Algorithms for NLP

Machine Translation III

Yulia Tsvetkov – CMU

Slides: Philipp Koehn – JHU; Chris Dyer – DeepMind
Taylor Berg-Kirkpatrick – CMU, Dan Klein – UC Berkeley

Centauri-Arcturan Parallel Text

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 .
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 .

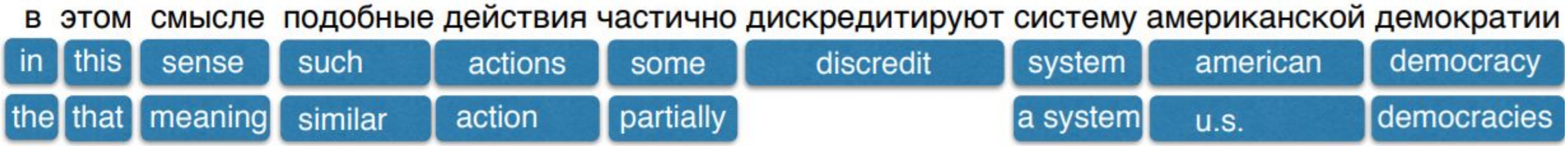
11a. lalok nok crrrok hihok yorok zanzanok .
11b. wat nnat arrat mat zanzanat .

12a. lalok rarok nok izok hihok mok .
12b. wat nnat forat arrat vat gat .

Translation challenge: **farok crrrok hihok yorok clok kantok ok-yurp**

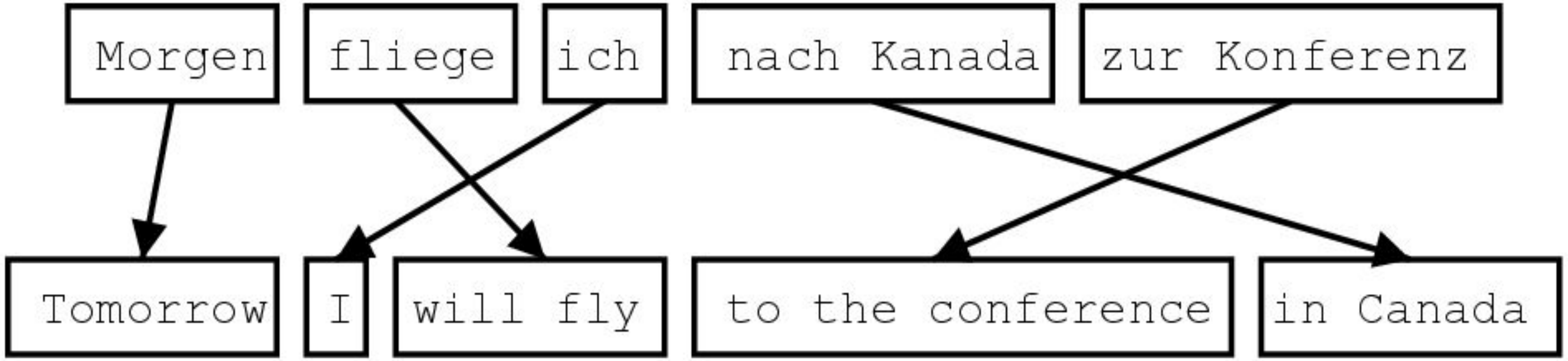
(from Knight (1997): Automating Knowledge Acquisition for Machine Translation)

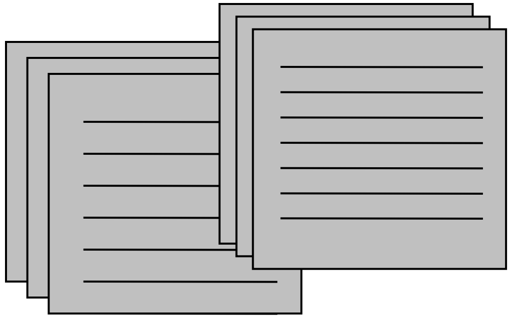
Lexical Translation



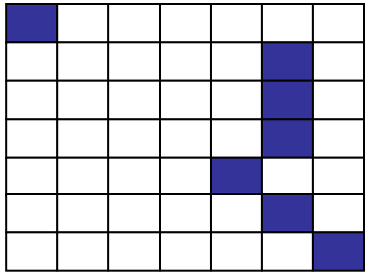
IBM Model 1	lexical translation
IBM Model 2	adds absolute reordering model
IBM Model 3	adds fertility model
IBM Model 4	relative reordering model
IBM Model 5	fixes deficiency

Phrase-Based System Overview





Sentence-aligned corpus



Word alignments

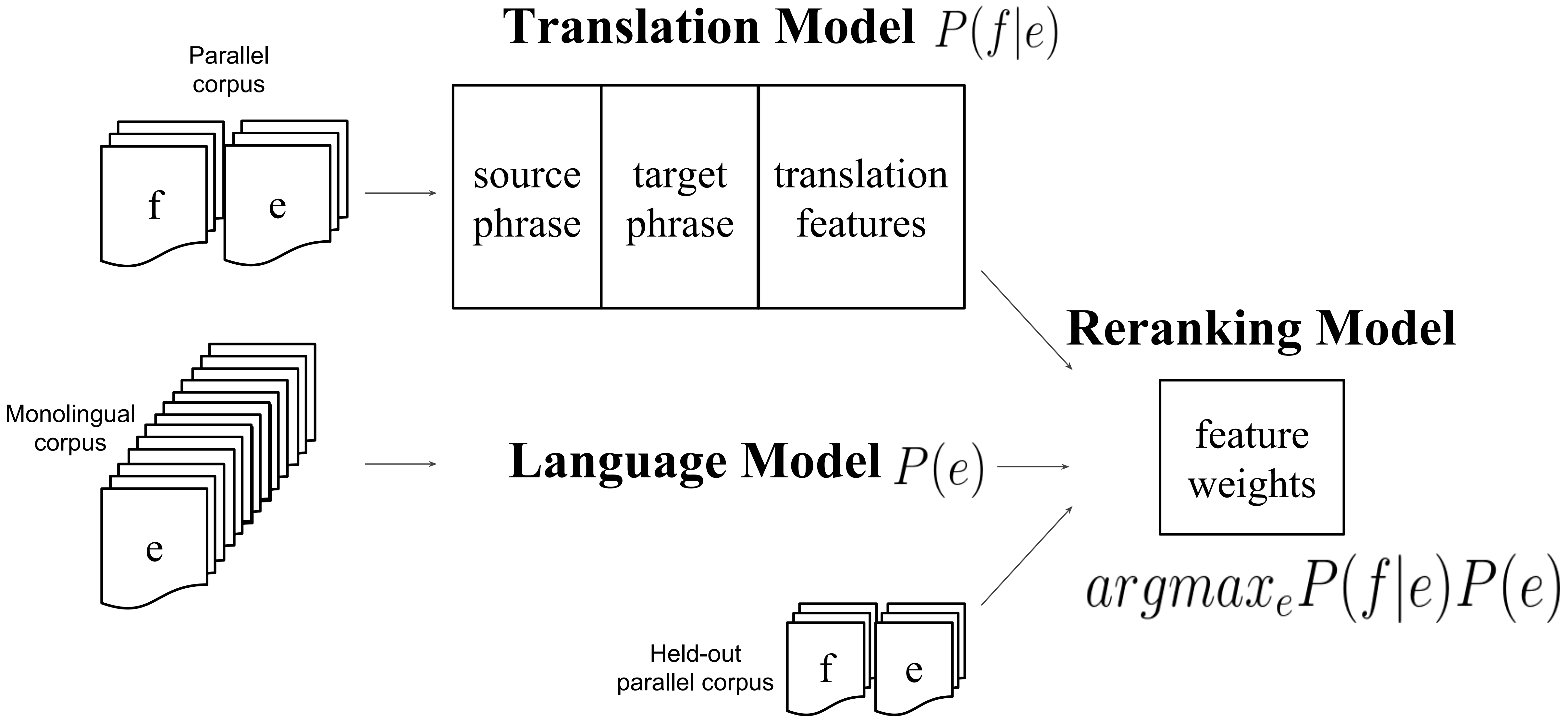
cat ||| chat ||| 0.9
the cat ||| le chat ||| 0.8
dog ||| chien ||| 0.8
house ||| maison ||| 0.6
my house ||| ma maison ||| 0.9
language ||| langue ||| 0.9
…

Phrase table (translation model)

Phrase-Based Translation

в	этом	смысле	подобные	действия	частично	дискредитируют	систему	американской	демократии
in	this	sense	such	actions	some	discredit	system	american	democracy
the	that	meaning	similar	action	partially		a system	u.s.	democracies
a	the	terms	these	the	part		systems	us	democratic
at	it	way	this	acts	in part		which	america	of democracy
	here	sense ,	like	steps	partly		network	america's	
this			these actions					american democracy	
in this sense								america’s democracy	
in that sense								us democracy	
in this respect									

Noisy Channel Model : Phrase-Based MT



Estimate Translation Probabilities

If we have alignments: Maximum Likelihood Estimation

$$\hat{p}_{\text{MLE}}(e \mid \texttt{Haus}) = \begin{cases} 0.8 & \text{if } e = \text{house}, \\ 0.16 & \text{if } e = \text{building}, \\ 0.02 & \text{if } e = \text{home}, \\ 0.015 & \text{if } e = \text{household}, \\ 0.005 & \text{if } e = \text{shell}. \end{cases}$$

Estimate Alignments

If we have translation probabilities:

das

e	$t(e\|f)$
the	0.7
that	0.15
which	0.075
who	0.05
this	0.025

Haus

e	$t(e\|f)$
house	0.8
building	0.16
home	0.02
household	0.015
shell	0.005

ist

e	$t(e\|f)$
is	0.8
's	0.16
exists	0.02
has	0.015
are	0.005

klein

e	$t(e\|f)$
small	0.4
little	0.4
short	0.1
minor	0.06
petty	0.04

We can estimate Viterbi alignment

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

Finding the Viterbi Alignment

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

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

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

In model 1:

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

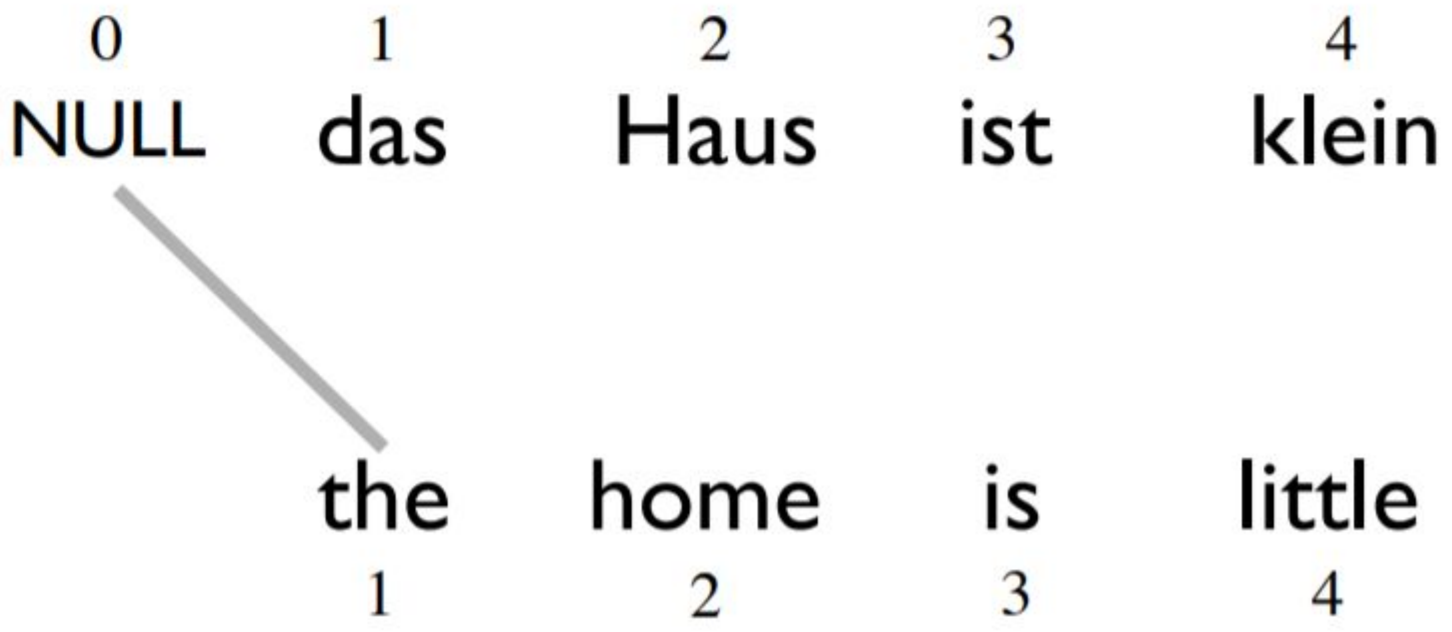
$$= \arg \max_{a_i=0}^{n} p(e_i \mid f_{a_i})$$

