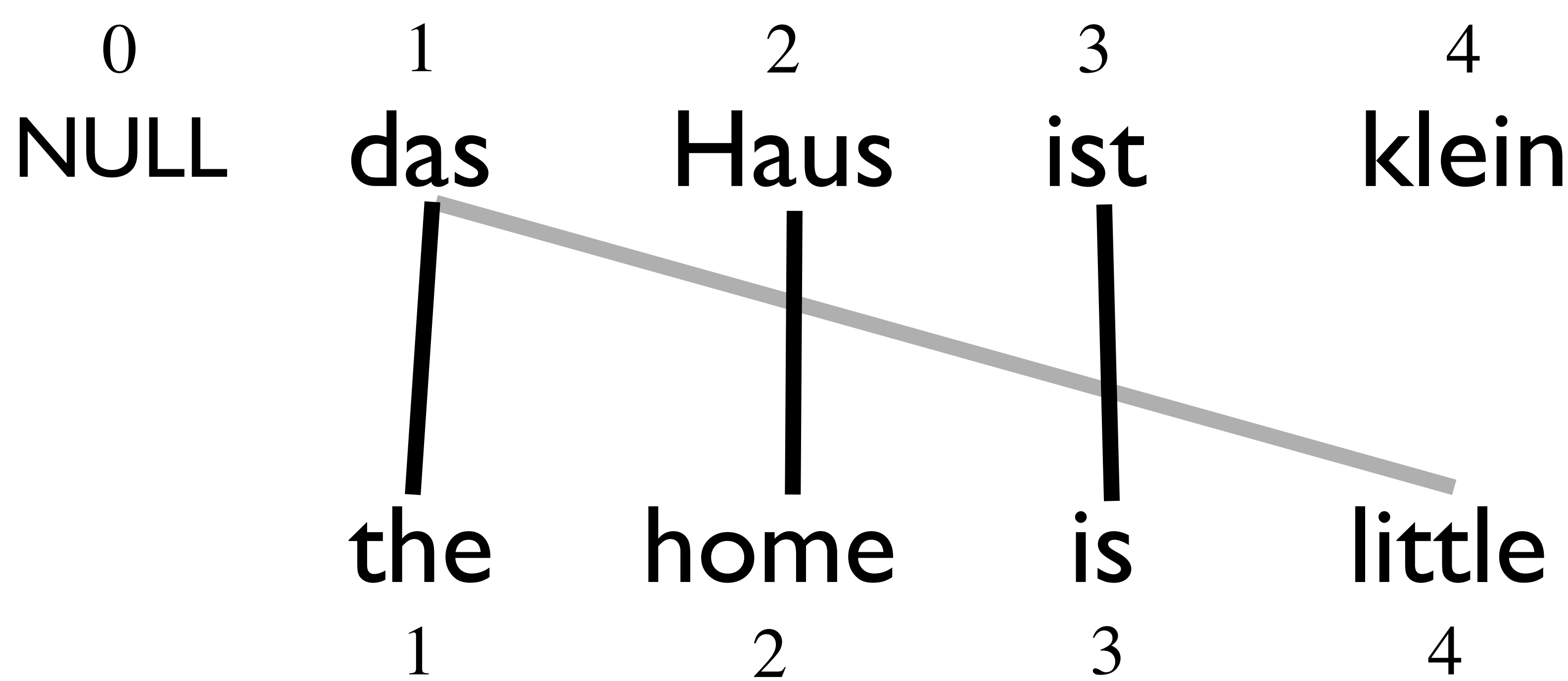
Finding the Viterbi Alignment



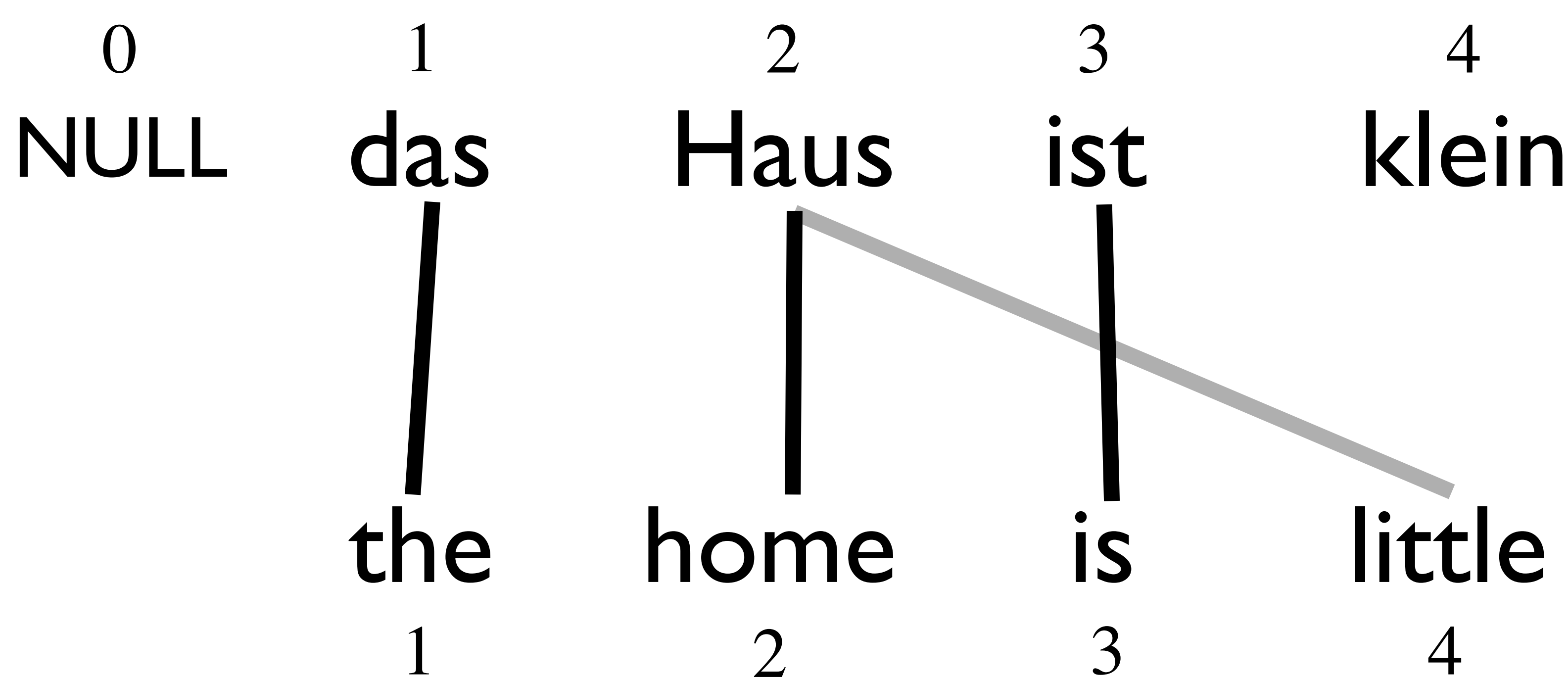
Finding the Viterbi Alignment



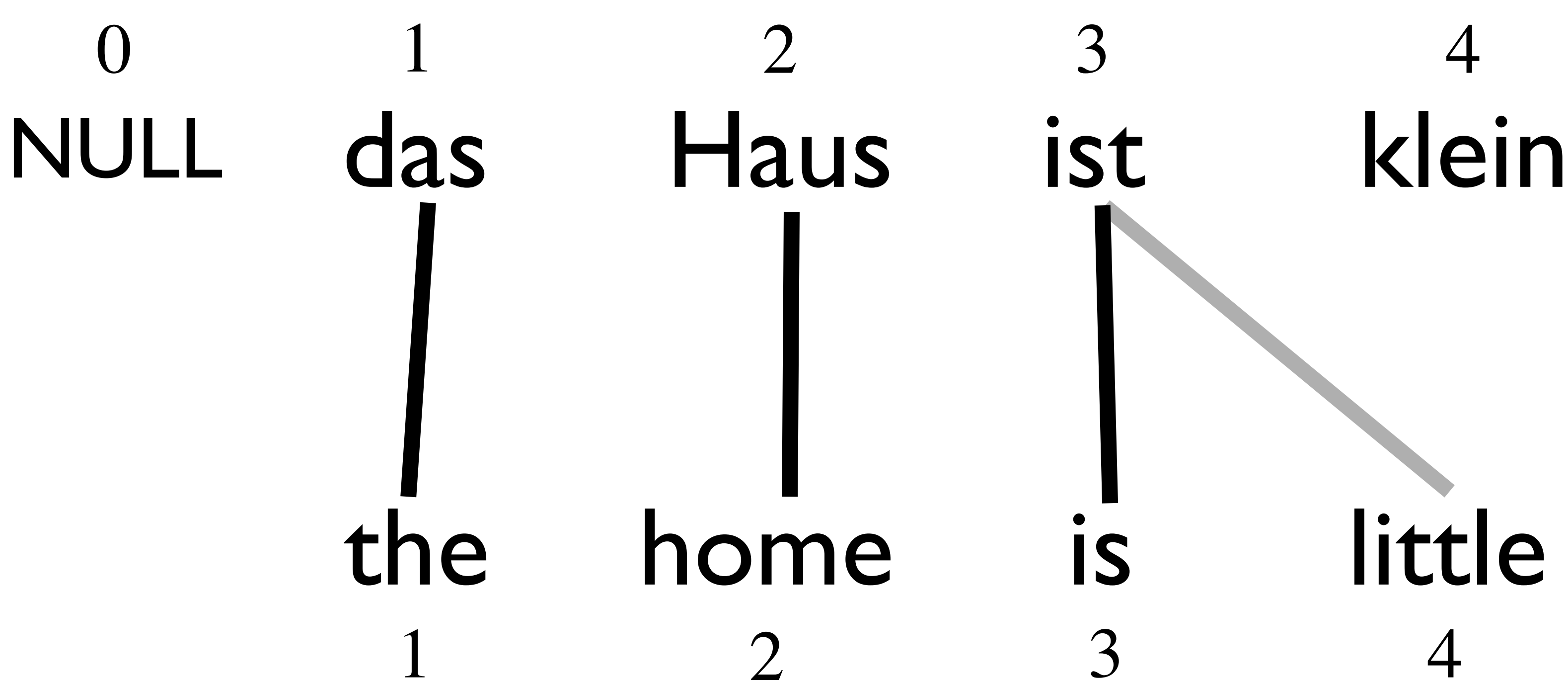
Finding the Viterbi Alignment



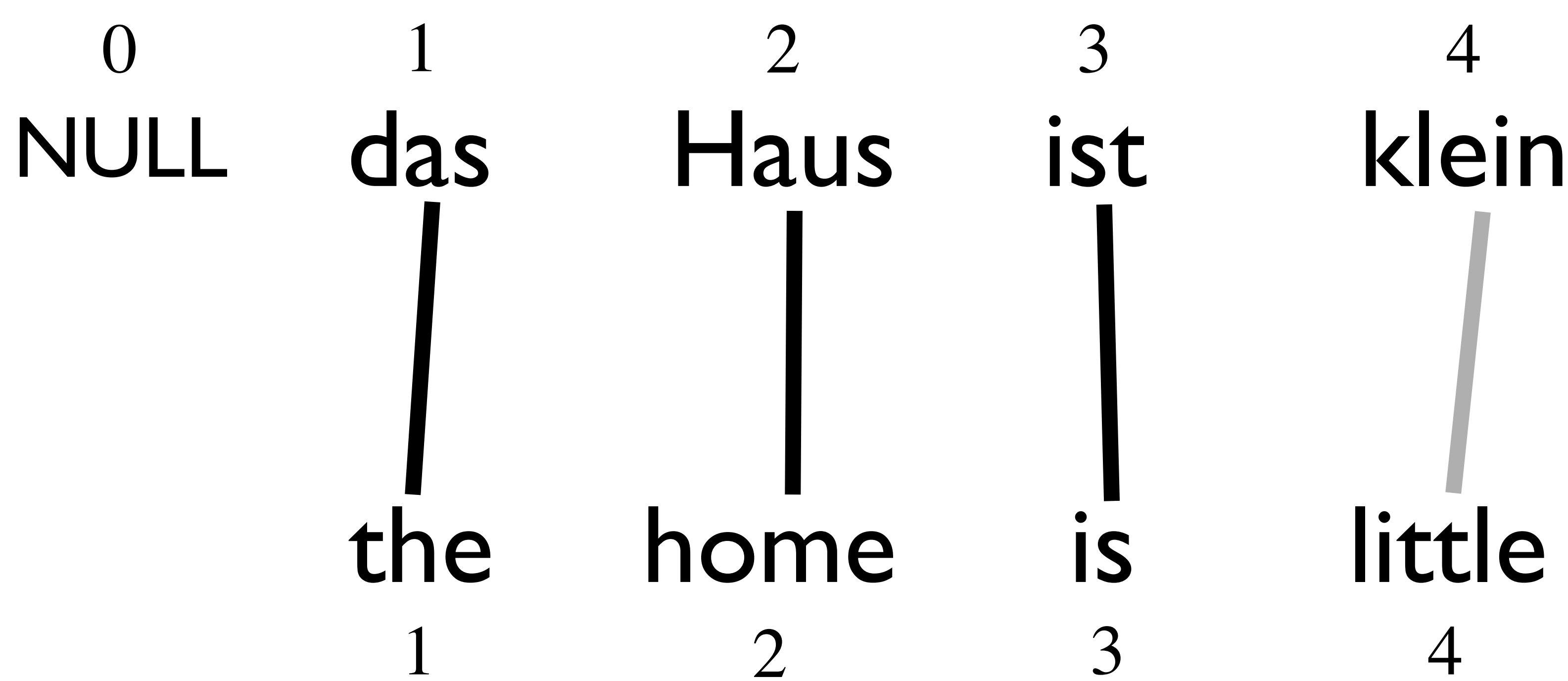
Finding the Viterbi Alignment



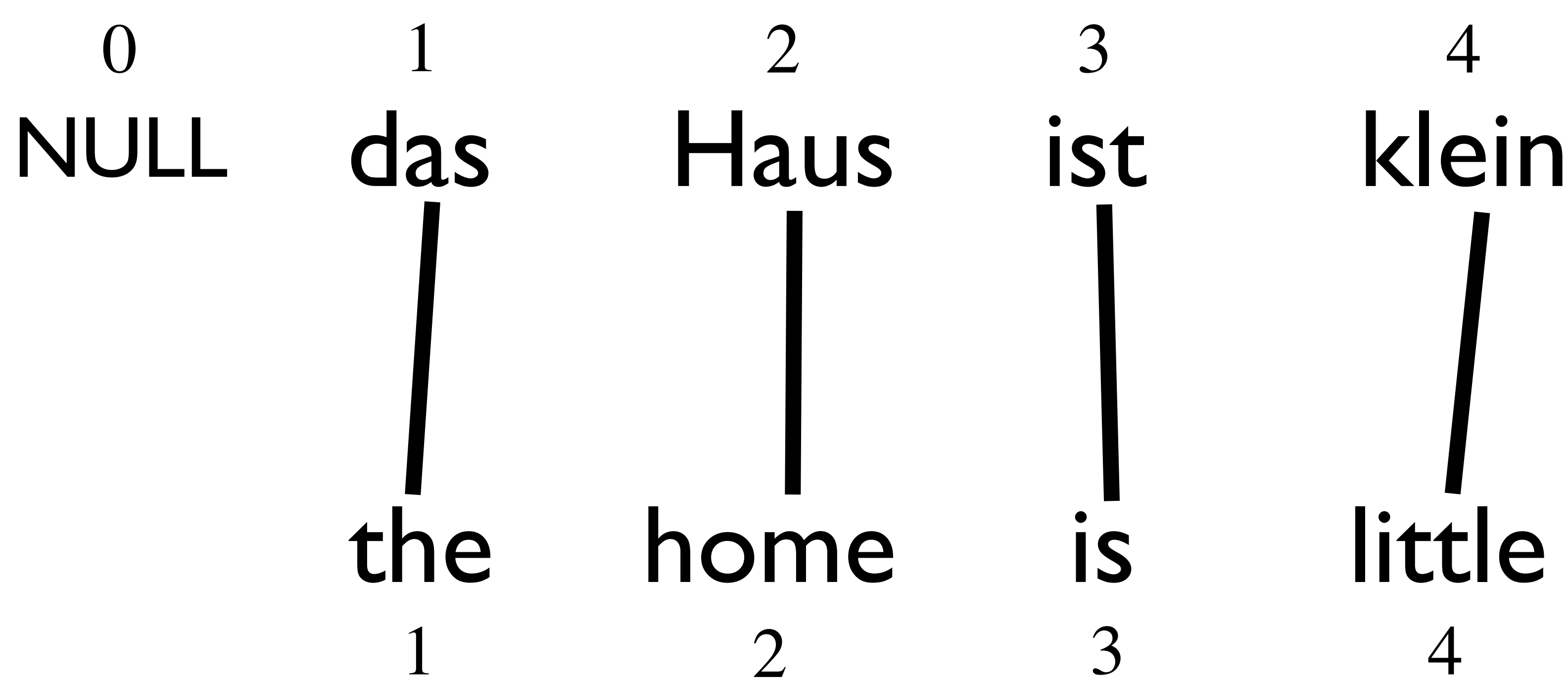