Finding the Viterbi Alignment

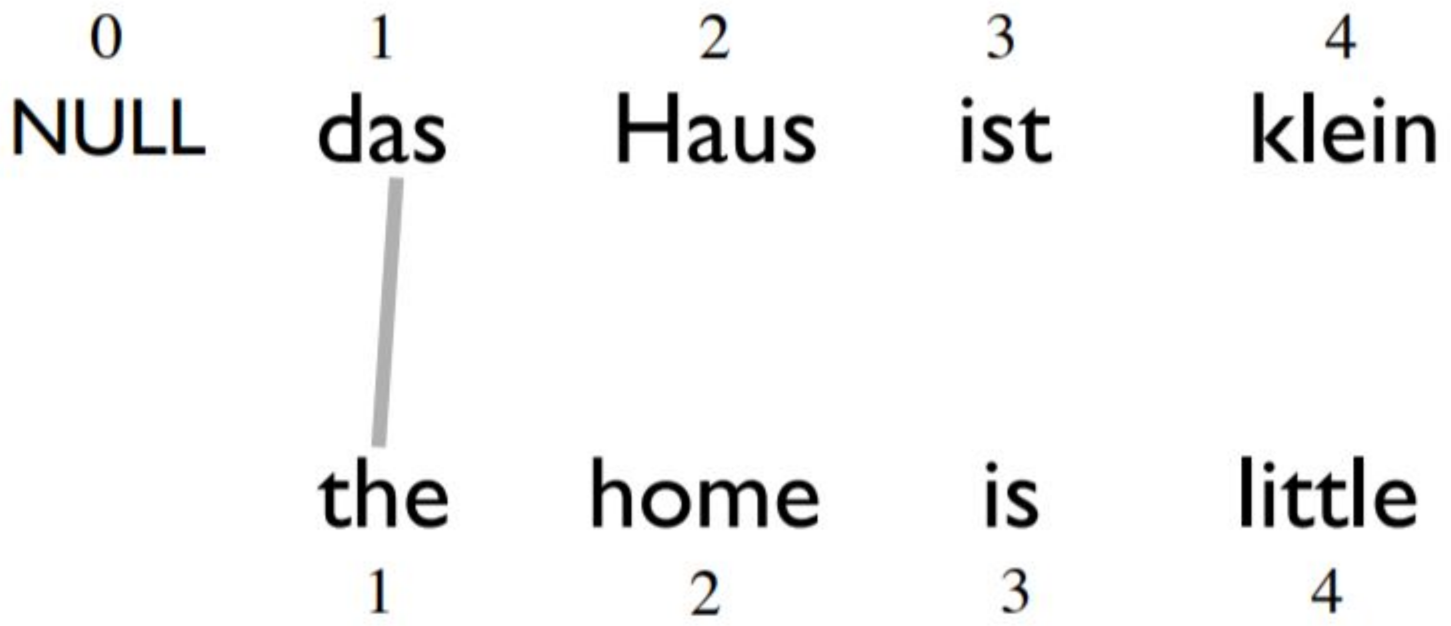
0	1	2	3	4
NULL	das	Haus	ist	klein

the	home	is	little
1	2	3	4

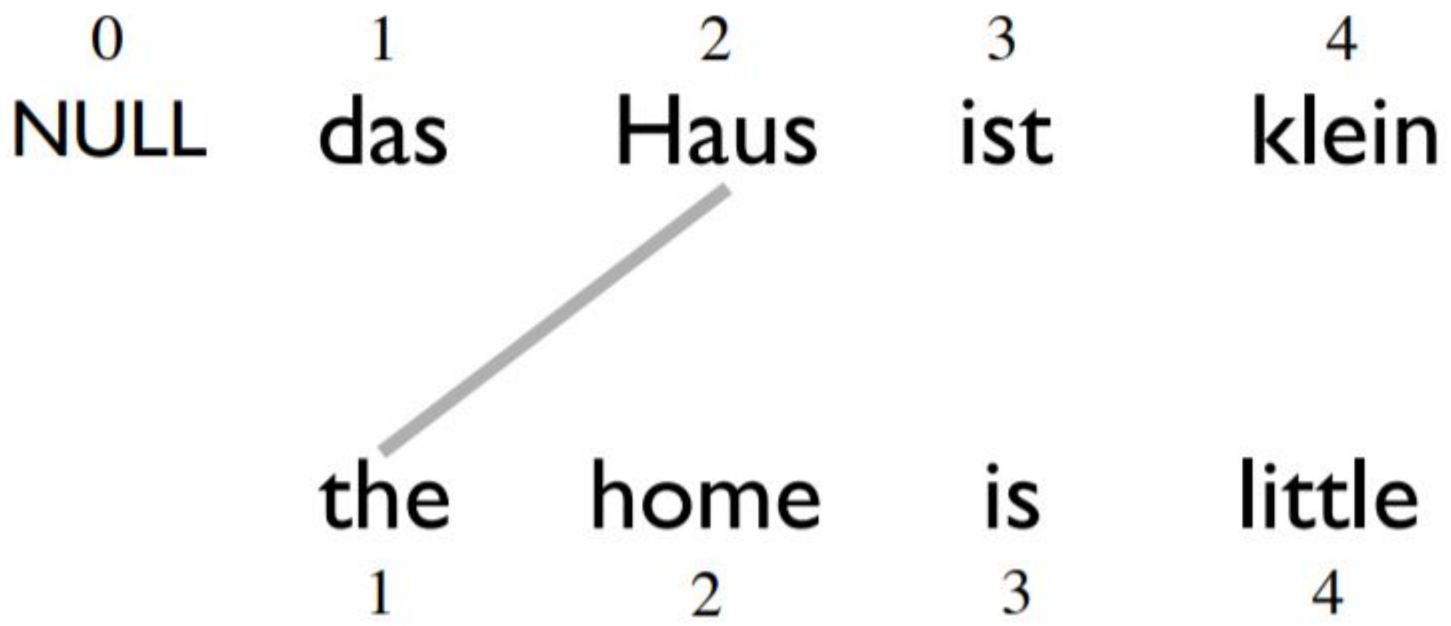
Finding the Viterbi Alignment



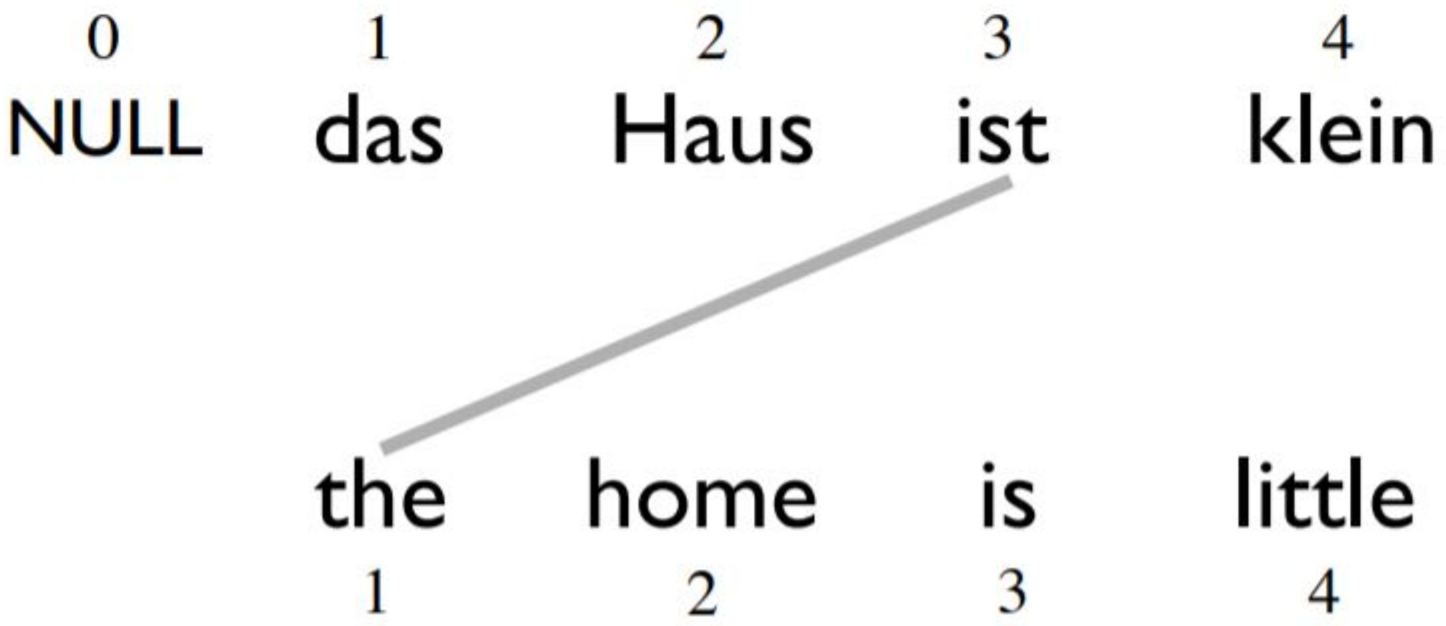
Finding the Viterbi Alignment



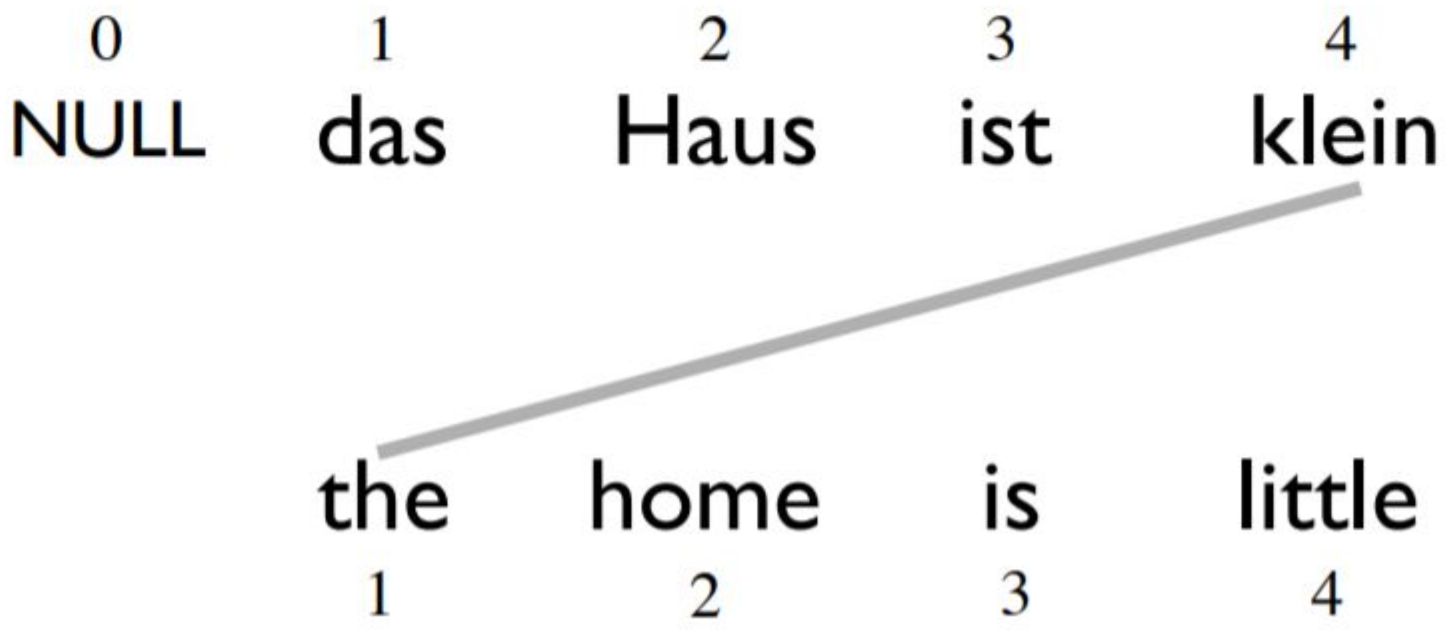
Finding the Viterbi Alignment



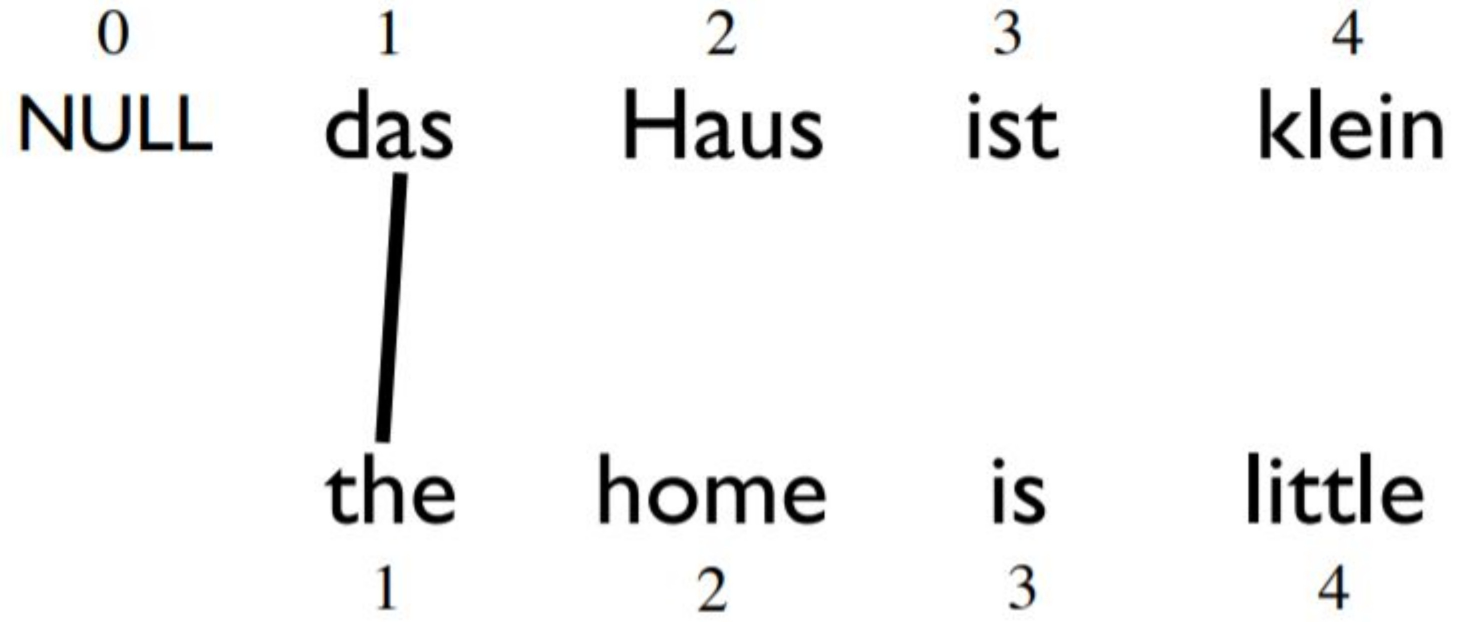
Finding the Viterbi Alignment



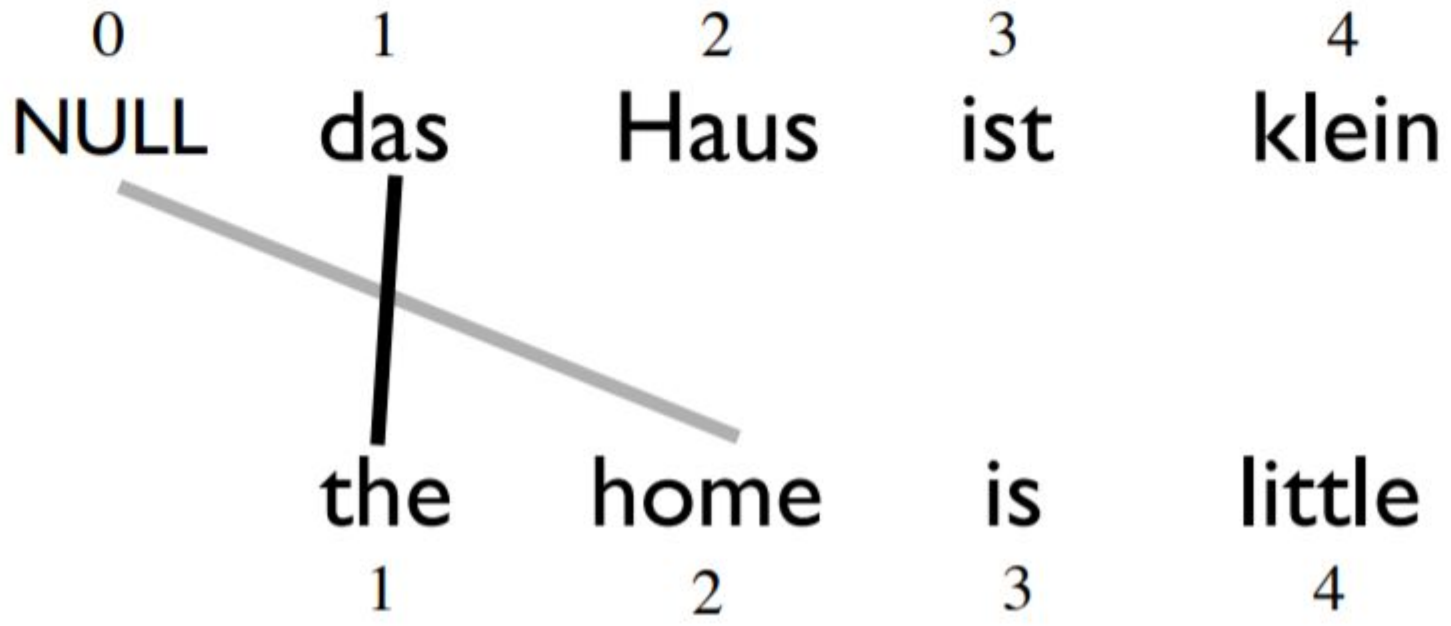
Finding the Viterbi Alignment



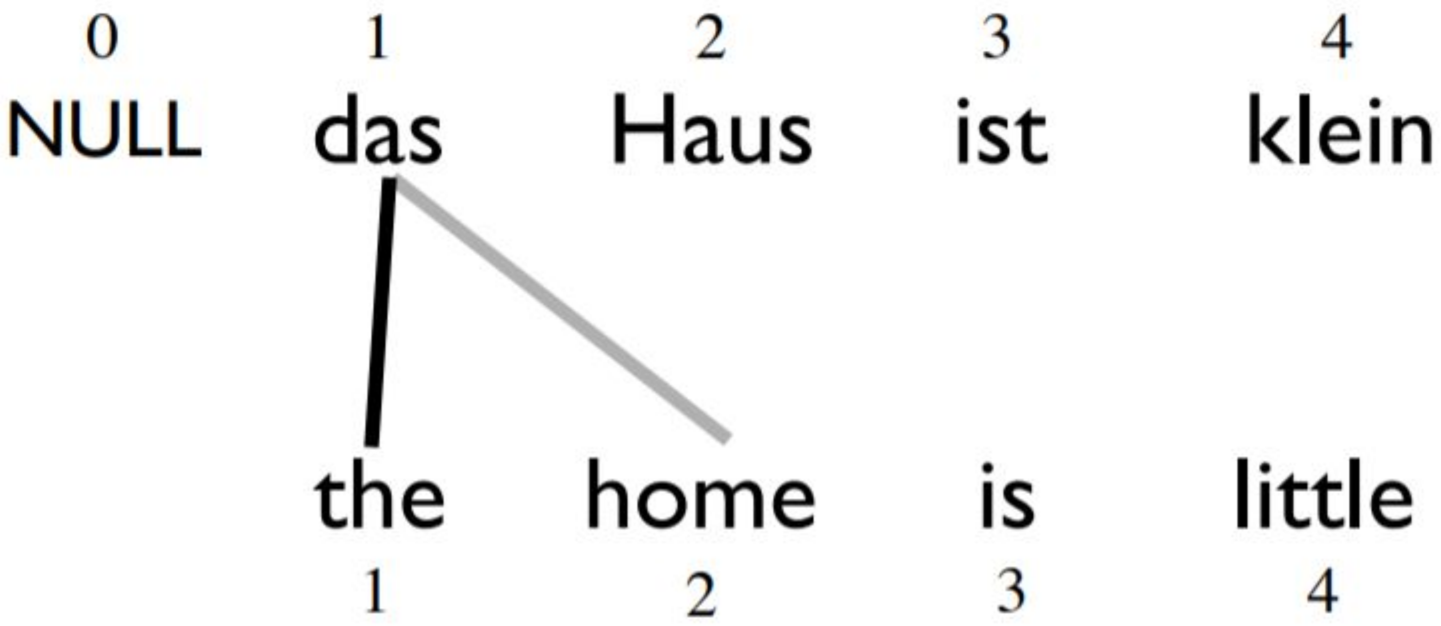
Finding the Viterbi Alignment



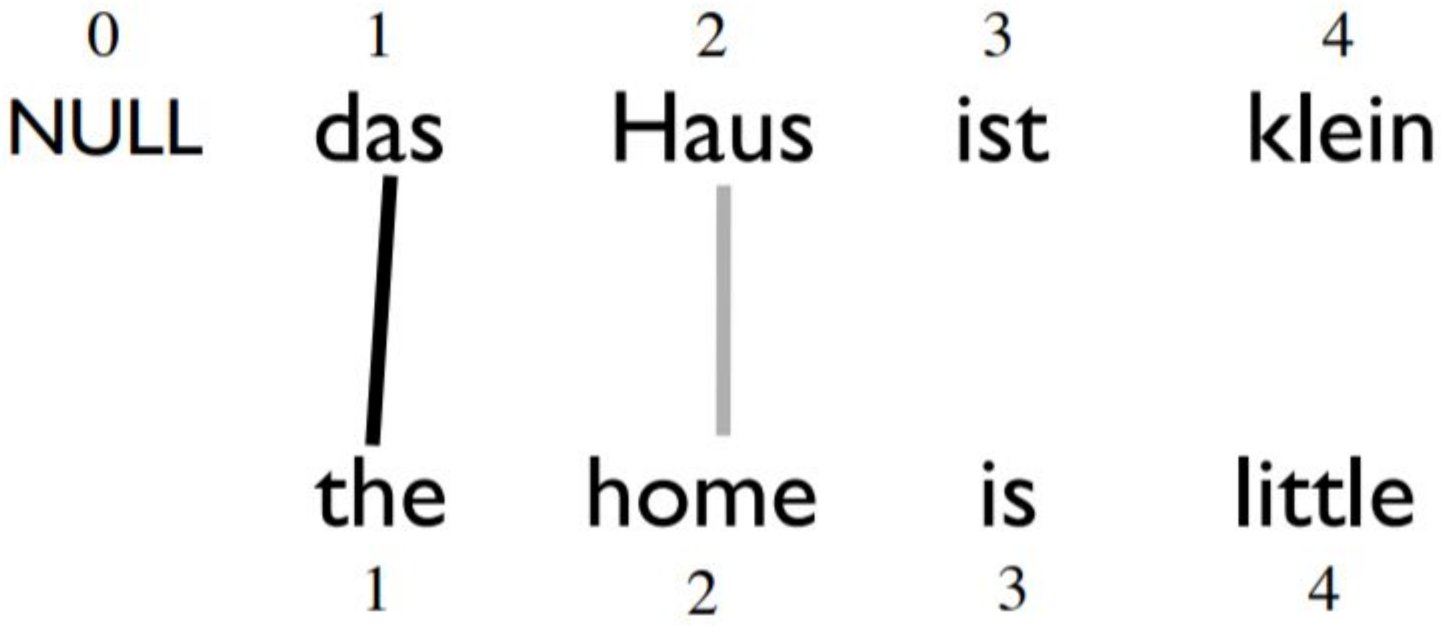
Finding the Viterbi Alignment



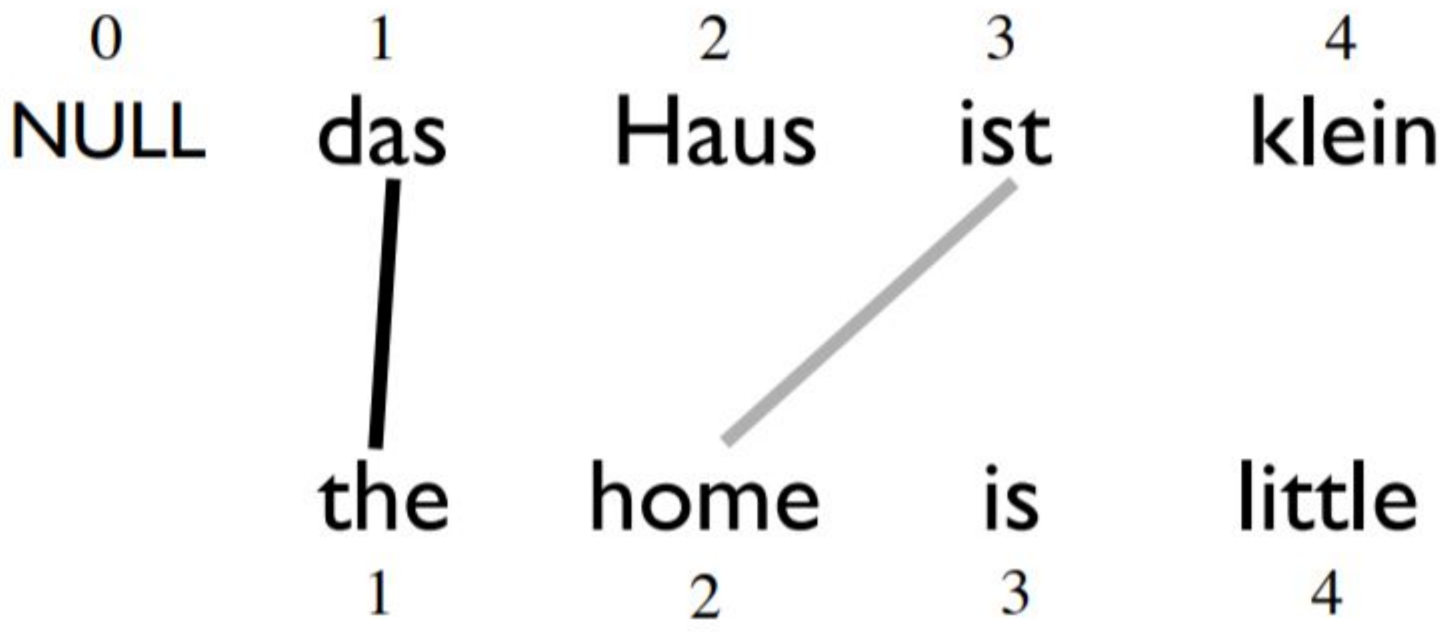
Finding the Viterbi Alignment



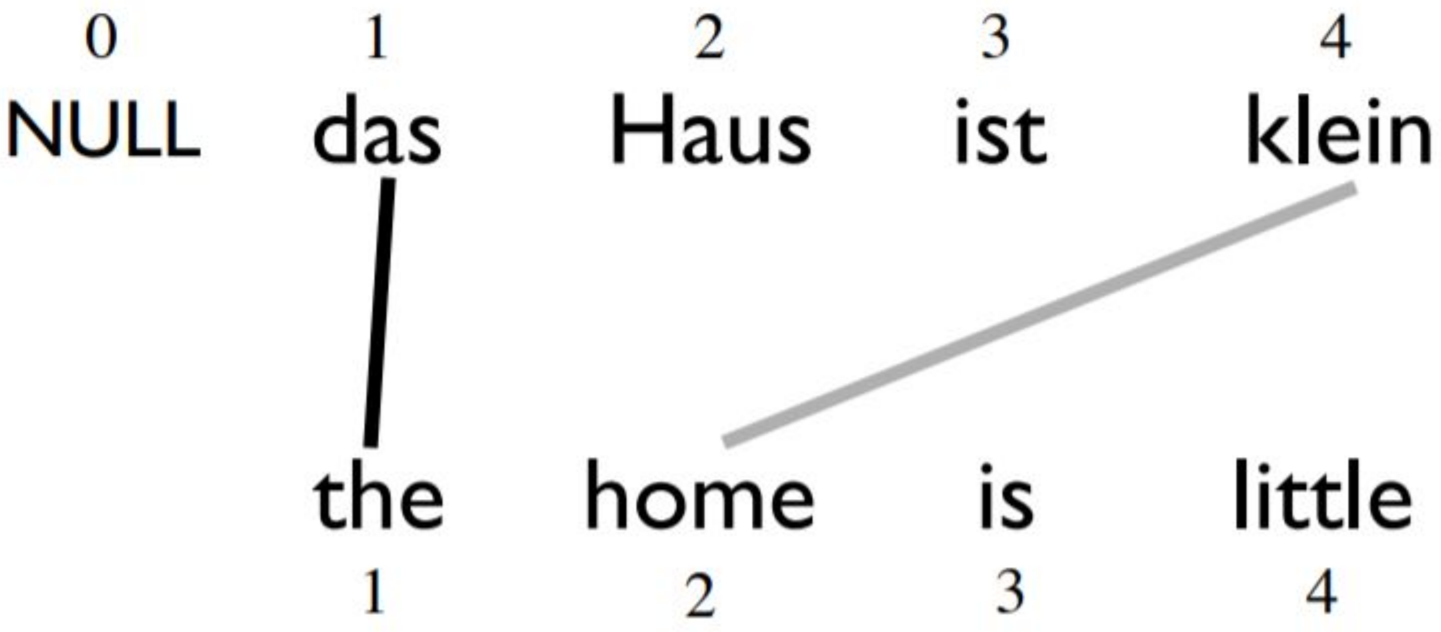
Finding the Viterbi Alignment



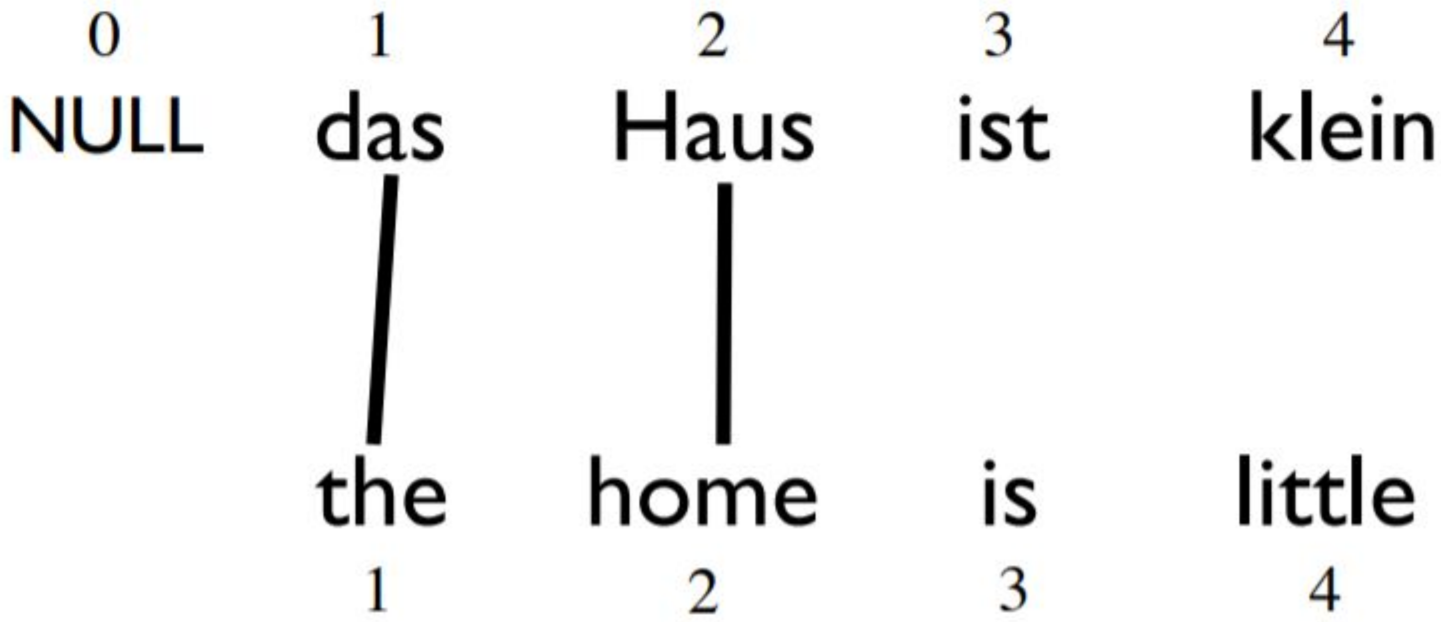
Finding the Viterbi Alignment



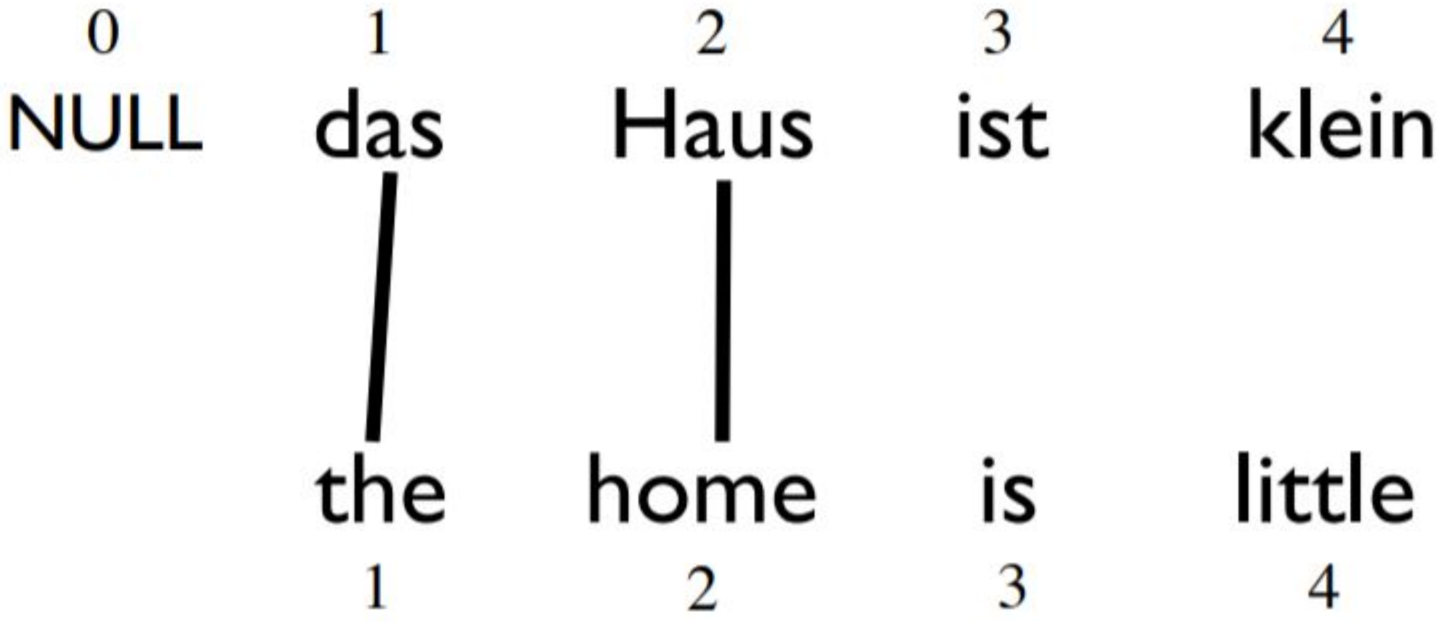
Finding the Viterbi Alignment



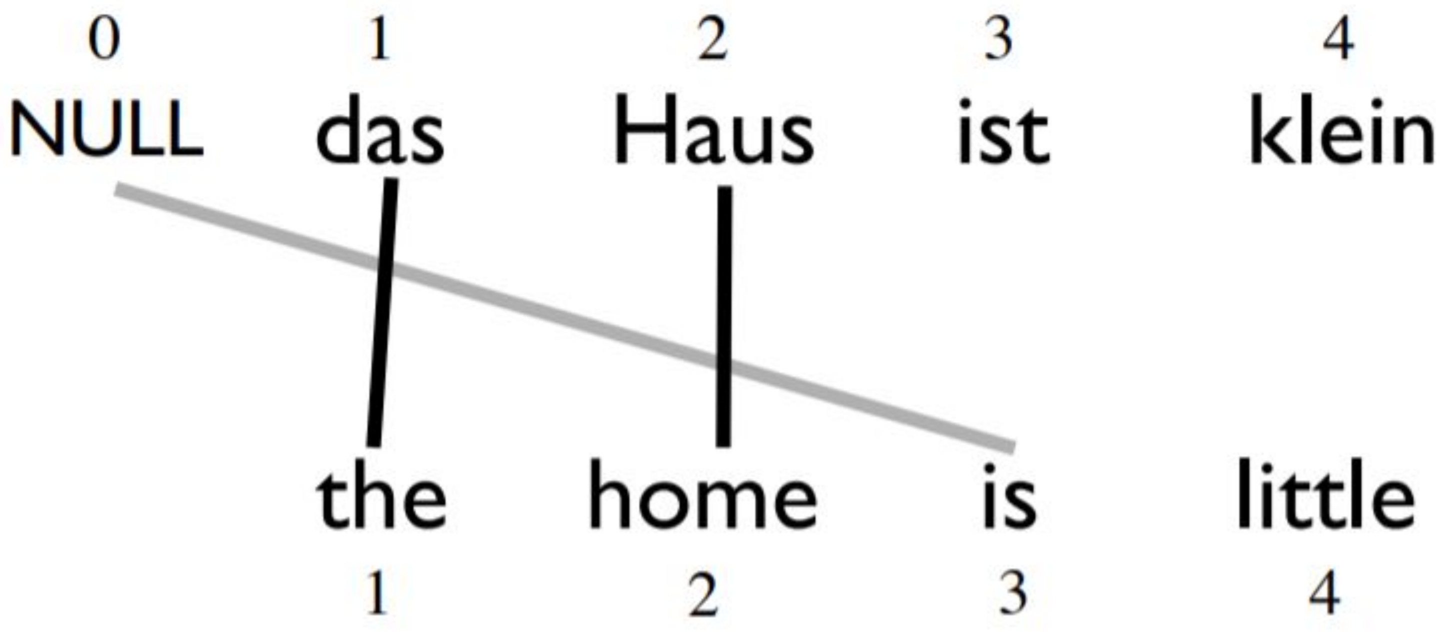
Finding the Viterbi Alignment



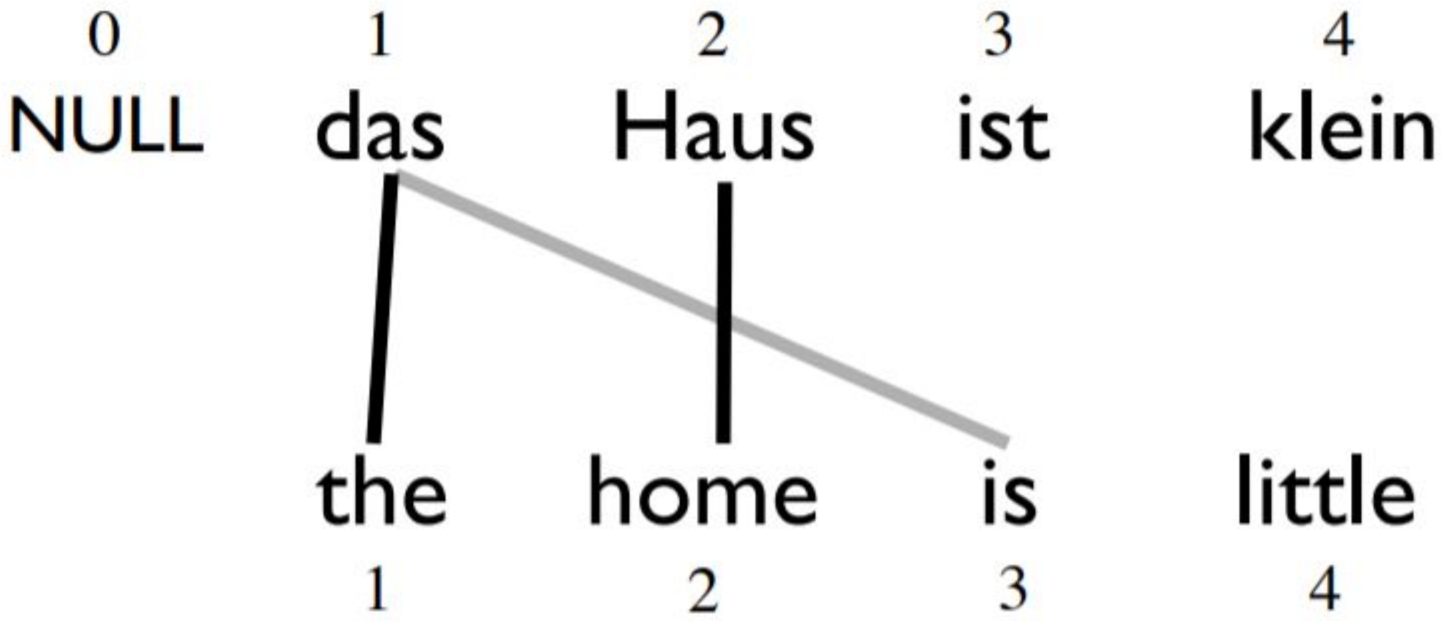
Finding the Viterbi Alignment



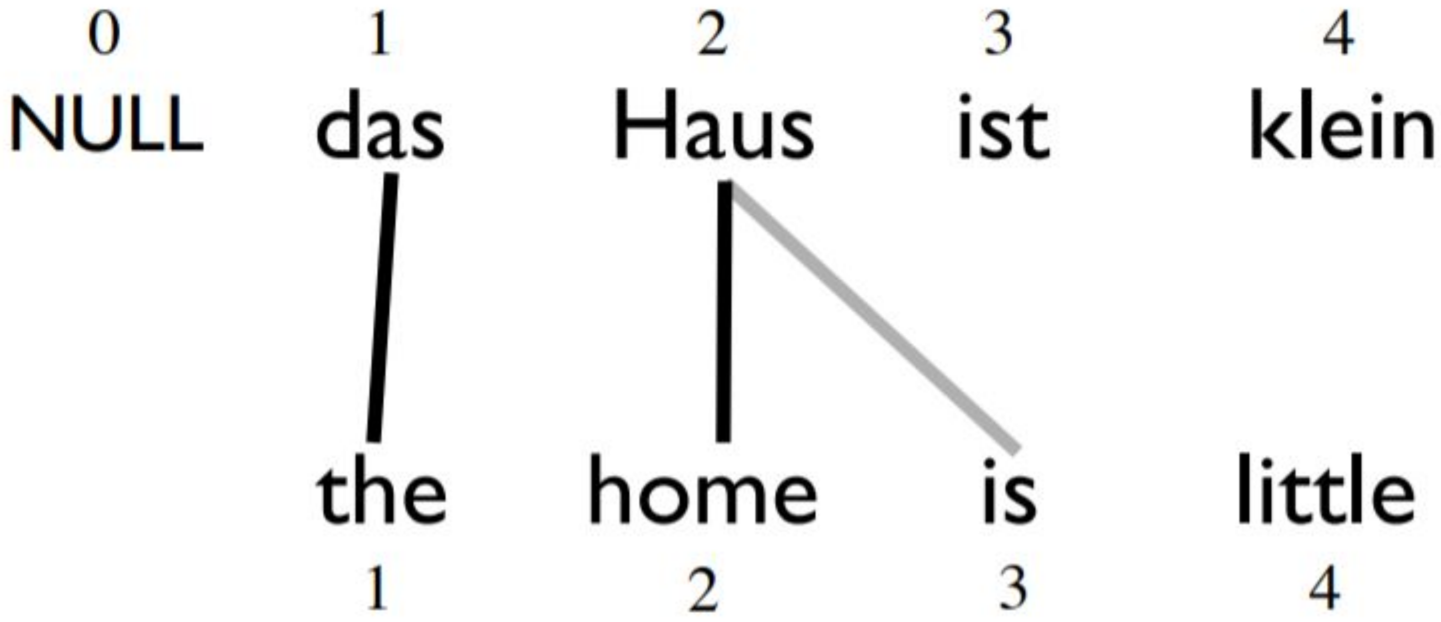
Finding the Viterbi Alignment



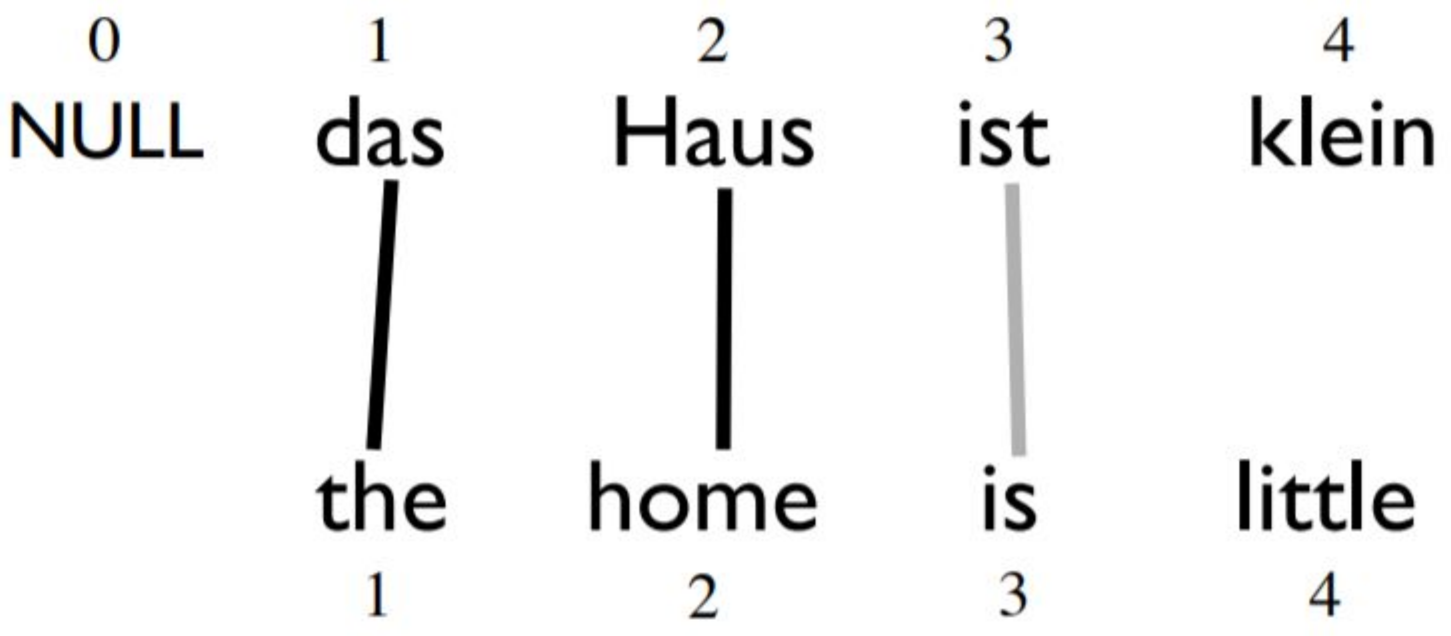
Finding the Viterbi Alignment



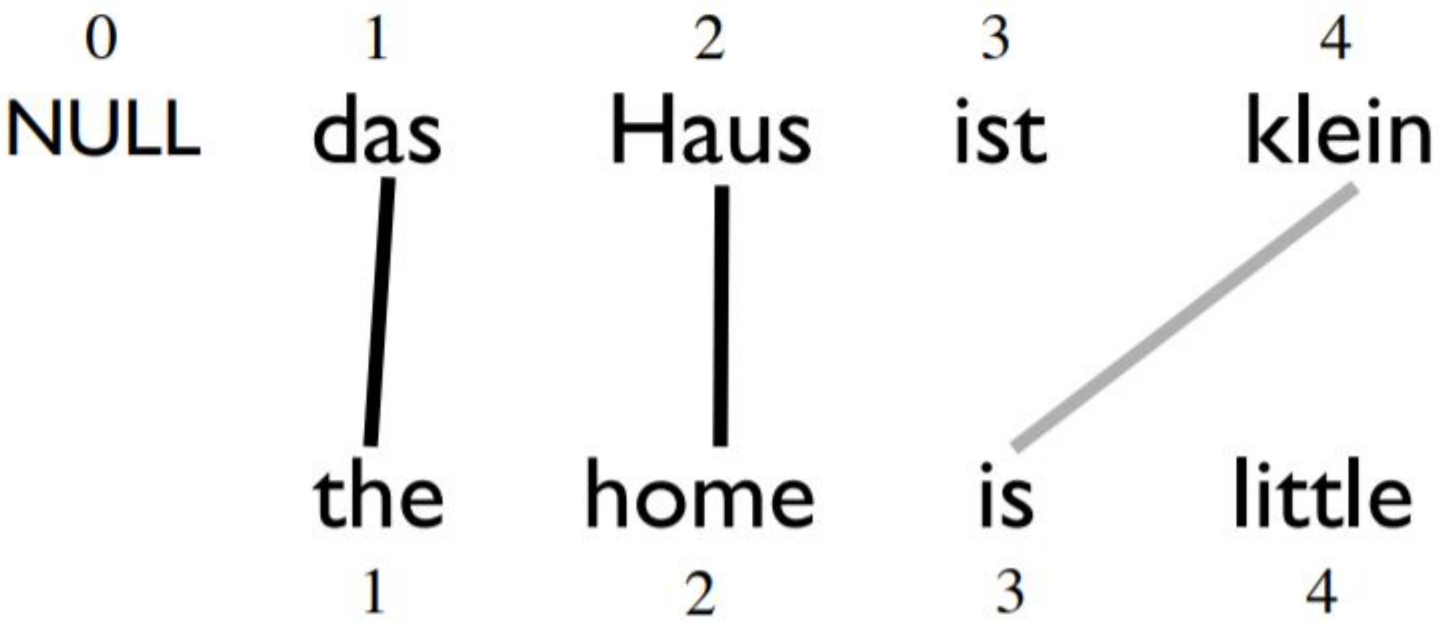