Finding the Viterbi Alignment



Finding the Viterbi Alignment

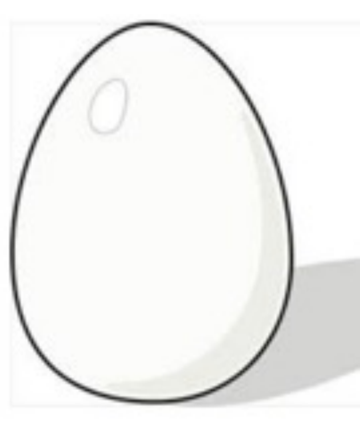


Finding the Viterbi Alignment



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



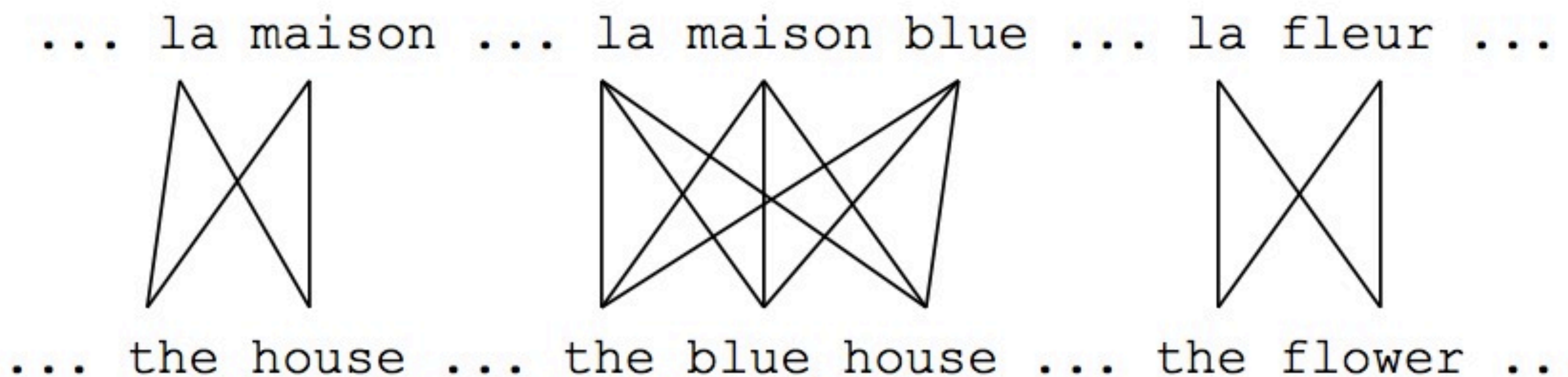
EM Algorithm

- 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

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

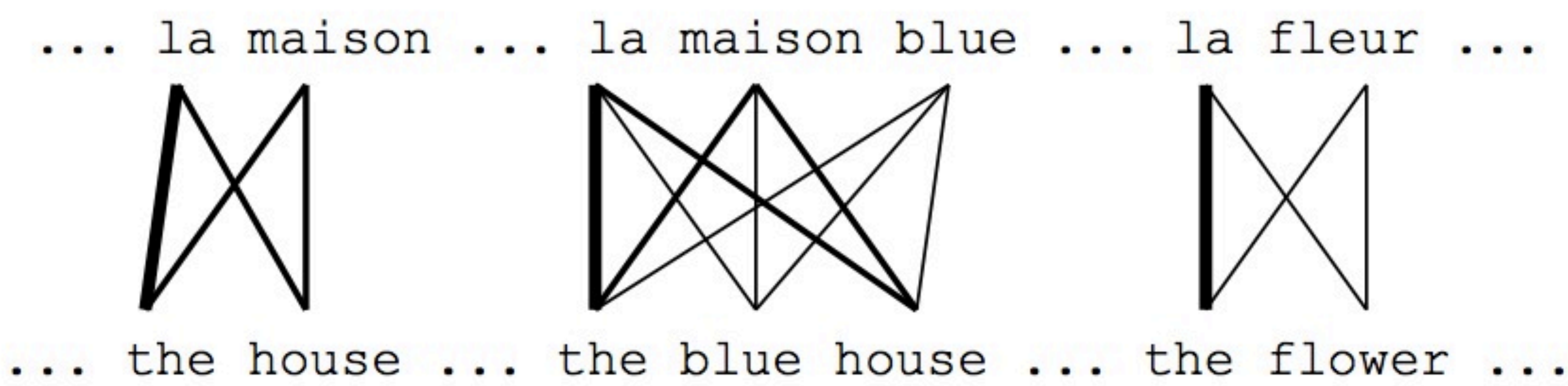
  - keep track of the expected number of times f translates into e throughout the whole corpus
  - keep track of the expected number of times that f is used as the source of any translation
  - use these expected counts as if they were "real" counts in the standard MLE equation