Finding the Viterbi Alignment



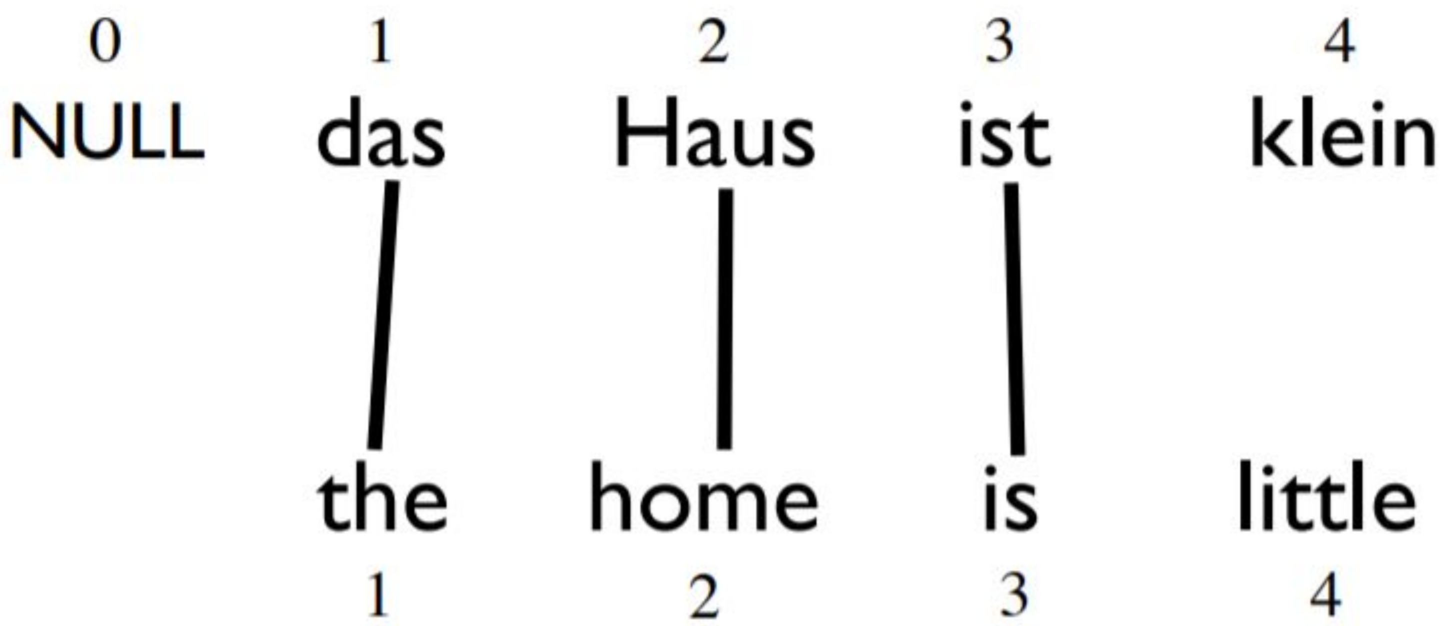
Finding the Viterbi Alignment



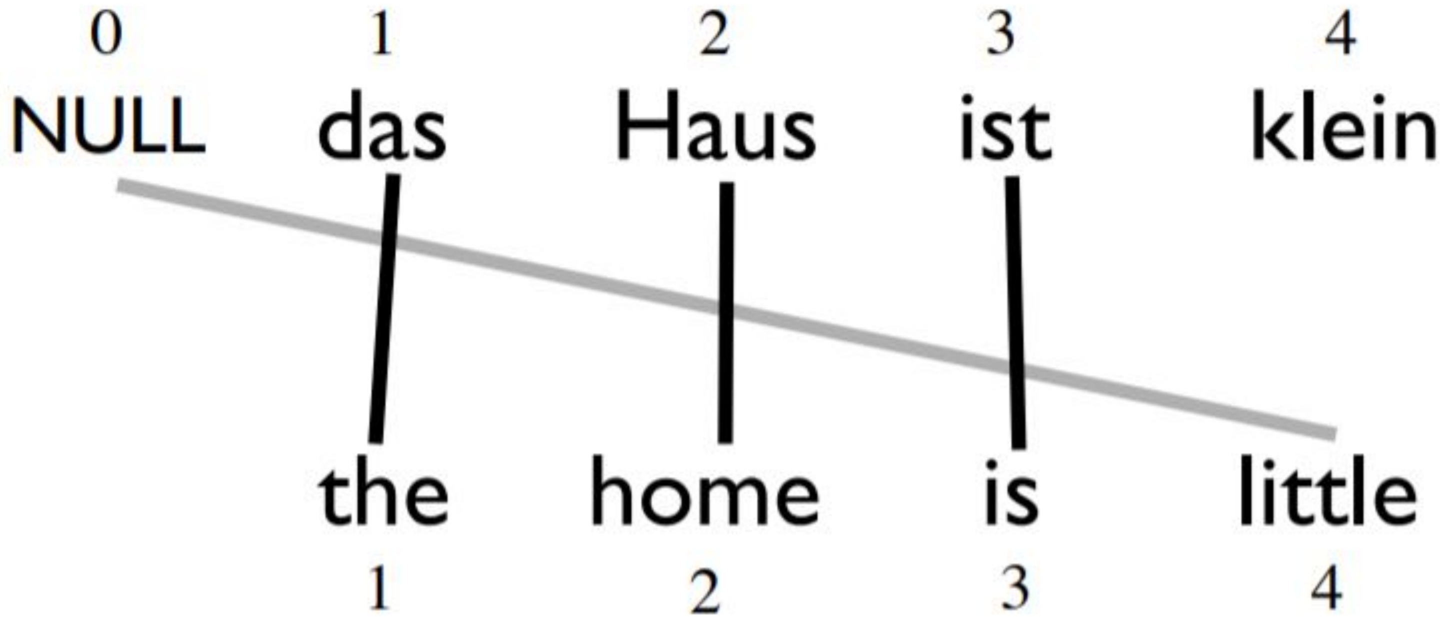
Finding the Viterbi Alignment



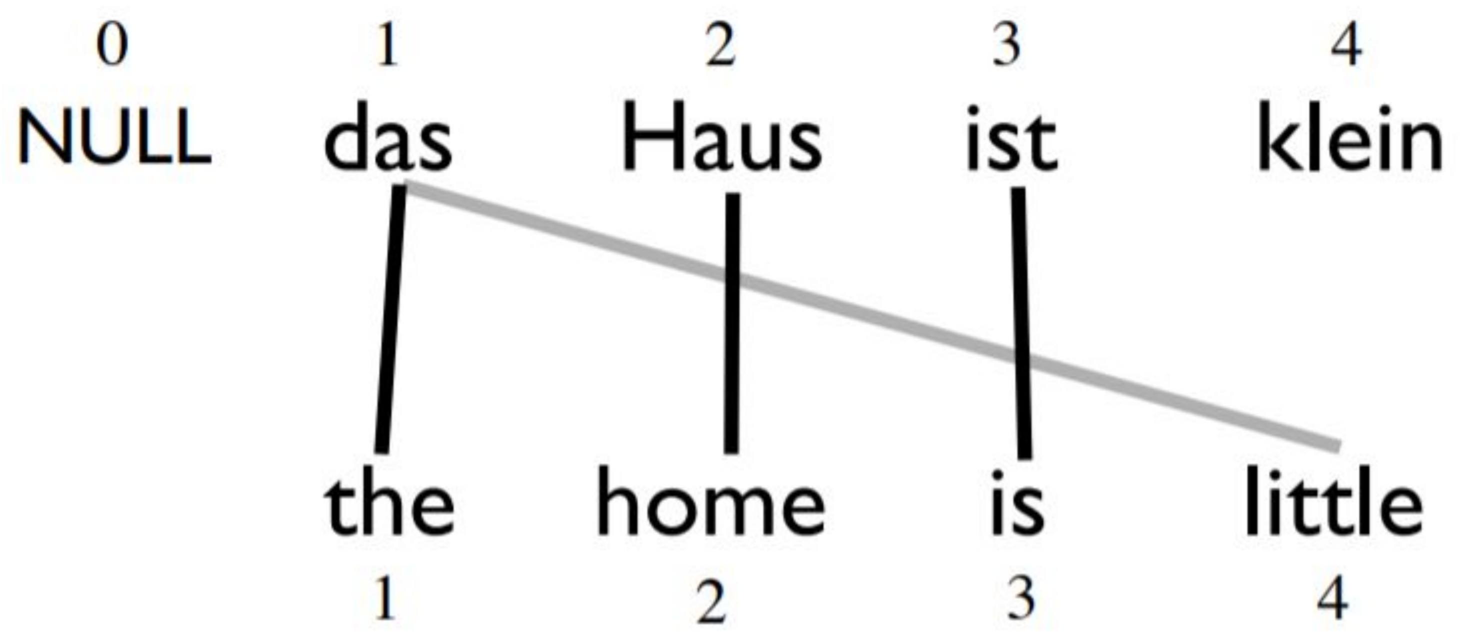
Finding the Viterbi Alignment



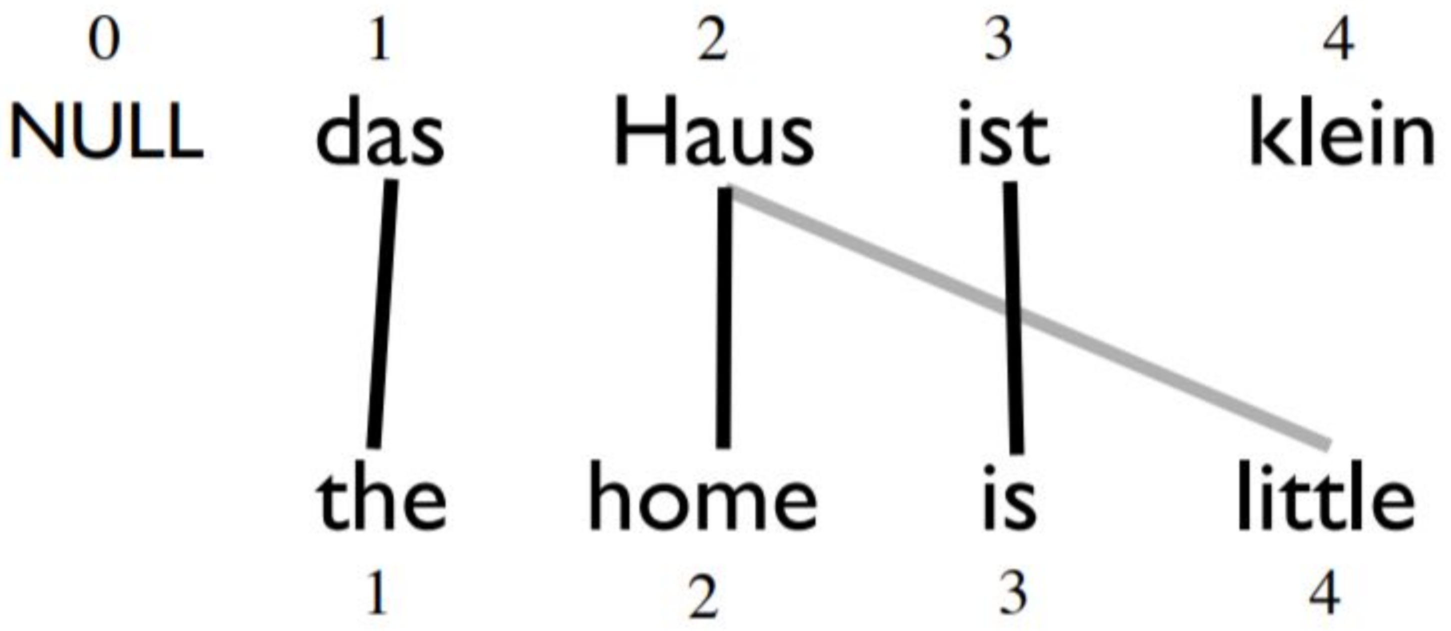
Finding the Viterbi Alignment



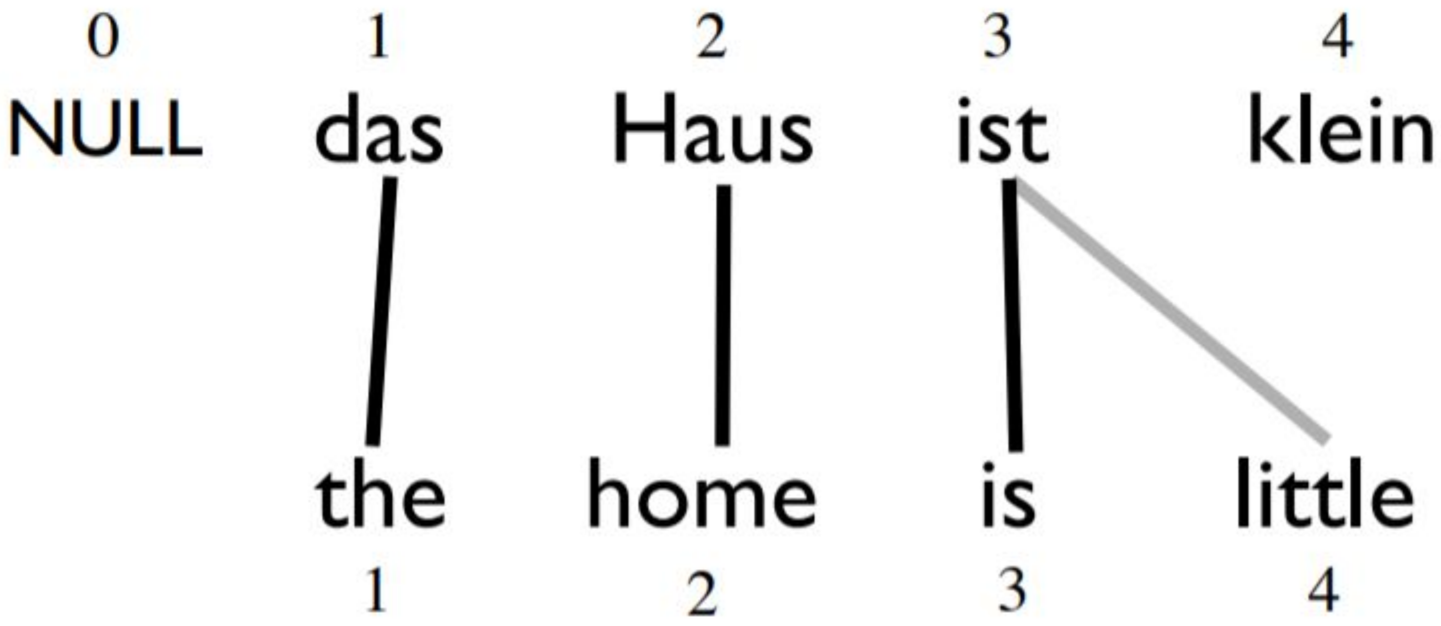
Finding the Viterbi Alignment



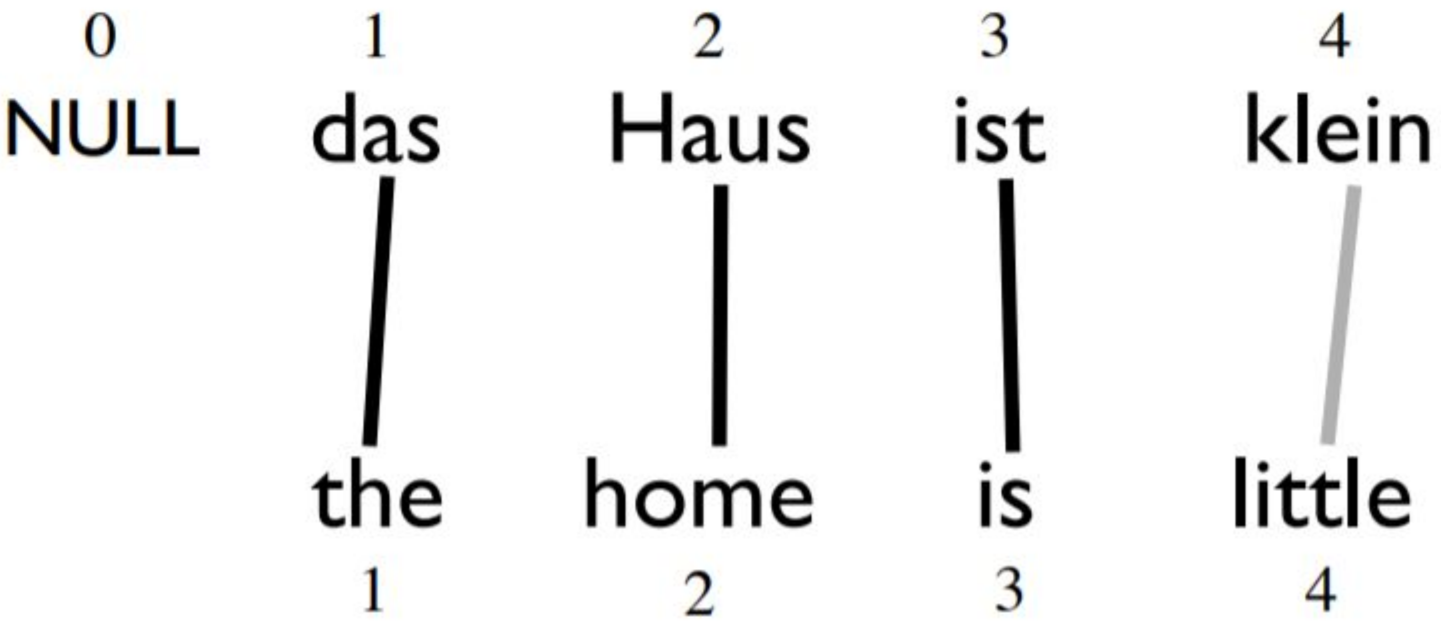
Finding the Viterbi Alignment



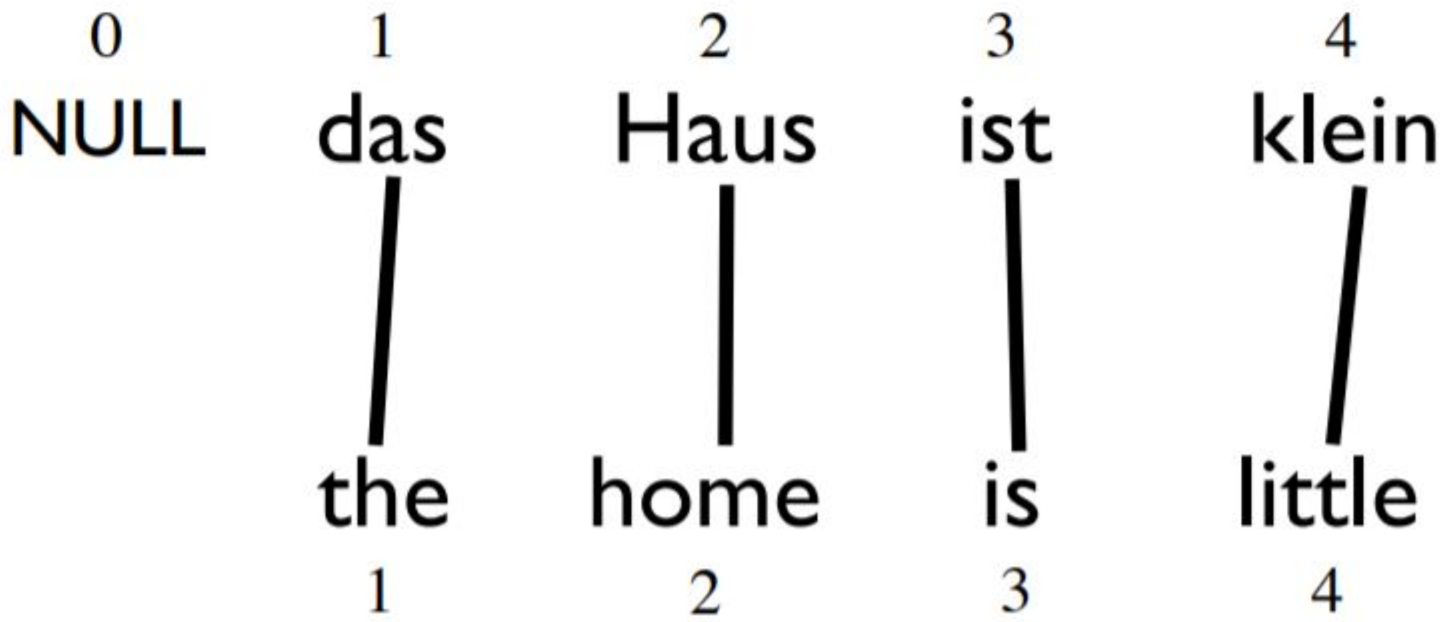
Finding the Viterbi Alignment



Finding the Viterbi Alignment



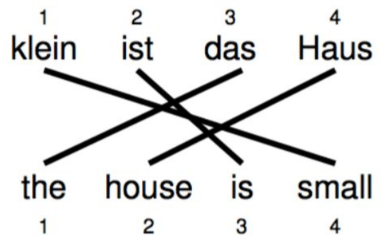
Finding the Viterbi Alignment



Learning Lexical Translation Models

We would like to estimate the lexical translation probabilities *t(e|f)* from a parallel corpus but we do not have the alignments

- Chicken and egg problem



  - if we had the alignments,

    → we could estimate the parameters of our generative model (MLE)

  - if we had the parameters,

    → we could estimate the alignments

klein

e	$t(e\|f)$
small	0.4
little	0.4
short	0.1
minor	0.06
petty	0.04

EM Algorithm

- Incomplete data
  - if we had complete data, would could estimate the model
  - if we had the model, we could fill in the gaps in the data