EM for Model 1



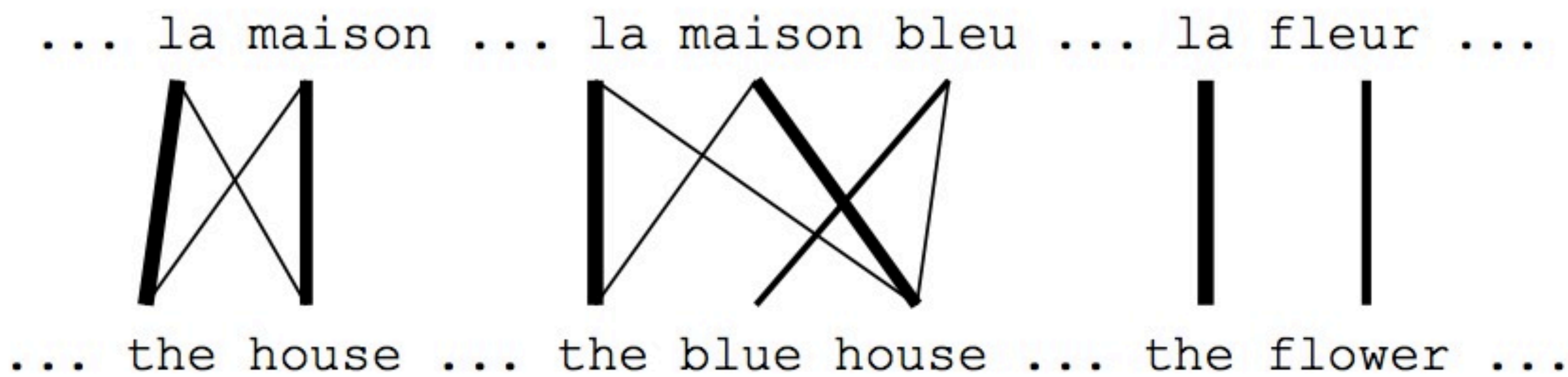
- Initial step: all alignments equally likely
- Model learns that, e.g., la is often aligned with the

EM for Model 1



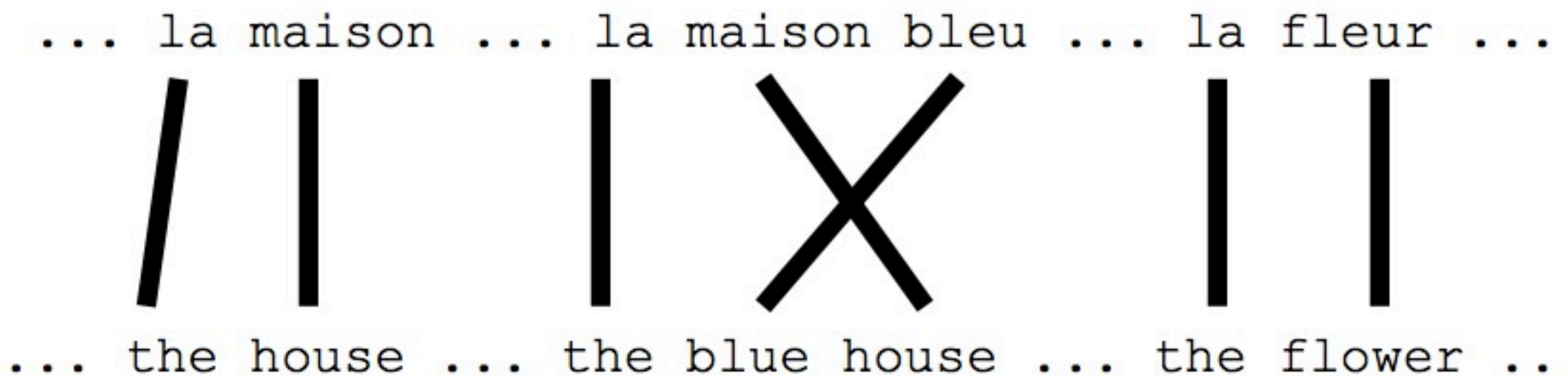
- After one iteration
- Alignments, e.g., between la and the are more likely

EM for Model 1



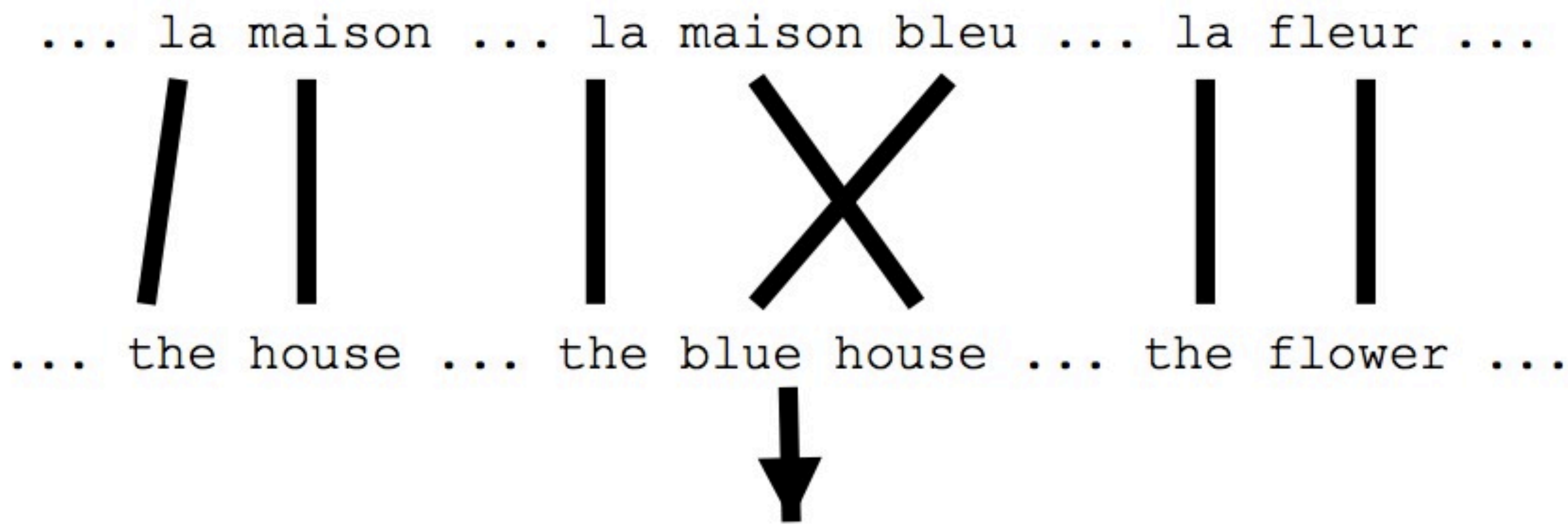
- After another iteration
- It becomes apparent that alignments, e.g., between fleur and flower are more likely (pigeon hole principle)

EM for Model 1



- Convergence
- Inherent hidden structure revealed by EM

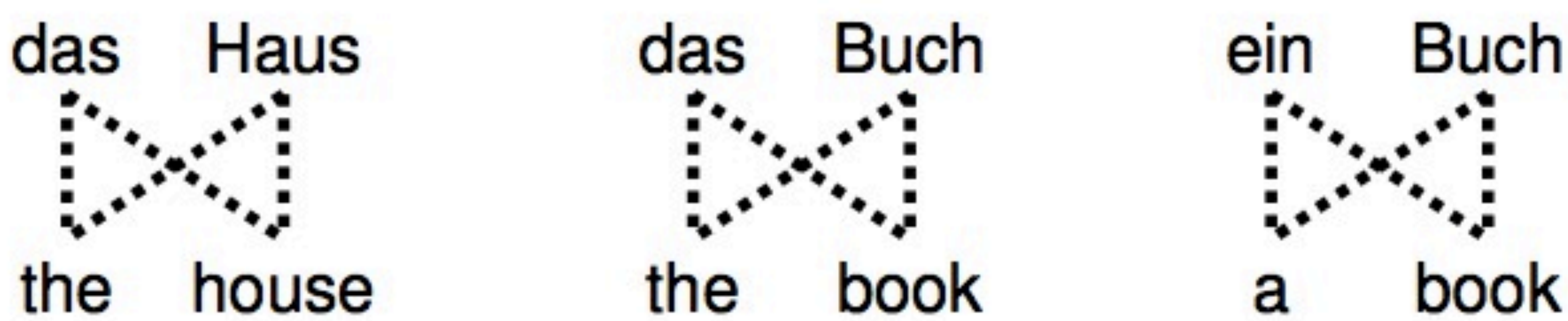
EM for Model 1



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

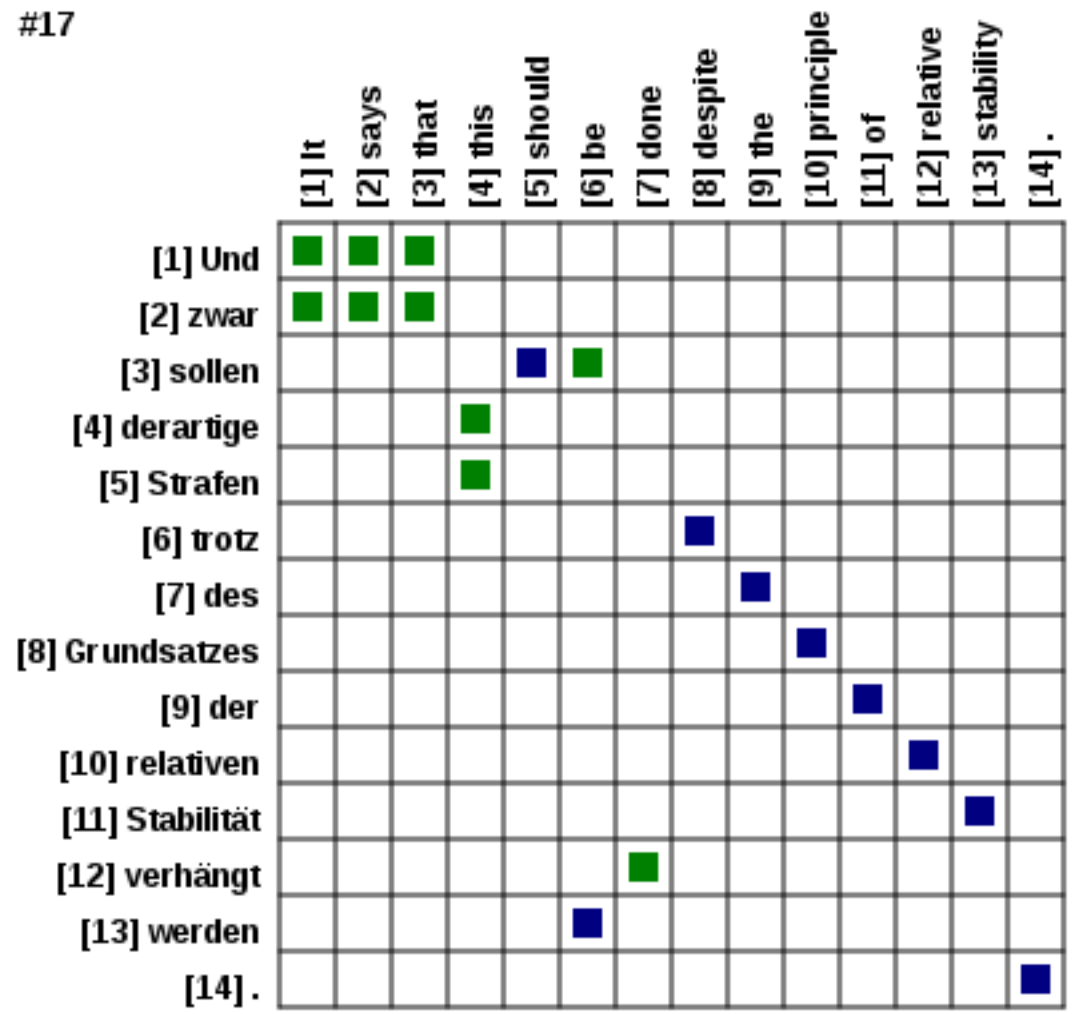
Evaluation

- 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	2.30	2.21	...	2

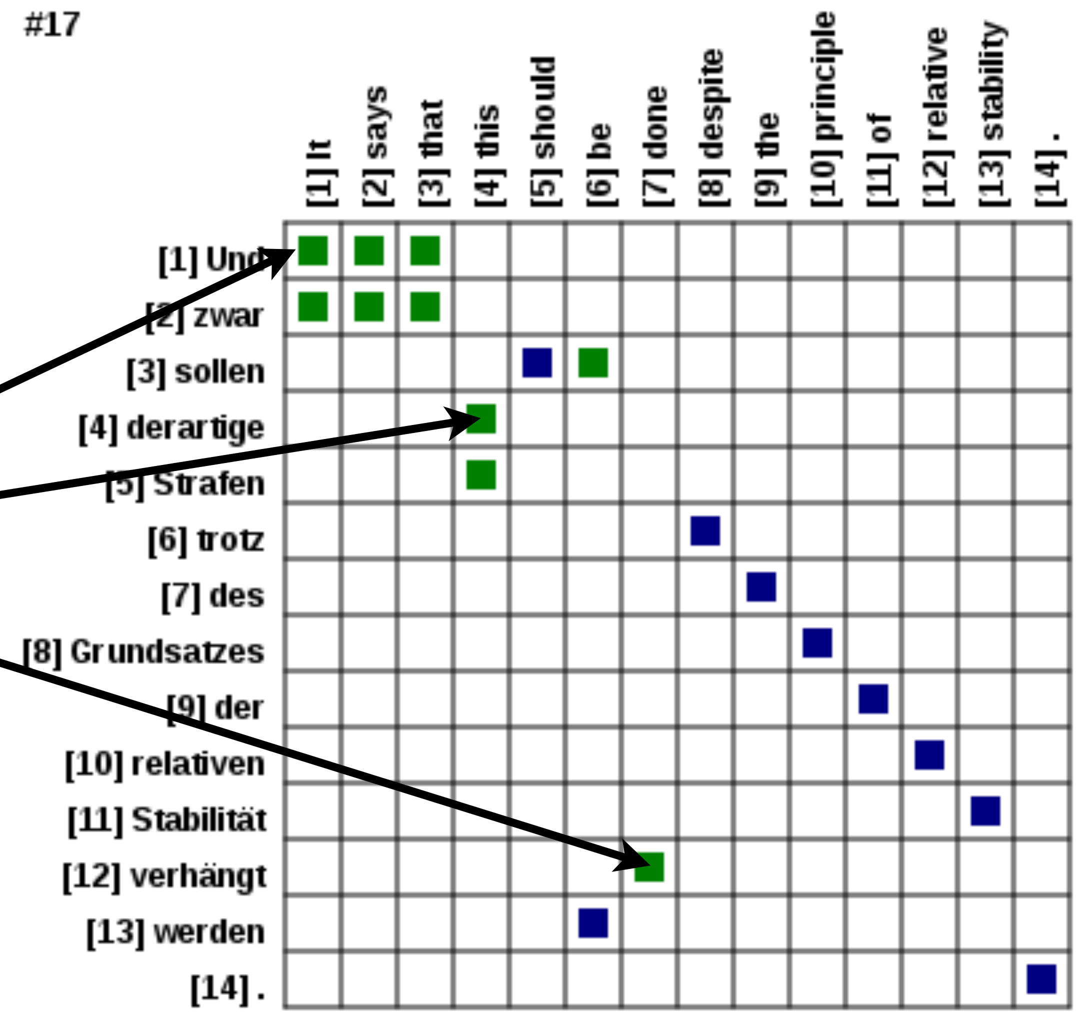
Alignment Error Rate



Alignment Error Rate

Possible links

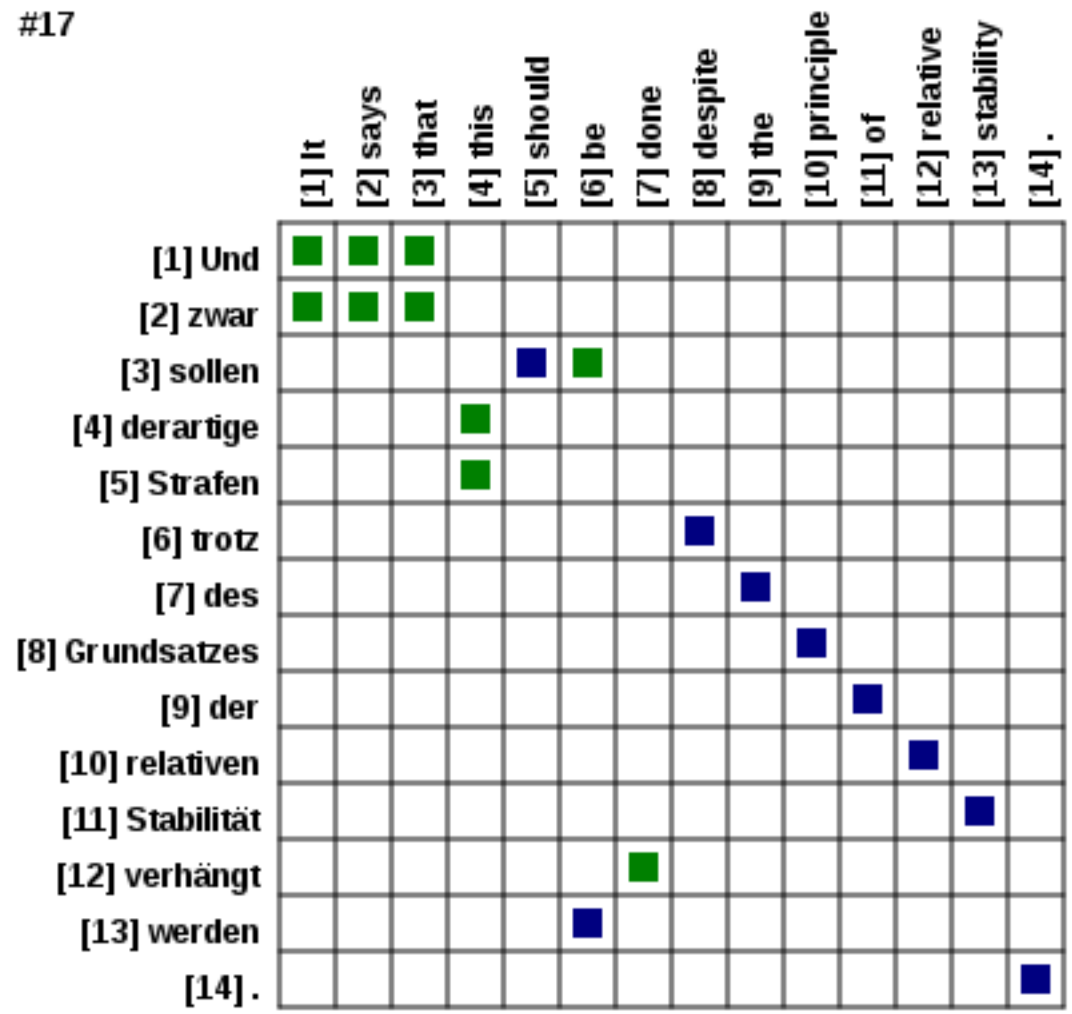
P



Alignment Error Rate

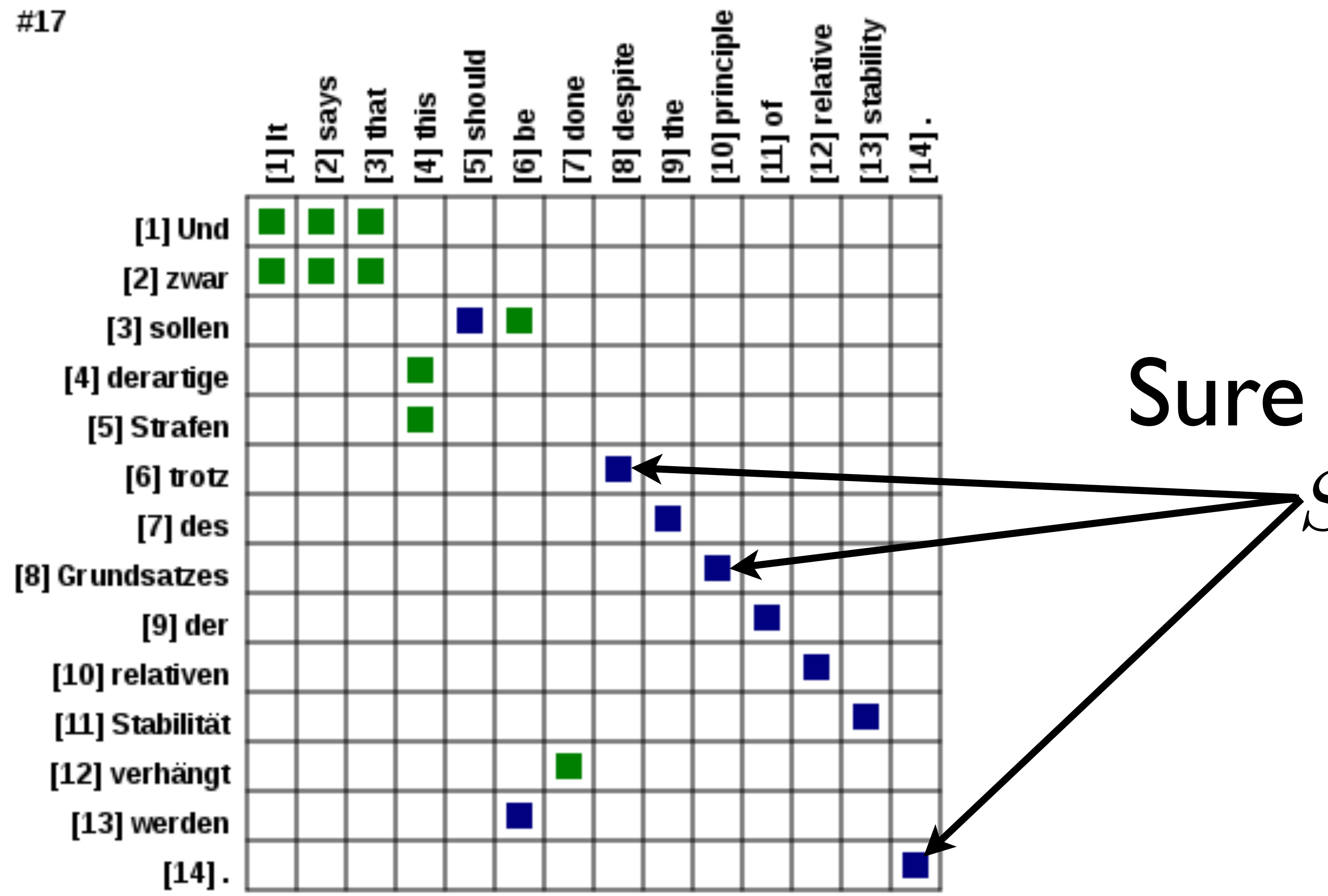
Possible links

P



Alignment Error Rate

Possible links
P

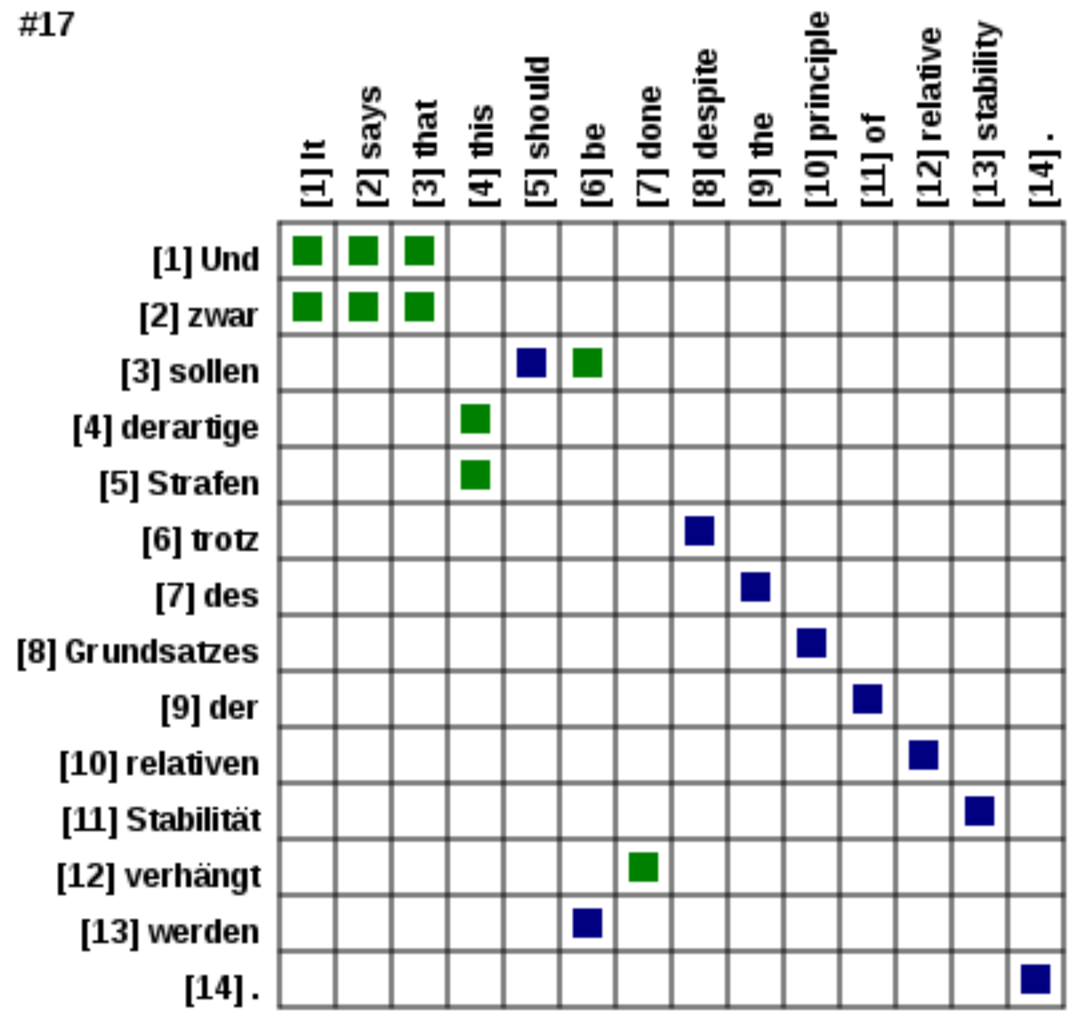


Sure links
S

Alignment Error Rate

Possible links

P



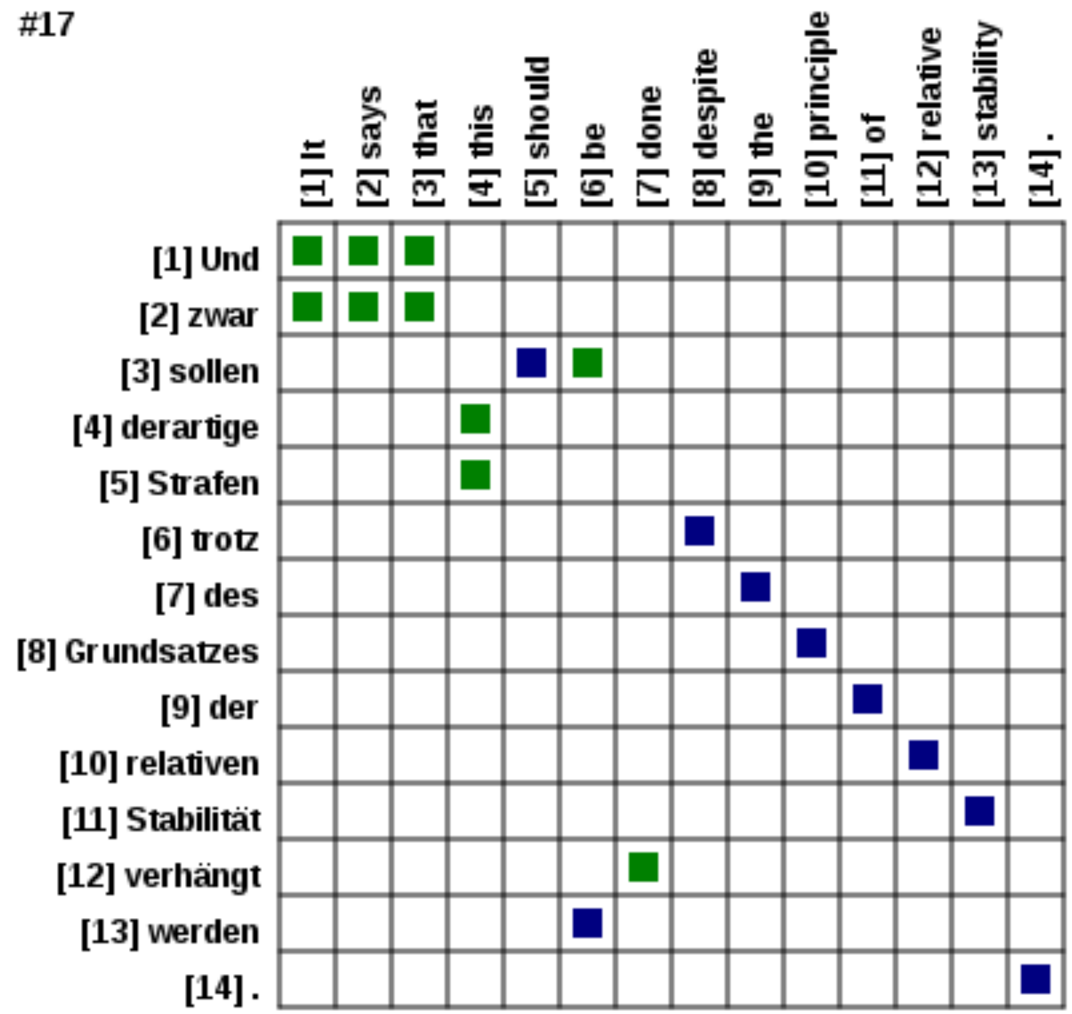
Sure links

S

Alignment Error Rate

Possible links