- Expectation Maximization (EM) in a nutshell

1. initialize model parameters (e.g. uniform, random)
2. assign probabilities to the missing data
3. estimate model parameters from complete data
4. iterate steps 2–3 until convergence

EM for Model 1

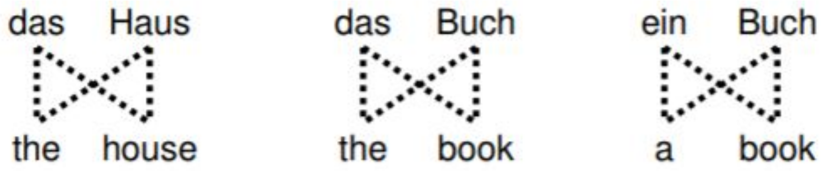
- initialize model parameters, e.g. uniform:

e	f	initial
the	das	0.25
book	das	0.25
house	das	0.25
the	buch	0.25
book	buch	0.25
a	buch	0.25
book	ein	0.25
a	ein	0.25
the	haus	0.25
house	haus	0.25

EM for Model 1

- initialize model parameters, e.g. uniform:
- repeat until convergence:
  - compute "expected" alignments

e	f	initial
the	das	0.25
book	das	0.25
house	das	0.25
the	buch	0.25



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

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

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

see simplification trick in the previous lecture

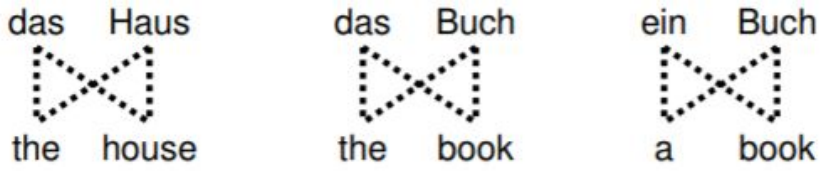
EM for Model 1

- initialize model parameters, e.g. uniform:

e	f	initial
the	das	0.25
book	das	0.25
house	das	0.25
the	buch	0.25

- repeat until convergence:
  - compute "expected" alignments

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



  - keep track of the expected number of times *f* translates into *e* throughout the whole corpus