P



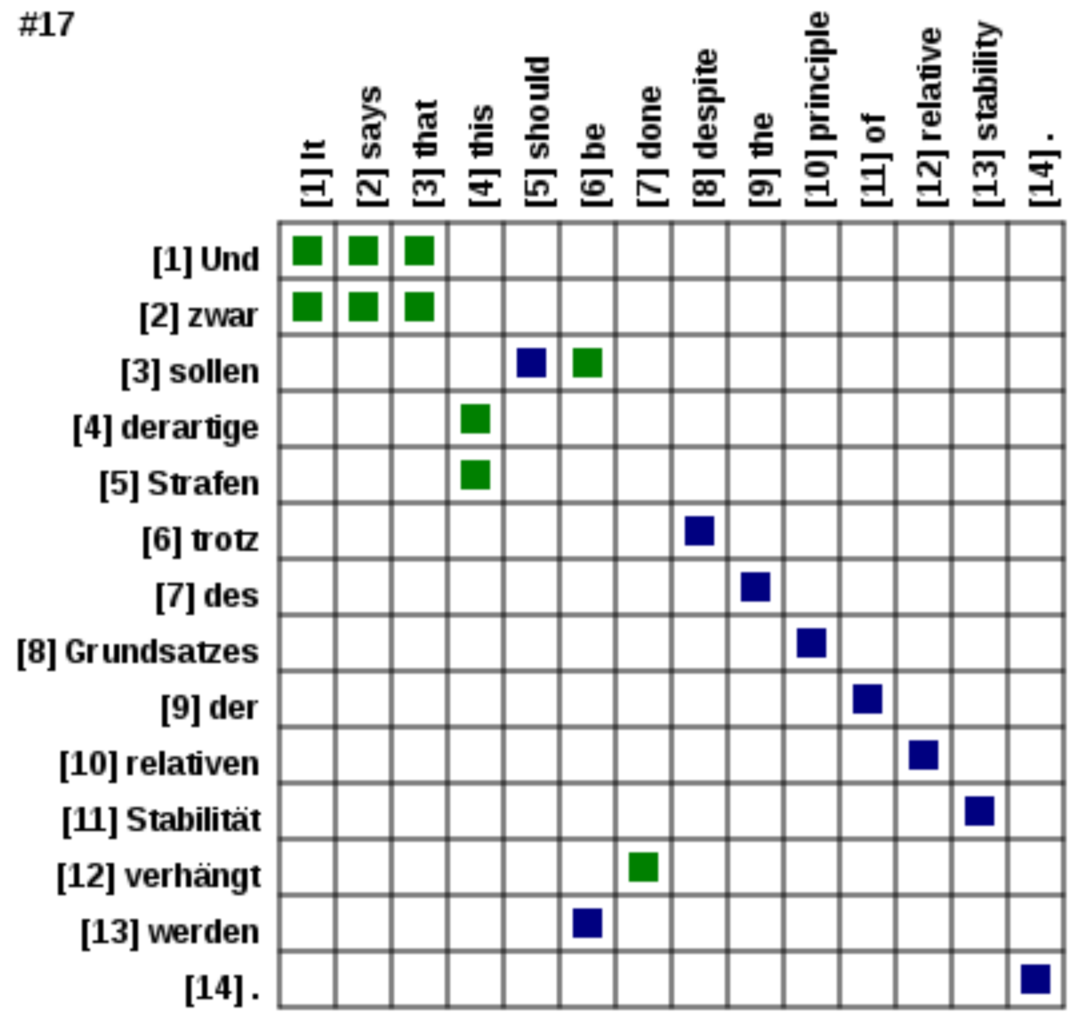
Sure links

S

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

Alignment Error Rate

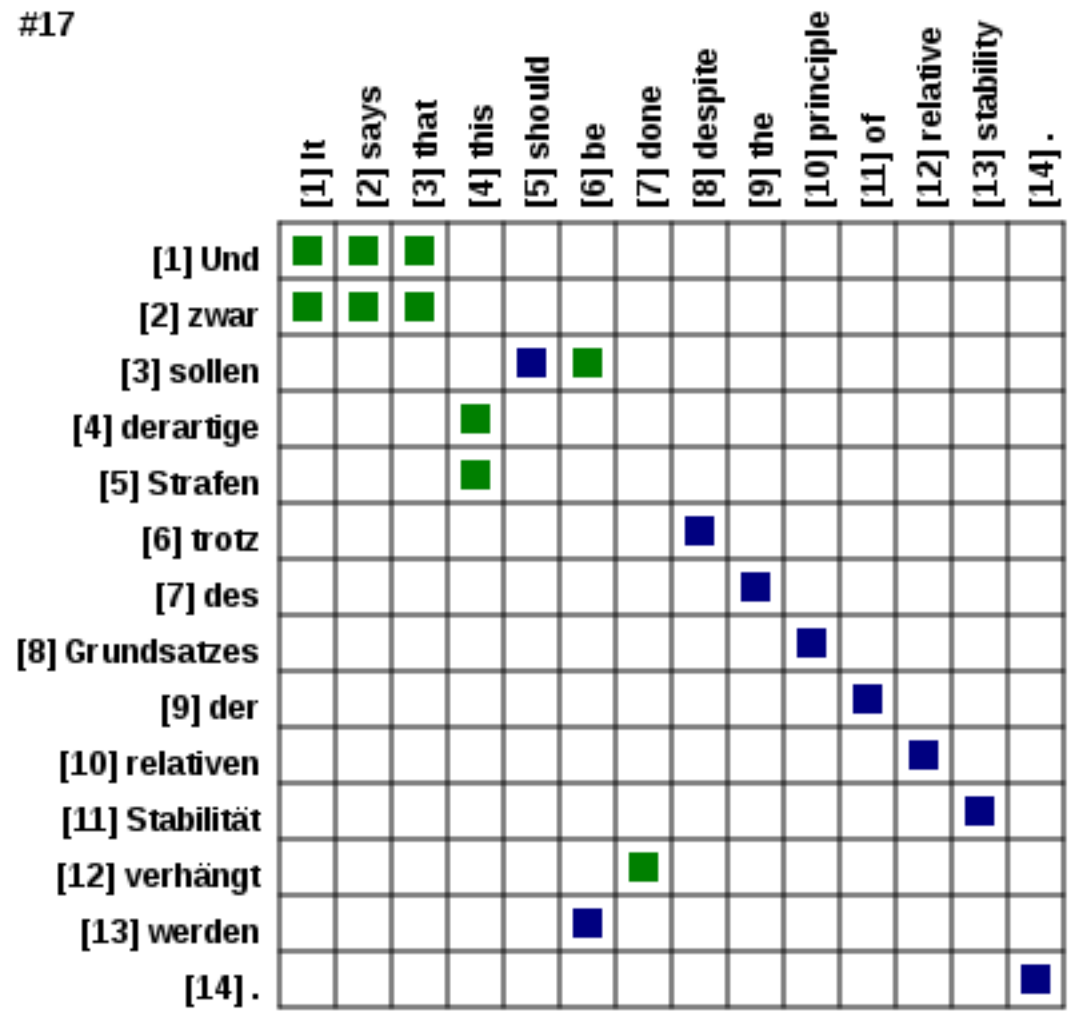
Possible links
P



Sure links
S

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

Alignment Error Rate



Possible links

P

Sure links

S

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

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

Announcements

- First language-in-10 start next week
  - Tuesday, Jan 29: David - Latin
  - Thursday, Jan 31: Weston - Mandarin
- HW 1 is now available (due Feb. 12)

HOMEWORK 1

Due 11:59pm on Tuesday, Feb. 12, 2013

Word alignment is a fundamental task in statistical machine translation. This homework will give you an opportunity to try your hand at developing solutions to this challenging and interesting problem.