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

```
Initialize c(e|f) = 0 for all e, f in vocab
for every sentence pair e,f:
  for every alignment a do
    for j = 1..len(e) do
       c(e_j|f_a(j)) += p(a|e,f)
```

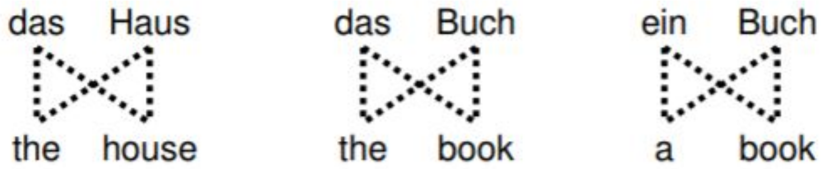
EM for Model 1

- initialize model parameters, e.g. uniform:

e	f	initial
the	das	0.25
book	das	0.25
house	das	0.25
the	buch	0.25

- repeat until convergence:
  - compute "expected" alignments

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



  - keep track of the expected number of times *f* translates into *e* throughout the whole corpus

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

  - apply MLE to estimate new model parameters

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

e	f	initial	1st it.
the	das	0.25	0.5
book	das	0.25	0.25
house	das	0.25	0.25
the	buch	0.25	0.25

EM for Model 1

- initialize model parameters, e.g. uniform:

e	f	initial
the	das	0.25
book	das	0.25
house	das	0.25
the	buch	0.25

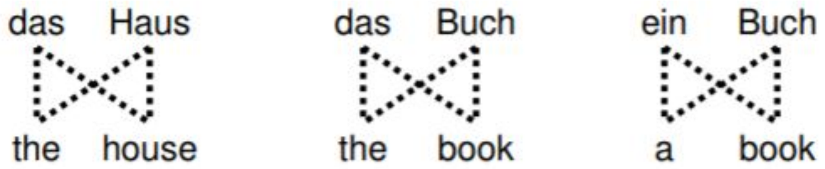
t-table

- repeat until convergence:

E-step

- compute "expected" alignments

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



M-step

- keep track of the expected number of times *f* translates into *e* throughout the whole corpus

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

- apply MLE to estimate new model parameters

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

e	f	initial	1st it.
the	das	0.25	0.5
book	das	0.25	0.25
house	das	0.25	0.25
the	buch	0.25	0.25

IBM Model 1 and EM

t-table **Probabilities**

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

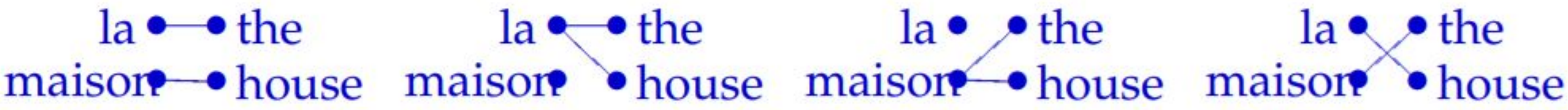
IBM Model 1 and EM

t-table **Probabilities**

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

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

Alignments



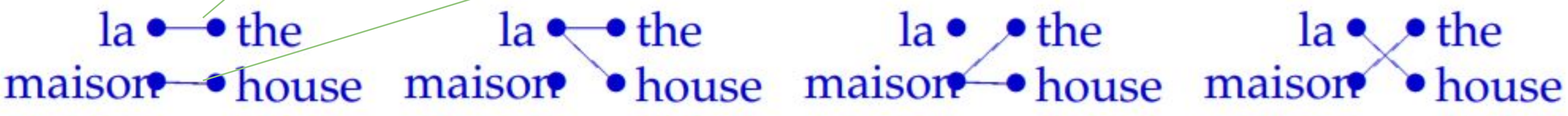
IBM Model 1 and EM

t-table **Probabilities**

$p(\text{the}|\text{la}) = 0.7$ $p(\text{house}|\text{la}) = 0.05$

$p(\text{the}|\text{maison}) = 0.1$ $p(\text{house}|\text{maison}) = 0.8$

Alignments



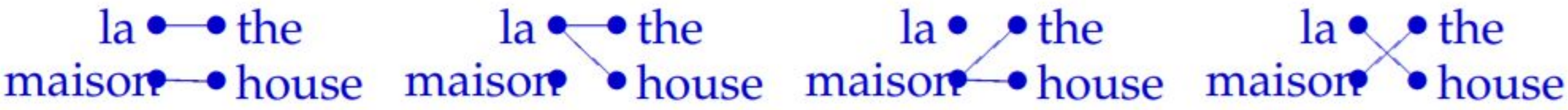
$p(\mathbf{e}, a|\mathbf{f}) = 0.56$ $p(\mathbf{e}, a|\mathbf{f}) = 0.035$ $p(\mathbf{e}, a|\mathbf{f}) = 0.08$ $p(\mathbf{e}, a|\mathbf{f}) = 0.005$

IBM Model 1 and EM

t-table **Probabilities**

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

Alignments



$p(\mathbf{e}, a|\mathbf{f}) = 0.56$ $\quad p(\mathbf{e}, a|\mathbf{f}) = 0.035$ $\quad p(\mathbf{e}, a|\mathbf{f}) = 0.08$ $\quad p(\mathbf{e}, a|\mathbf{f}) = 0.005$

Applying the chain rule: $p(a|\mathbf{e}, \mathbf{f}) = \dfrac{p(\mathbf{e}, a|\mathbf{f})}{p(\mathbf{e}|\mathbf{f})}$

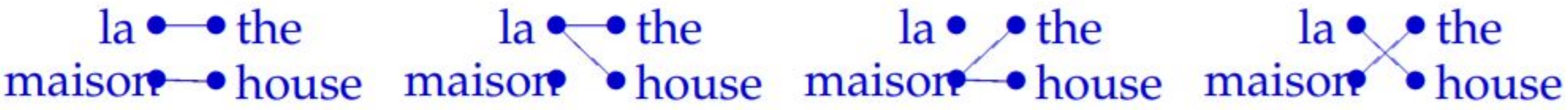
$$p(e, a) = p(e)p(a|e)$$

IBM Model 1 and EM: Expectation Step

t-table **Probabilities**

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

Alignments



$p(\mathbf{e}, a|\mathbf{f}) = 0.56$ $\quad p(\mathbf{e}, a|\mathbf{f}) = 0.035$ $\quad p(\mathbf{e}, a|\mathbf{f}) = 0.08$ $\quad p(\mathbf{e}, a|\mathbf{f}) = 0.005$

E-step $\quad p(a|\mathbf{e}, \mathbf{f}) = 0.824$ $\quad p(a|\mathbf{e}, \mathbf{f}) = 0.052$ $\quad p(a|\mathbf{e}, \mathbf{f}) = 0.118$ $\quad p(a|\mathbf{e}, \mathbf{f}) = 0.007$

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

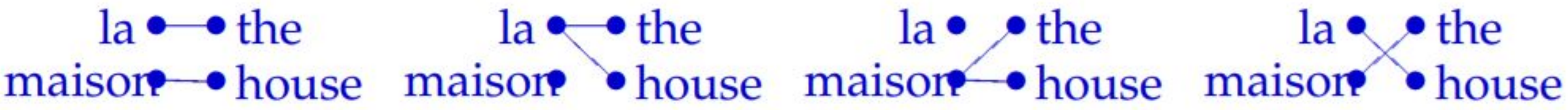
IBM Model 1 and EM: Maximization Step

t-table **Probabilities**

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

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

Alignments



$p(\mathbf{e}, a|\mathbf{f}) = 0.56$ $\quad$ $p(\mathbf{e}, a|\mathbf{f}) = 0.035$ $\quad$ $p(\mathbf{e}, a|\mathbf{f}) = 0.08$ $\quad$ $p(\mathbf{e}, a|\mathbf{f}) = 0.005$

E-step $p(a|\mathbf{e}, \mathbf{f}) = 0.824$ $\quad$ $p(a|\mathbf{e}, \mathbf{f}) = 0.052$ $\quad$ $p(a|\mathbf{e}, \mathbf{f}) = 0.118$ $\quad$ $p(a|\mathbf{e}, \mathbf{f}) = 0.007$

M-step **Counts**

$c(\text{the}|\text{la}) = 0.824 + 0.052$ $\quad$ $c(\text{house}|\text{la}) = 0.052 + 0.007$

$c(\text{the}|\text{maison}) = 0.118 + 0.007$ $\quad$ $c(\text{house}|\text{maison}) = 0.824 + 0.118$

IBM Model 1 and EM: Maximization Step

t-table **Probabilities**

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

E-step **Alignments**

$$p(a|\mathbf{e},\mathbf{f}) = 0.824 \quad p(a|\mathbf{e},\mathbf{f}) = 0.052 \quad p(a|\mathbf{e},\mathbf{f}) = 0.118 \quad p(a|\mathbf{e},\mathbf{f}) = 0.007$$

M-step **Counts**

$$c(\text{the}|\text{la}) = 0.824 + 0.052 \quad c(\text{house}|\text{la}) = 0.052 + 0.007$$
$$c(\text{the}|\text{maison}) = 0.118 + 0.007 \quad c(\text{house}|\text{maison}) = 0.824 + 0.118$$

Update t-table:

$$p(\text{the}|\text{la}) = c(\text{the}|\text{la})/c(\text{la})$$

IBM Model 1 and EM: Pseudocode

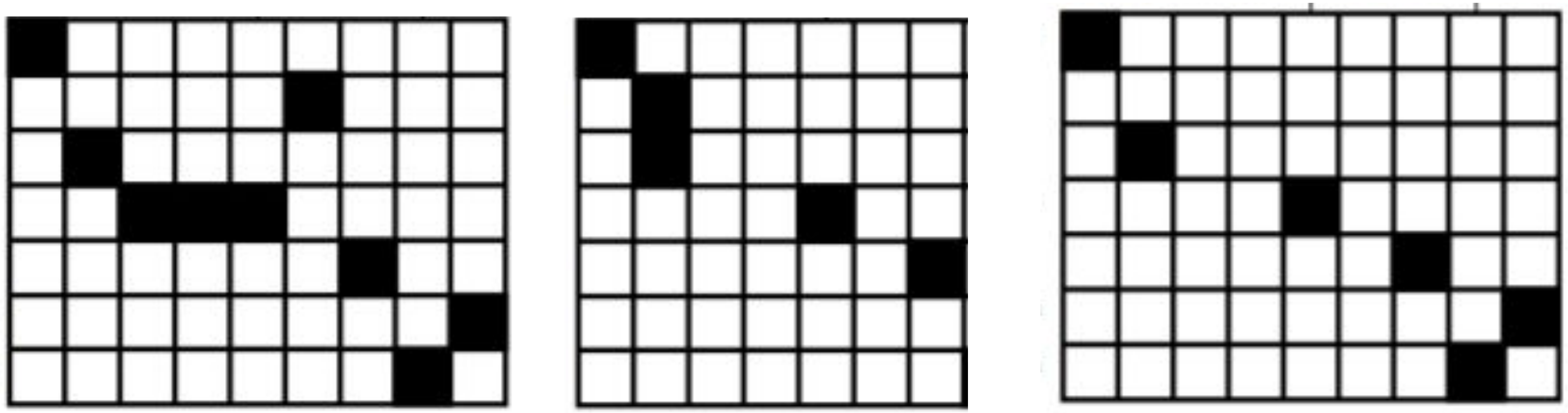
```
Input: set of sentence pairs (e, f)
Output: translation prob. t(e|f)
initialize t(e|f) uniformly
while not converged do
    // initialize
    count(e|f) = 0 for all e, f
    total(f) = 0 for all f
    for all sentence pairs (e,f) do
        // compute normalization
        for all words e in e do
            s-total(e) = 0
            for all words f in f do
                s-total(e) += t(e|f)
            end for
        end for
        // collect counts
        for all words e in e do
            for all words f in f do
                count(e|f) += t(e|f) / s-total(e)
                total(f) += t(e|f) / s-total(e)
            end for
        end for
    end for
    // estimate probabilities
    for all foreign words f do
        for all English words e do
            t(e|f) = count(e|f) / total(f)
        end for
    end for
end while
```

- implementation of IBM Model 1 after applying the simplification trick (in previous lecture)

IBM Model 2

$$p(\mathbf{e}, a|\mathbf{f}) = \epsilon \prod_{j=1}^{l_e} t(e_j|f_{a(j)}) align\_prob(a(j)|j, l_e, l_f)$$

how many alignment parameters?



EM for Model 2

- Now we have two sets of parameters:
  - initialize t-table parameters uniformly or **carry over from trained Model 1**
  - initialize alignment probabilities uniformly

```
for each l_e do:
  for each l_f do:
    for i = 0..l_f do
      for j = 1..l_e do
        align_prob(i|j, l_e, l_f) = 1/(l_f+1)
```

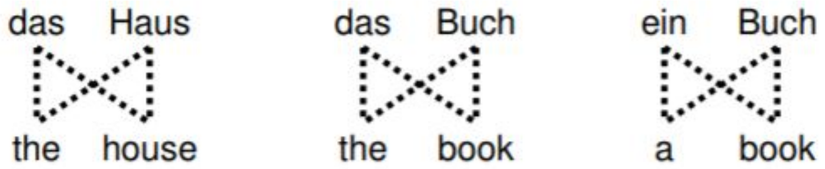
e	f	initial
the	das	0.25
book	das	0.25
house	das	0.25
the	buch	0.25
book	buch	0.25
a	buch	0.25
book	ein	0.25
a	ein	0.25
the	haus	0.25
house	haus	0.25

EM for Model 2

- initialize model parameters:
- repeat until convergence:
  - compute “expected” alignments

e	f	initial
the	das	0.25
book	das	0.25
house	das	0.25
the	buch	0.25

```
for each l_e do:
  for each l_f do:
    for i = 0..l_f do
      for j = 1..l_e do
        a(i|j, l_e, l_f) = 1/(l_f+1)
```



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

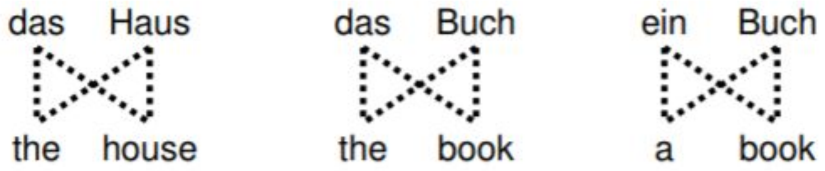
$$p(\mathbf{e}, a|\mathbf{f}) = \epsilon \prod_{j=1}^{l_e} t(e_j|f_{a(j)}) align\_prob(a(j)|j, l_e, l_f)$$

$$p(\mathbf{e}|\mathbf{f}) = \epsilon \prod_{j=1}^{l_e} \sum_{i=0}^{l_f} t(e_j|f_i) align\_prob(i|j, l_e, l_f)$$

EM for Model 2

- initialize model parameters, e.g. uniform:
- repeat until convergence:
    - compute "expected" alignments

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



    - keep track of the expected number of times *f* translates into *e* throughout the whole corpus

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

same as in Model 1

```
Initialize c(e|f) = 0 for all e, f in vocab
for every sentence pair e,f:
  for every alignment a do
    for j = 1..len(e) do
      c(e_j|f_a(j)) += p(a|e,f)
```

EM for Model 2

- initialize model parameters, e.g. uniform:
- repeat until convergence:
    - compute "expected" alignments

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

    - keep track of the expected number of times *f* translates into *e* throughout the whole corpus

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

    - apply MLE to estimate new model parameters

same as in Model 1 for t-table parameters

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

e	f	initial	1st it.
the	das	0.25	0.5
book	das	0.25	0.25
house	das	0.25	0.25
the	buch	0.25	0.25

EM for Model 2

- keep track of the expected number of times *f* translates into *e* throughout the whole corpus

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

- apply MLE to estimate new model parameters

same as in Model 1 for t-table parameters

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

e	f	initial	1st it.
the	das	0.25	0.5
book	das	0.25	0.25
house	das	0.25	0.25
the	buch	0.25	0.25

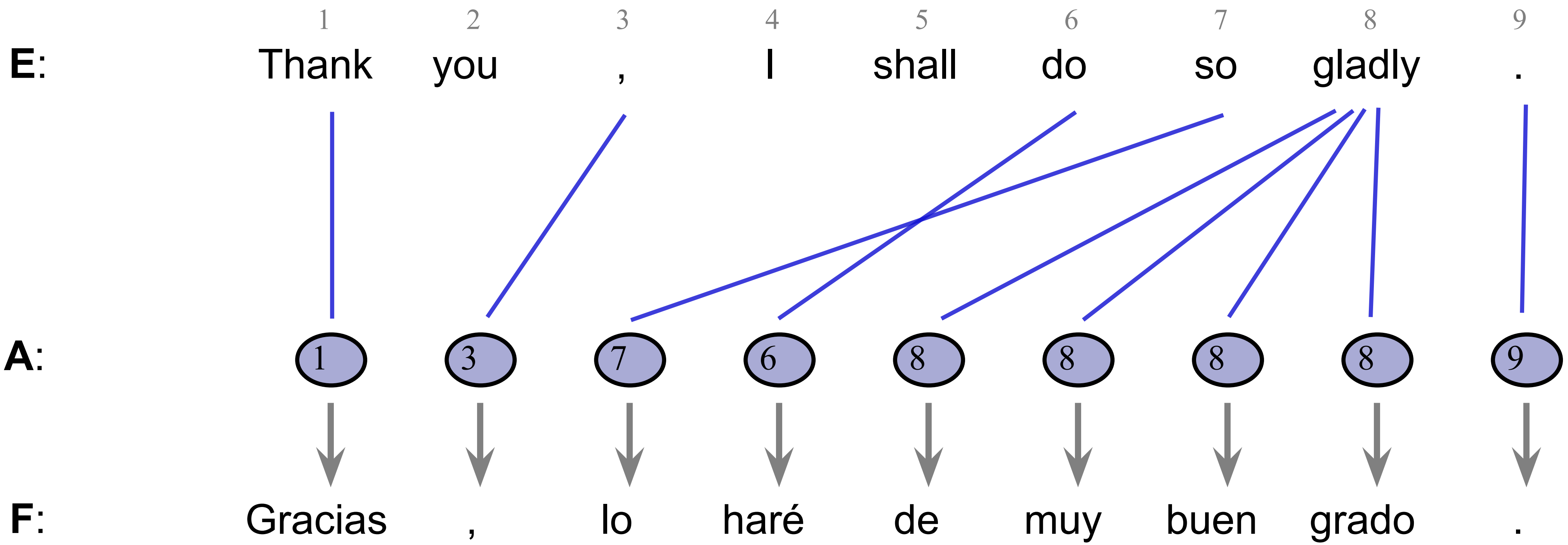
$$c(i|j,l_e,l_f;\mathbf{e},\mathbf{f}) = \frac{t(e_j|f_i) align\_prob(a(j)|j,l_e,l_f)}{\sum_{i'=0}^{l_f} t(e_j|f_{i'}) align\_prob(a(i')|j,l_e,l_f)}$$

IBM Model 2 and EM

```
Input: set of sentence pairs (e,f)
Output: probability distributions t (lexical translation)
    and a (alignment)
carry over t(e|f) from Model 1
initialize a(i|j,l_e,l_f) = 1/(l_f+1) for all i,j,l_e,l_f
while not converged do
  // initialize
  count(e|f) = 0 for all e,f
  total(f) = 0 for all f
  count_a(i|j,l_e,l_f) = 0 for all i,j,l_e,l_f
  total_a(j,l_e,l_f) = 0 for all j,l_e,l_f
  for all sentence pairs (e,f) do
    l_e = length(e), l_f = length(f)
    // compute normalization
    for j = 1 .. l_e do // all word positions in e
      s-total(e_j) = 0
      for i = 0 .. l_f do // all word positions in f
        s-total(e_j) += t(e_j|f_i) * a(i|j,l_e,l_f)
      end for
    end for
    // collect counts
    for j = 1 .. l_e do // all word positions in e
      for i = 0 .. l_f do // all word positions in f
        c = t(e_j|f_i) * a(i|j,le,lf) / s-total(e_j)
        count(e_j|f_i) += c
        total(f_i) += c
        count_a(i|j,le,lf) += c
        total_a(j,l_e,l_f) += c
      end for
    end for
  end for
  // estimate probabilities
  t(e|f) = 0 for all e,f
  a(i|j,l_e,l_f) = 0 for all i,j,l_e,l_f
  for all e,f do
    t(e|f) = count(e|f) / total(f)
  end for
  for all i,j,l_e,l_f do
    a(i|j,l_e,l_f) = count_a(i|j,l_e,l_f) / total_a(j,l_e,l_f)
  end for
end while
```

- implementation of IBM Model 2 after applying the simplification trick (in previous lecture)