Getting started

Go to your clone of your course GitHub repository on the machine where you will be doing this assignment, and run the following command to obain the code and data you will need:

```
./tools/get-new-assignments
```

You will obtain a very simple heuristic aligner written in Python and 100,000 German-English parallel sentences from the Europarl corpus, version 7. The heuristic aligner uses ***set similarity*** to determine which words are aligned to each other in a corpus of parallel sentences. The intuition is that if you look at the set of sentence pairs that contain an English word x, and that set is similar to the set of sentence pairs that contain a German word y, then these words are likely to be translations of each other. The set similarity measure we use is Dice's coefficient, defined in terms of sets X and Y as follows:

$$D(X,Y) = \frac{2 \times |X \cap Y|}{|X| + |Y|}$$

Dice's coefficient ranges in value from 0 to 1.

In our formulation, every pair of words (e,g) in the parallel corpus receives a Dice "score" $\delta(e,g)$. The aligner goes through all pairs of sentences and aligns English word e_i to German word g_j if $\delta(e_i,g_j) > \tau$. By making τ closer to 1, fewer (hopefully, higher precision) points are aligned; by making it closer to 0, more points are aligned. By default, our aligner uses $\tau = 0.5$ as its threshold.

Run the baseline heuristic model 1,000 sentences using the command:

```
./align -n 1000 | ./check > dice.al
```