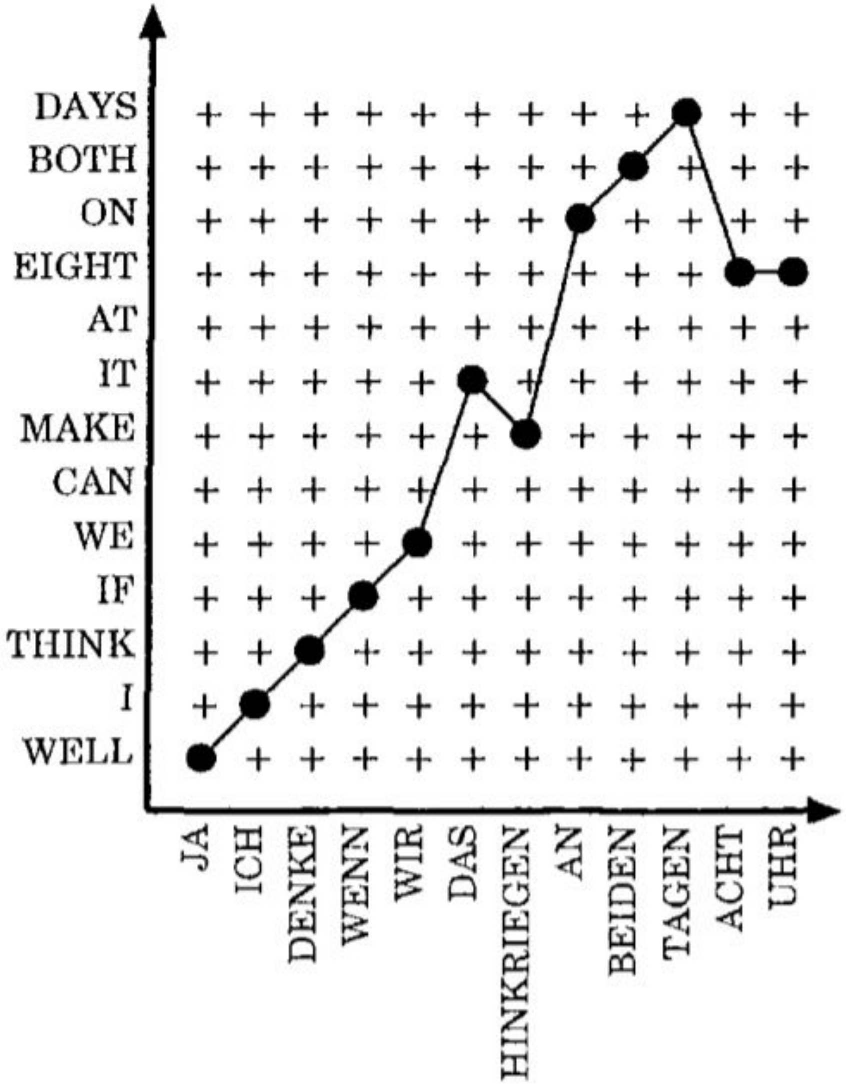
IBM Models 1/2



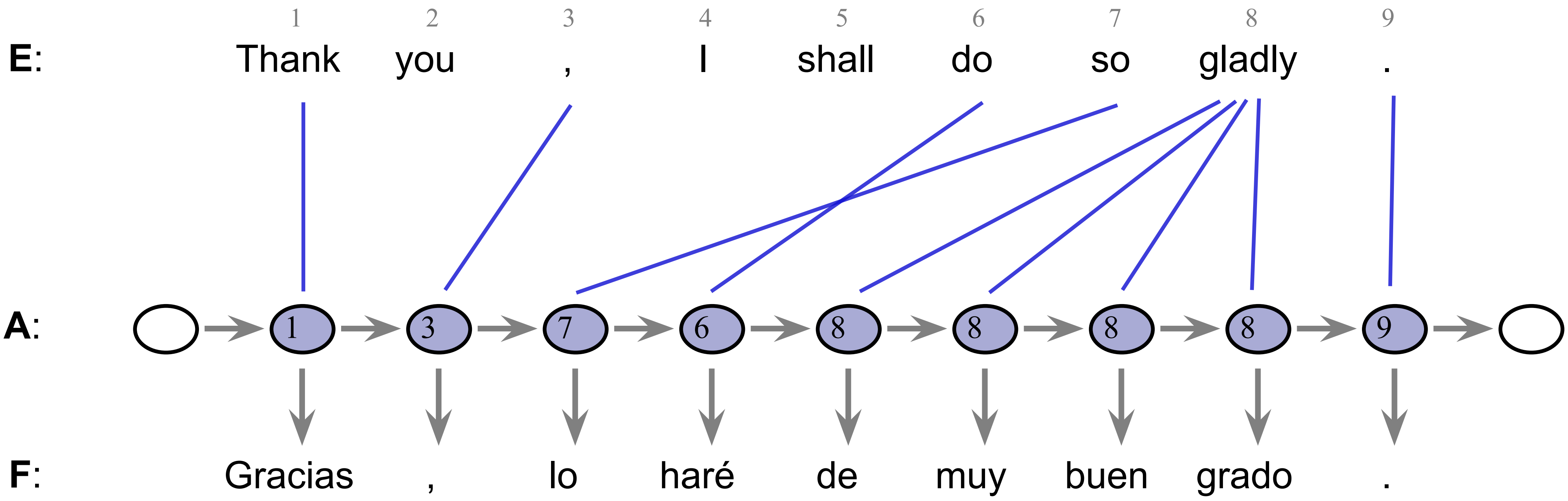
Model Parameters

Emissions: P(F_1 = Gracias | E_{A1} = Thank) *Transitions*: P(A_2 = 3)

Localization effect in aligning the words



The HMM Model



Model Parameters

Emissions: P(F_1 = Gracias | E_{A1} = Thank)

Transitions: P(A_2 = 3 | A_1 = 1)

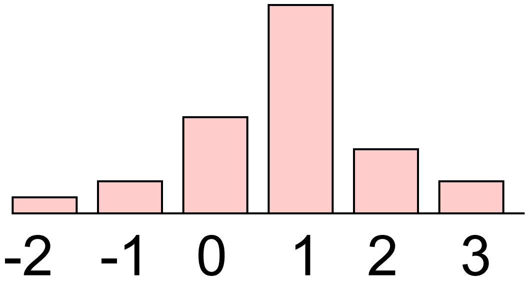
The HMM Model

- Model 2 preferred global monotonicity
- We want local monotonicity:
  - Most jumps are small
- HMM model (Vogel 96)

f	$t(f \mid e)$
nationale	0.469
national	0.418
nationaux	0.054
nationales	0.029

$$P(f, a|e) = \prod_j P(a_j|a_{j-1}) P(f_j|e_i)$$

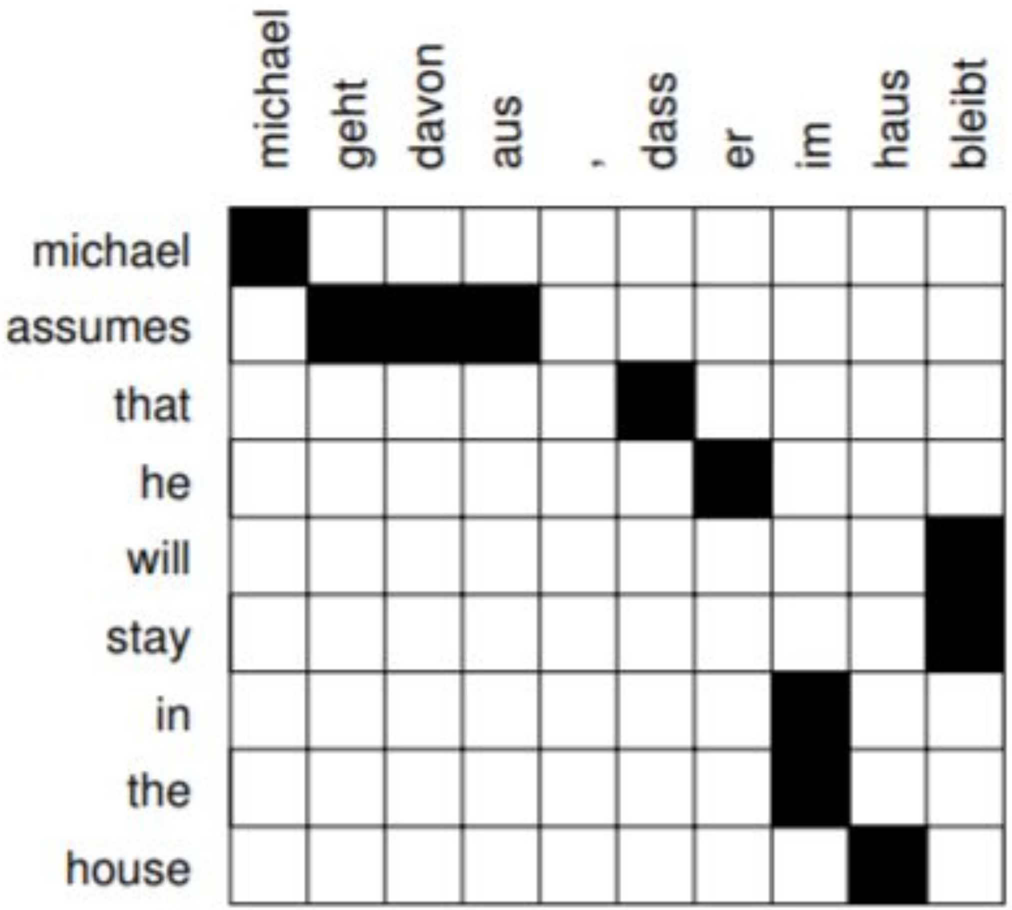
$$P(a_j - a_{j-1})$$



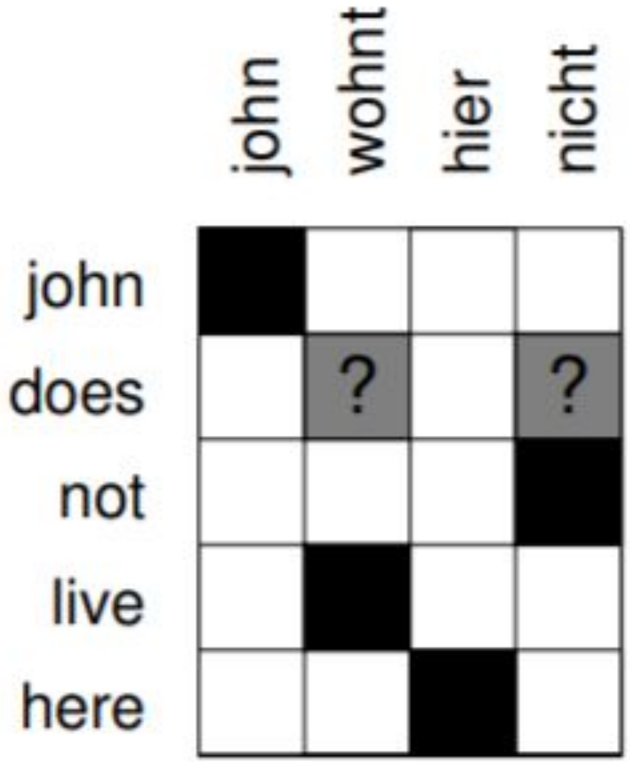
  - Re-estimate using the forward-backward algorithm
  - Handling nulls requires some care

Word Alignment

Given a sentence pair, which words correspond to each other?

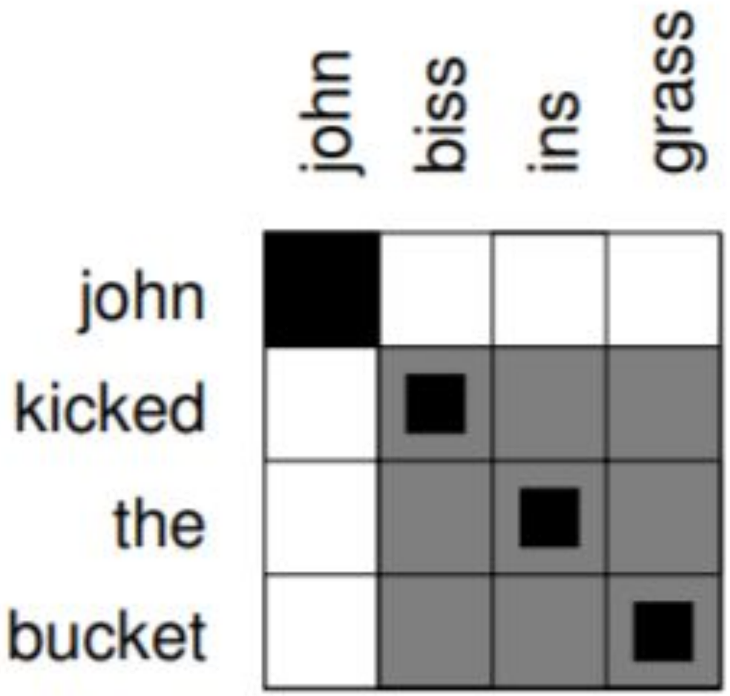


Word Alignment?



Is the English word does aligned to
the German wohnt (verb) or nicht (negation) or neither?

Word Alignment?

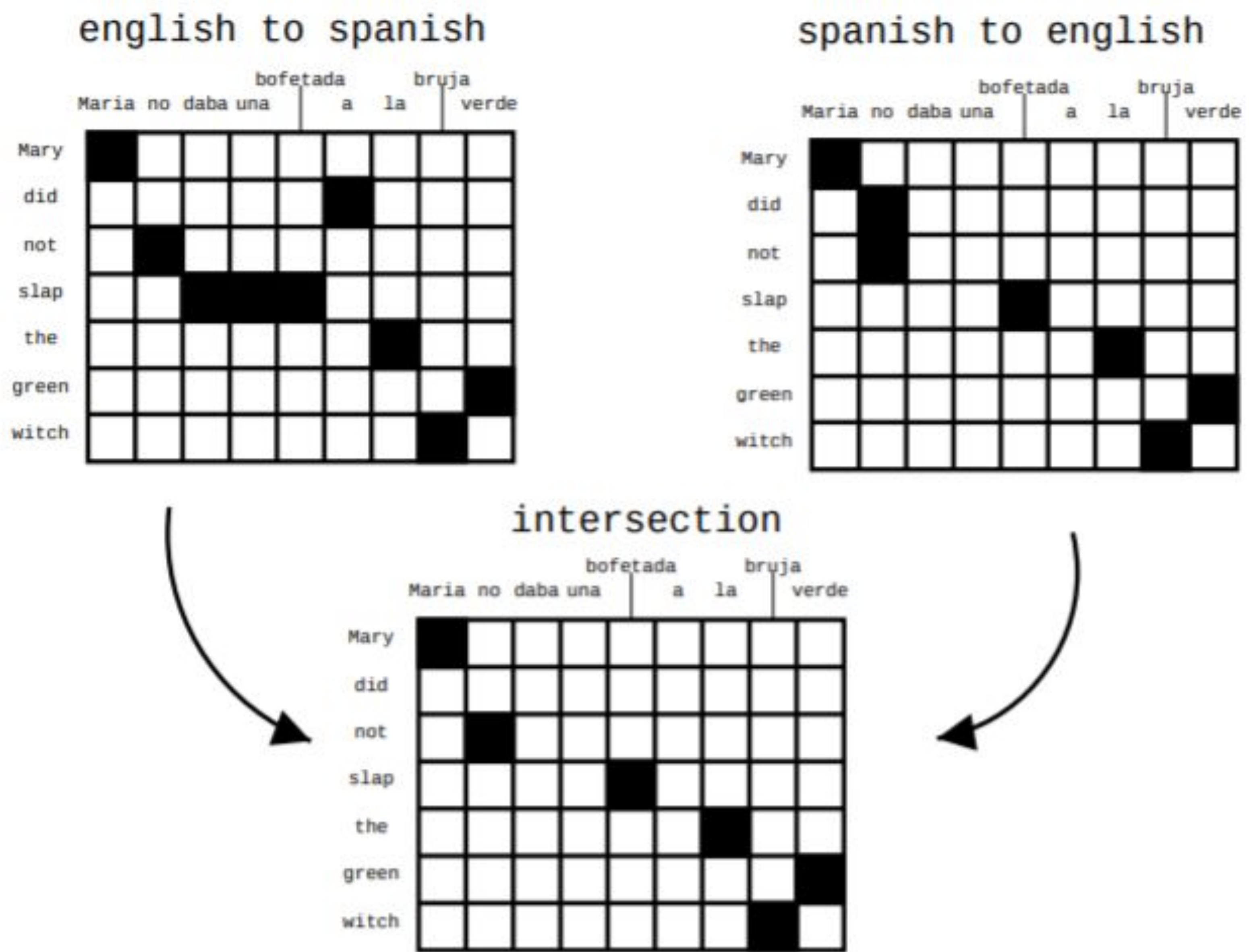


How do the idioms kicked the bucket and biss ins grass match up?
Outside this exceptional context, bucket is never a good translation for grass

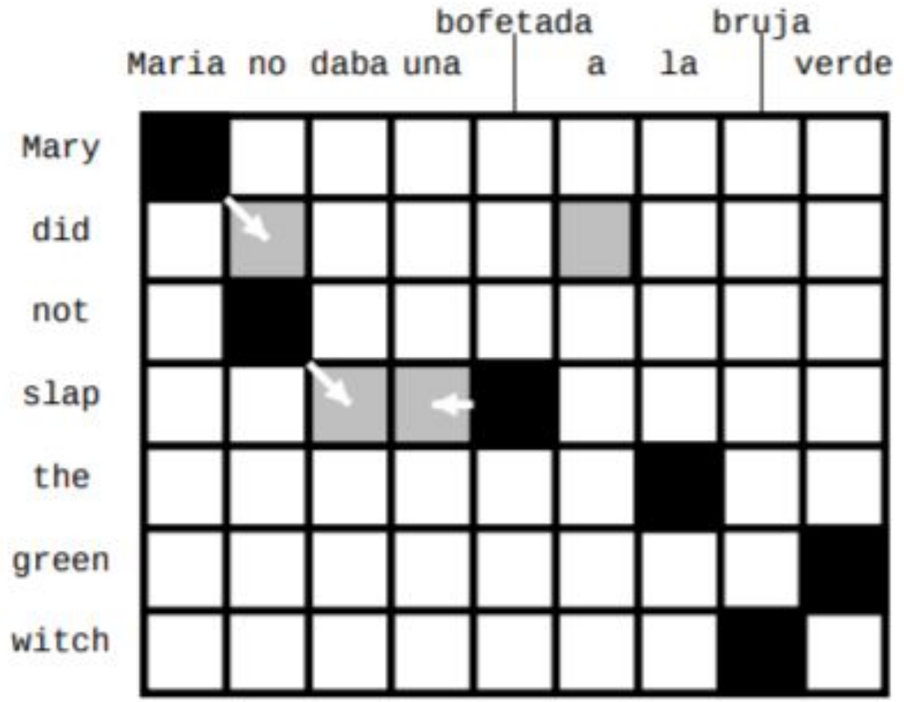
Word Alignment and IBM Models

- IBM Models create a many-to-one mapping
  - words are aligned using an alignment function
  - a function may return the same value for different input (one-to-many mapping)
  - a function can not return multiple values for one input (no many-to-one mapping)
- Real word alignments have many-to-many mappings

Symmetrization



Growing Heuristics



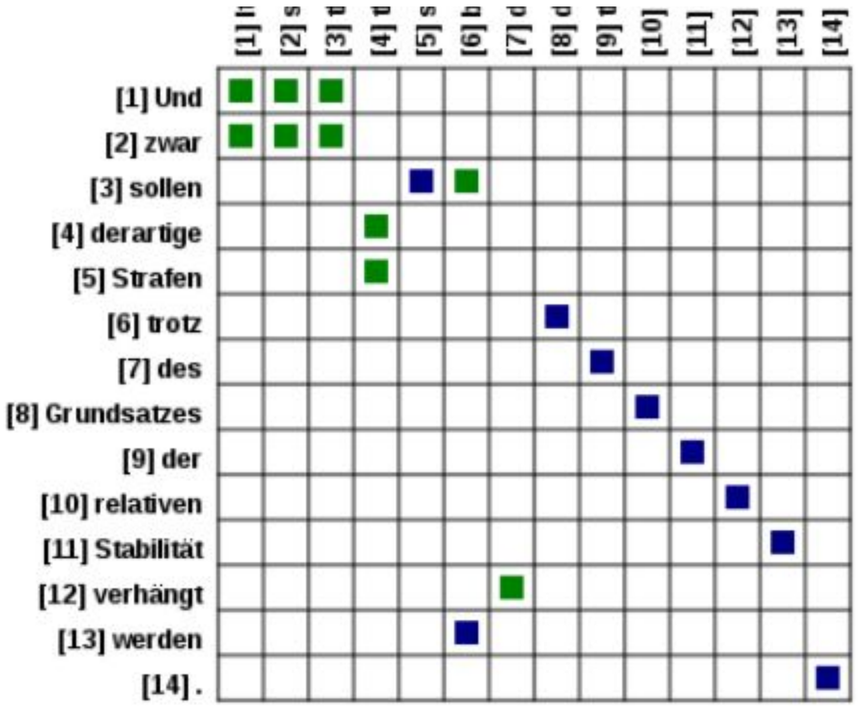
black: intersection grey: additional points in union

- Add alignment points from union based on heuristics
- Popular method: grow-diag-final-and

Alignment Error Rate

Possible links

P



Sure links

S

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

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

AER for HMMs

Model	AER
Model 1 INT	19.5
HMM E→F	11.4
HMM F→E	10.8
HMM AND	7.1
HMM INT	4.7
GIZA M4 AND	6.9

Phrase-Based MT

Phrase-Based Translation Overview

Input:	lo haré	rápidamente	.	*tries different segmentations,*
Translations:	I'll do it	quickly	.	*translates phrase by phrase,*
	quickly	I'll do it	.	*and considers reorderings.*

The decoder...

tries different segmentations,

translates phrase by phrase,

and considers reorderings.

Objective:

$$\arg\max_{\mathbf{e}} \left[P(\mathbf{f}|\mathbf{e}) \cdot P(\mathbf{e}) \right]$$

$$\arg\max_{\mathbf{e}} \left[\prod_{\langle \bar{e}, \bar{f} \rangle} P(\bar{f}|\bar{e}) \cdot \prod_{i=1}^{|\mathbf{e}|} P(e_i | e_{i-1}, e_{i-2}) \right]$$

Phrase Translation Example

- Phrase translations for den Vorschlag learned from the Europarl corpus:

English	$\phi(\bar{e}\|\bar{f})$	English	$\phi(\bar{e}\|\bar{f})$
the proposal	0.6227	the suggestions	0.0114
's proposal	0.1068	the proposed	0.0114
a proposal	0.0341	the motion	0.0091
the idea	0.0250	the idea of	0.0091
this proposal	0.0227	the proposal ,	0.0068
proposal	0.0205	its proposal	0.0068
of the proposal	0.0159	it	0.0068
the proposals	0.0159	...	...

– lexical variation (proposal vs suggestions)
– morphological variation (proposal vs proposals)
– included function words (the, a, ...)
– noise (it)

Linguistic Phrases?

- Model is not limited to linguistic phrases (noun phrases, verb phrases, prepositional phrases, ...)
- Example non-linguistic phrase pair

spass am $\longrightarrow$ fun with the

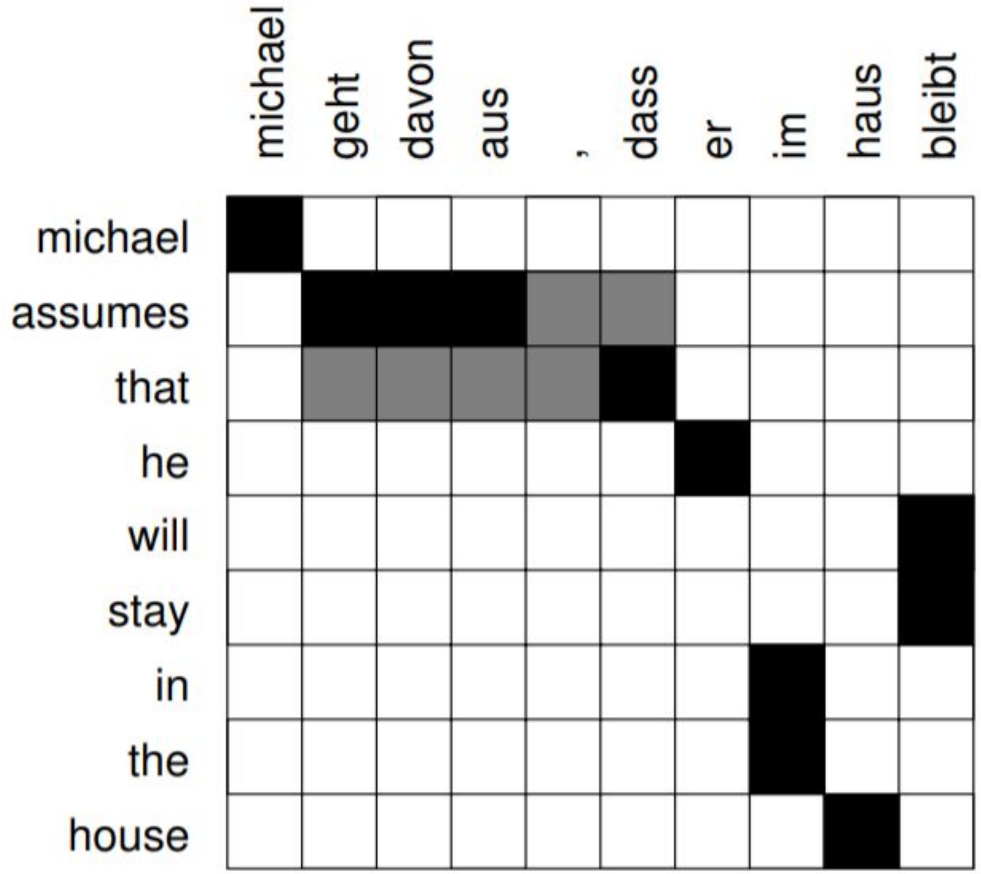
- Prior noun often helps with translation of preposition
- Experiments show that limitation to linguistic phrases hurts quality

Another Example

这	7人	中包括	来自	法国	和	俄罗斯	的	宇航	员	.
the	7 people	including	by some		**and**	the russian	**the**	the astronauts		,
it	7 people included		by france		and the	the russian		international astronautical	of rapporteur .	
this	7 out	including the	**from**	the french	and the russian		the fifth		.	
these	7 among	including from		the french and		of the russian	of	space	members	.
that	7 persons	including from the		of france	and to	russian	of the	aerospace	members .	
	7 include		from the	of france and		russian		**astronauts**		. the
	7 numbers include		**from france**		and russian		of astronauts who			. "
	7 populations include		those from france		and russian			astronauts .		
	7 deportees included		come from	**france**	**and russia**		in	astronautical	personnel	;
	7 philtrum	including those from		**france and**		**russia**	a space		**member**	
		including representatives from		france and the		**russia**		astronaut		
		include	came from	**france and russia**			by cosmonauts			
		include representatives from		french	**and russia**			cosmonauts		
		include	came from france		and russia 's			cosmonauts .		
		includes	coming from	french and		russia 's		cosmonaut		
				french and russian			's	astronavigation	member .	
				french	**and russia**		**astronauts**			
					and russia 's				special rapporteur	
					, and	**russia**			rapporteur	
					, and russia				rapporteur .	
					, and russia					
					or	russia 's				

Decoder design is important: [Koehn et al. 03]

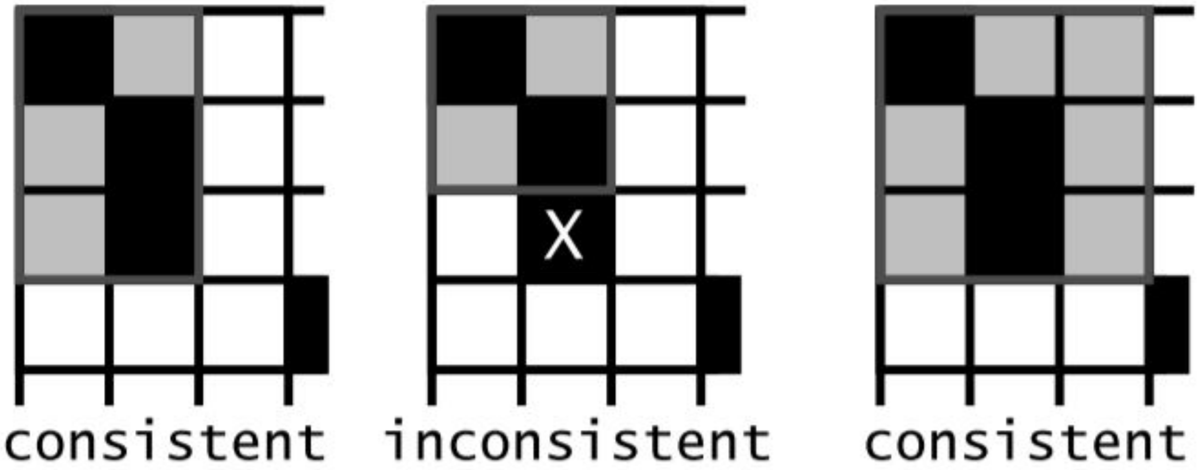
Extracting Phrase Pairs



extract phrase pair consistent with word alignment:

assumes that / geht davon aus , dass

Consistent



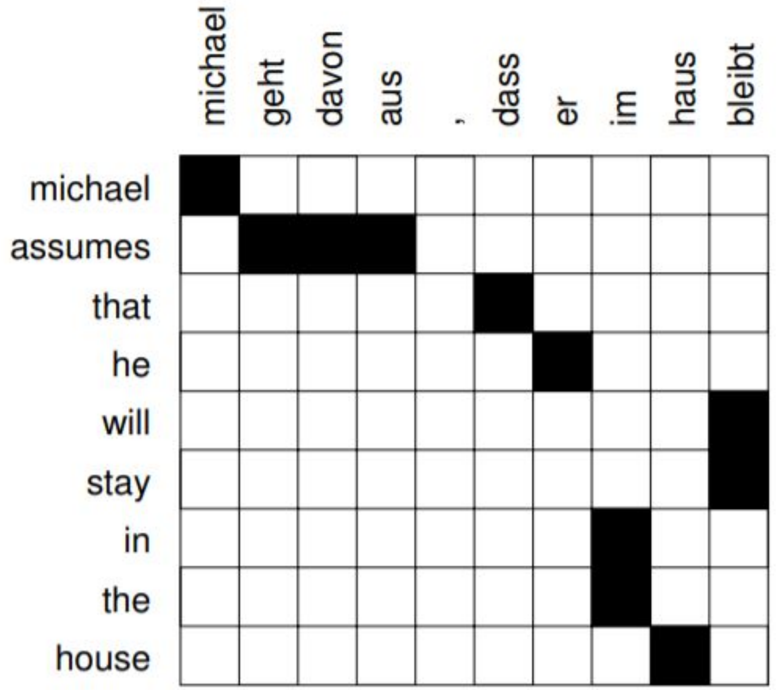
ok

violated
one alignment point outside

ok
unaligned word is fine

All words of the phrase pair have to align to each other.

Phrase Pair Extraction

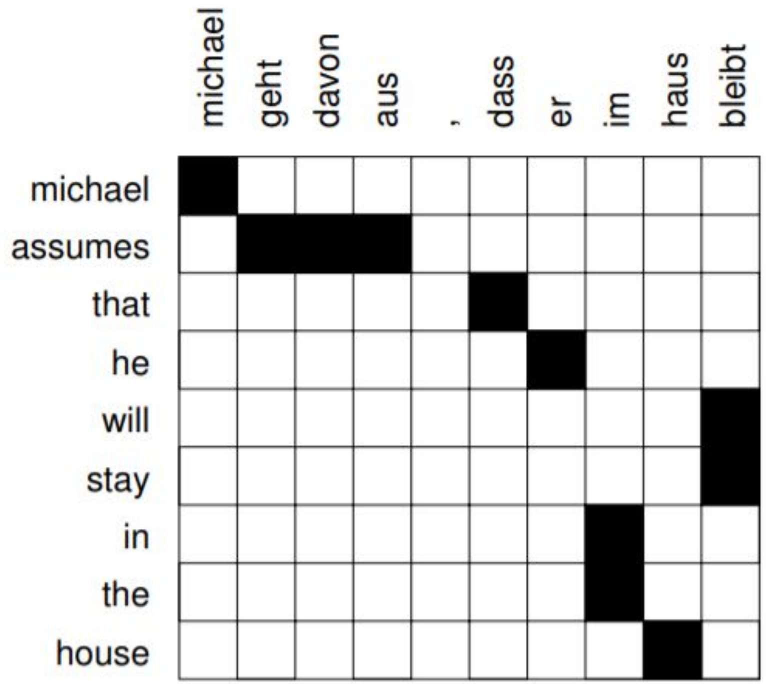


Smallest phrase pairs:

michael — michael
assumes — geht davon aus / geht davon aus ,
that — dass / , dass
he — er
will stay — bleibt
in the — im
house — haus

unaligned words (here: German comma) lead to multiple translations

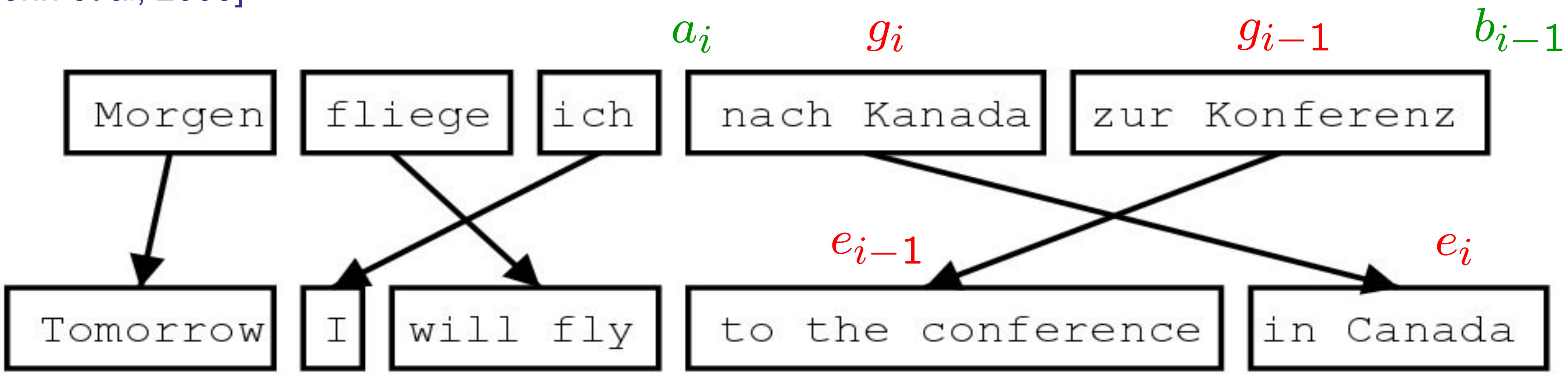
Larger Phrase Pairs



michael assumes — michael geht davon aus / michael geht davon aus ,
assumes that — geht davon aus , dass ; assumes that he — geht davon aus , dass er
that he — dass er / , dass er ; in the house — im haus
michael assumes that — michael geht davon aus , dass
michael assumes that he — michael geht davon aus , dass er
michael assumes that he will stay in the house — michael geht davon aus , dass er im haus bleibt
assumes that he will stay in the house — geht davon aus , dass er im haus bleibt
that he will stay in the house — dass er im haus bleibt ; dass er im haus bleibt ,
he will stay in the house — er im haus bleibt ; will stay in the house — im haus bleibt

The Pharaoh "Model"

[Koehn et al, 2003]



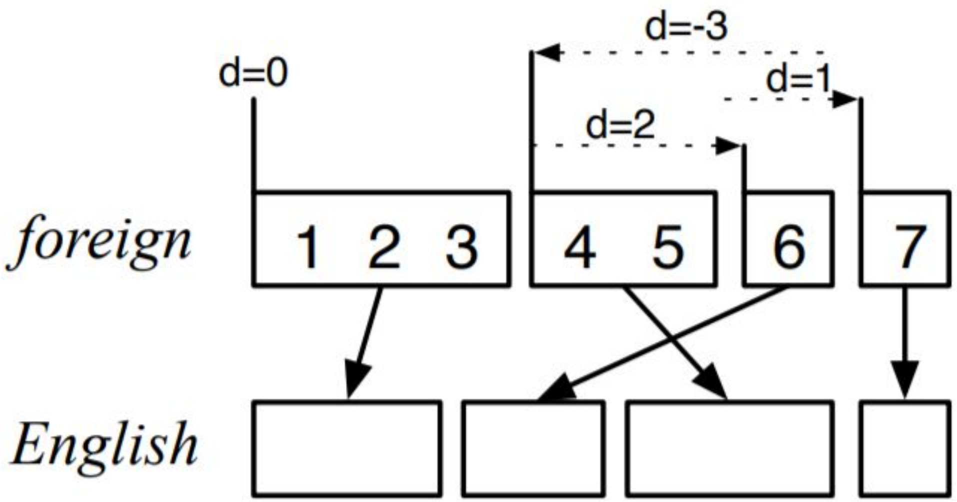
$$P(e|g) = P(\{\bar{g}_i\}|g) \prod_i \phi(\bar{e}_i|\bar{g}_i) d(a_i - b_{i-1})$$

Segmentation

Translation

Distortion

Distance-Based Reordering



phrase	translates	movement	distance
1	1–3	start at beginning	0
2	6	skip over 4–5	+2
3	4–5	move back over 4–6	-3
4	7	skip over 6	+1

Scoring function: $d(x) = \alpha^{|x|}$ — exponential with distance

The Pharaoh "Model"

$$P(f|e) = P(\{\bar{e}_i\}|e) \prod_i \phi(\bar{f}_i|\bar{e}_i) d(a_i - b_{i-1})$$

$$\frac{1}{K} \qquad \frac{count(\bar{f}_i, \bar{e}_i)}{count(\bar{e}_i)} \qquad \alpha^{|a_i - b_{i-1}|}$$

Where do we get these counts?

Phrase Weights

How the MT community estimates $P(\bar{f}|\bar{e})$

Parallel training sentences *provide phrase pair counts.*

Gracias , lo haré de muy buen grado .
Thank you , I shall do so gladly .

lo haré $\Leftrightarrow$ I shall do so
44 times in the corpus

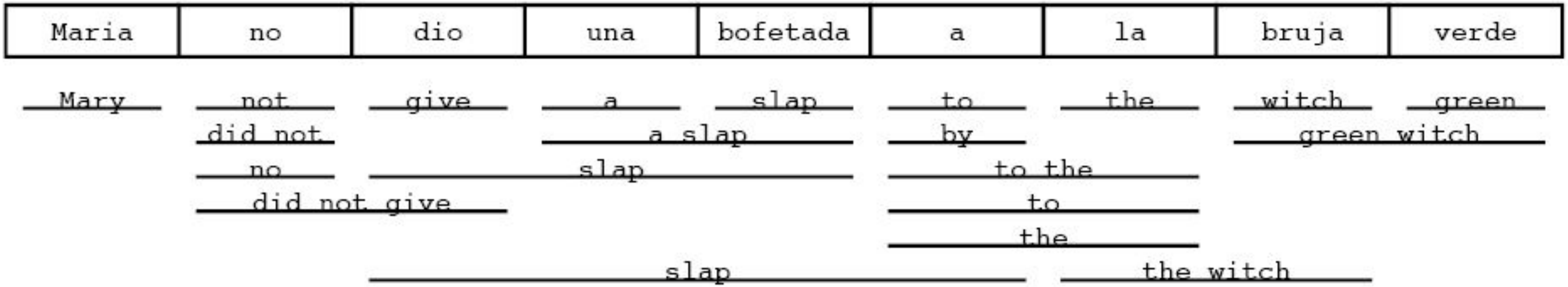
All phrase pairs are counted, *and counts are normalized.*

Gracias , lo haré de muy buen grado .

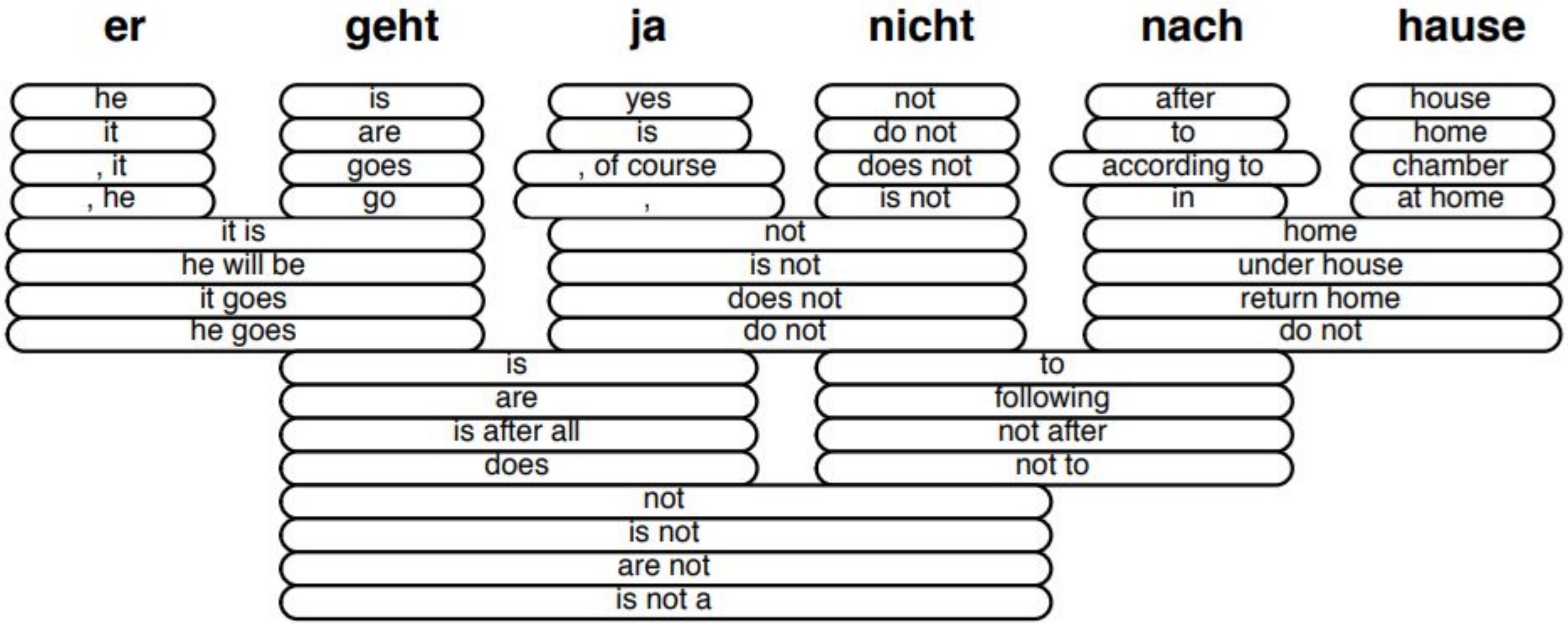
Thank you , I shall do so gladly .

$$P(\bar{f}|\bar{e}) = \frac{\text{count}(\bar{f}, \bar{e})}{\text{count}(\bar{e})}$$

Phrase-Based Decoding

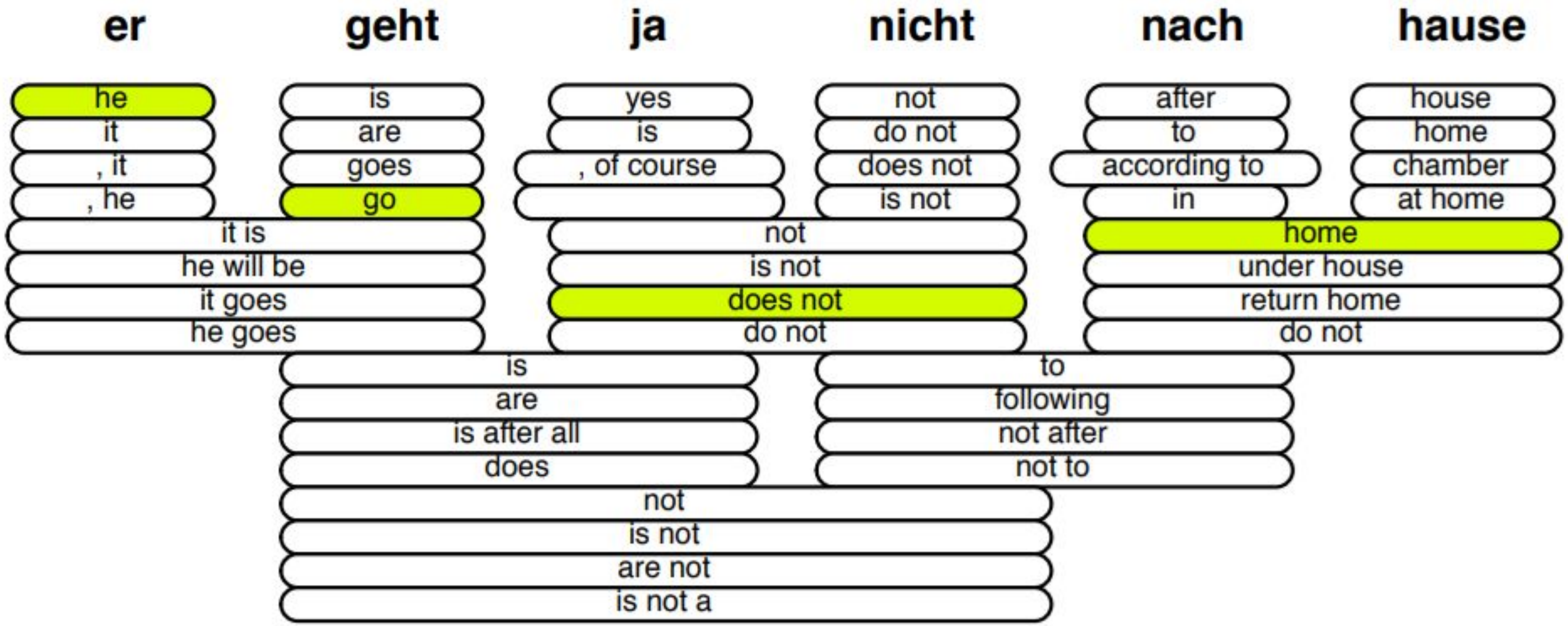


Translation Options



- Many translation options to choose from
  - in Europarl phrase table: 2727 matching phrase pairs for this sentence
  - by pruning to the top 20 per phrase, 202 translation options remain

Translation Options



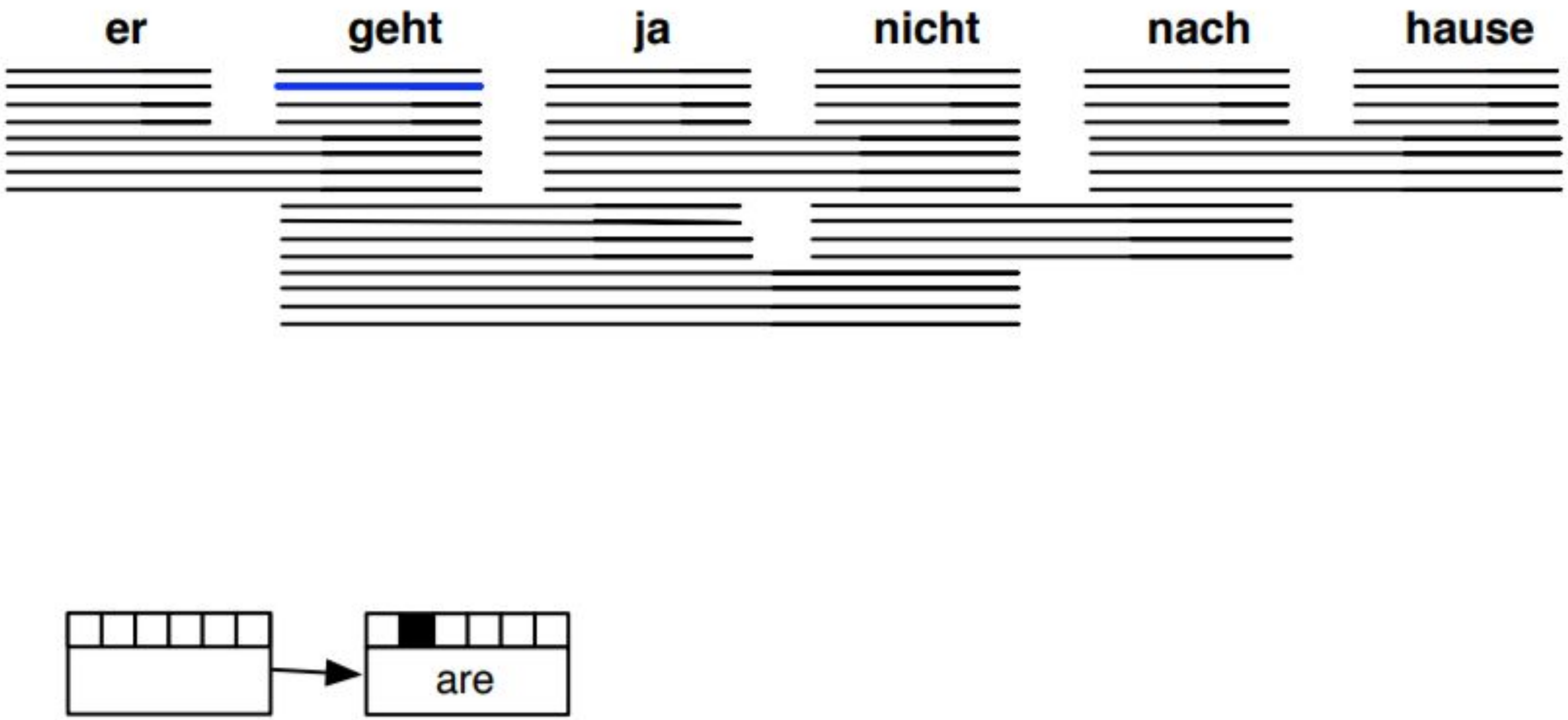
- The machine translation decoder does not know the right answer
  - picking the right translation options
  - arranging them in the right order

→ Search problem solved by heuristic beam search

Decoding: Start with Initial Hypothesis

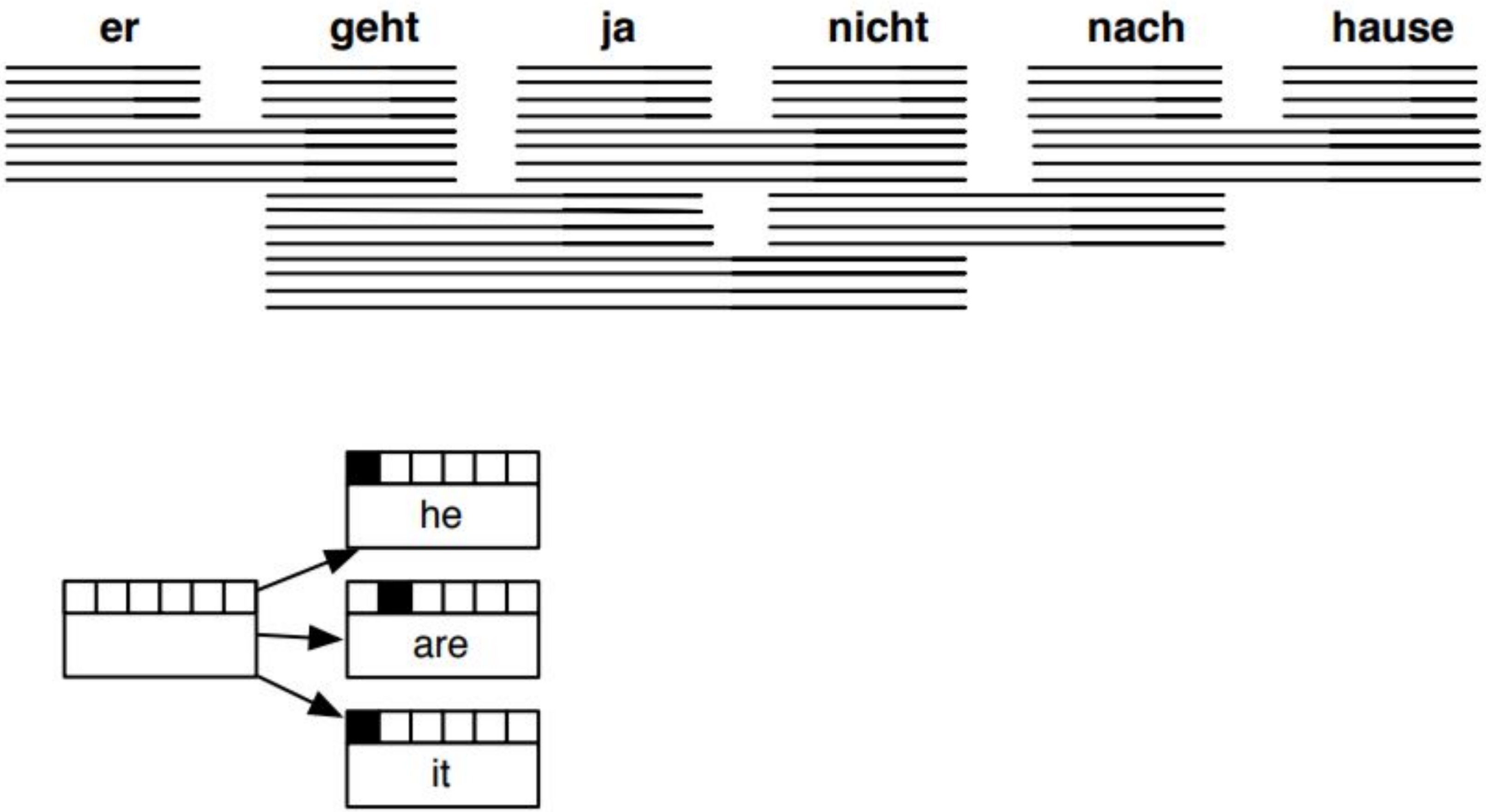
initial hypothesis: no input words covered, no output produced

Decoding: Hypothesis Expansion



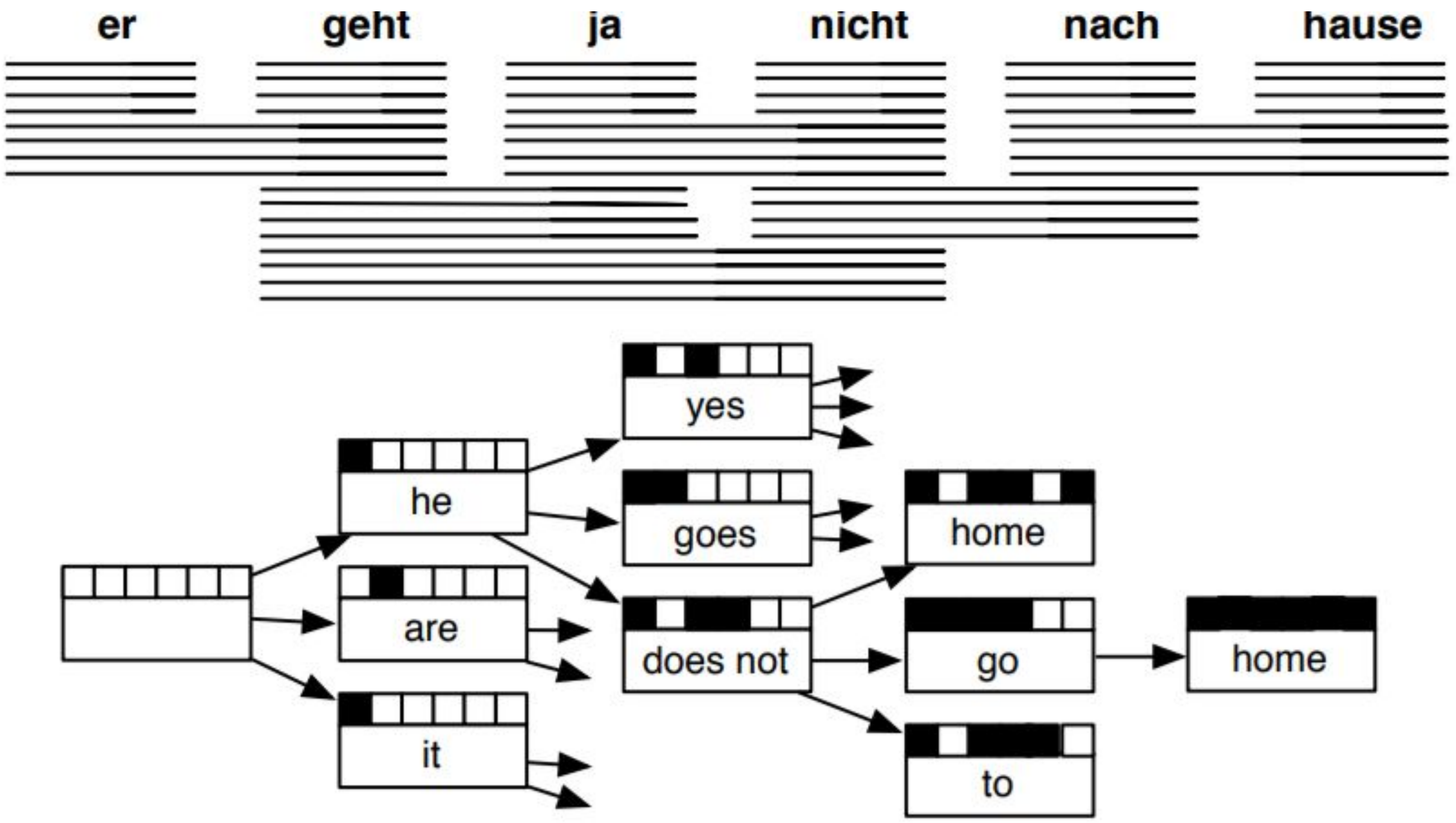
pick any translation option, create new hypothesis

Decoding: Hypothesis Expansion



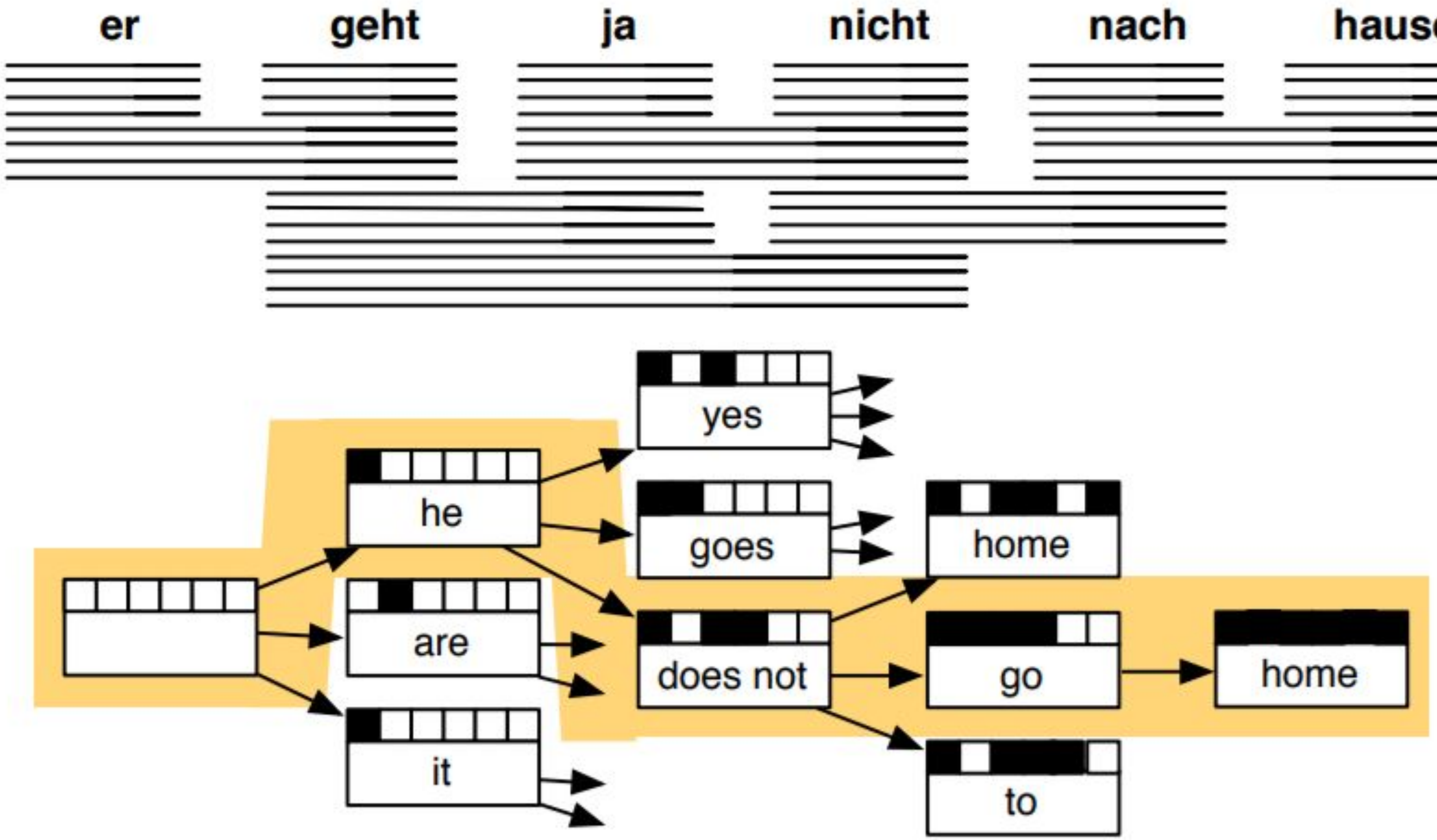
create hypotheses for all other translation options

Decoding: Hypothesis Expansion



also create hypotheses from created partial hypothesis

Decoding: Find Best Path



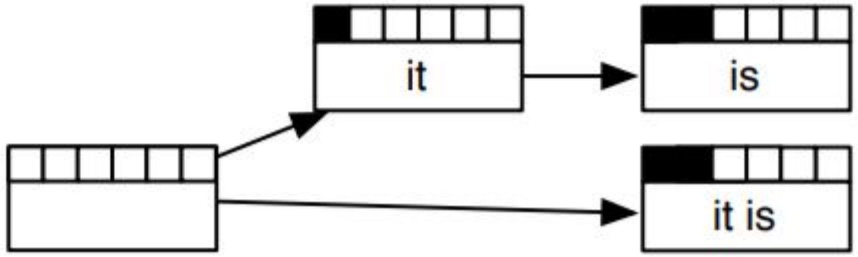
backtrack from highest scoring complete hypothesis

Computational Complexity

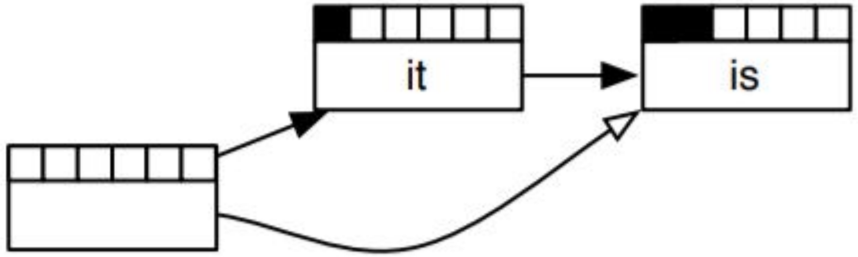
- The suggested process creates exponential number of hypothesis
- Machine translation decoding is NP-complete
- Reduction of search space:
  - recombination (risk-free)
  - pruning (risky)

Recombination

- Two hypothesis paths lead to two matching hypotheses
  - same foreign words translated
  - same English words in the output

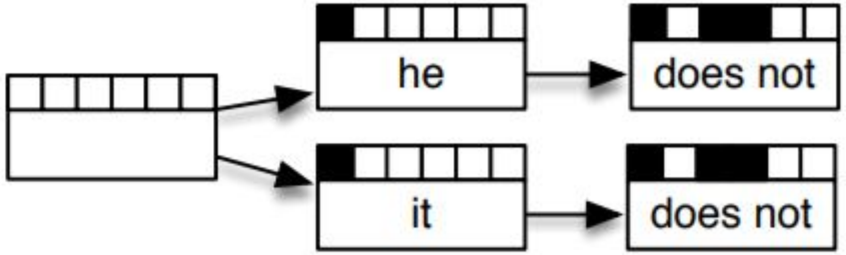


- Worse hypothesis is dropped

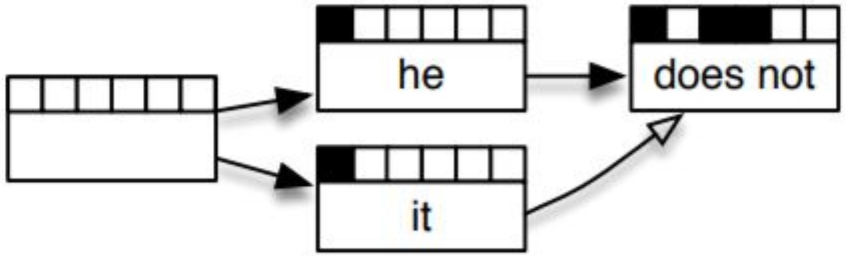


Recombination

- Two hypothesis paths lead to hypotheses indistinguishable in subsequent search
  - same foreign words translated
  - same last two English words in output (assuming trigram language model)
  - same last foreign word translated



- Worse hypothesis is dropped



Pruning: Beams + Forward Costs

```
Maria no    dio una bofetada    a la    bruja verde
```

```
e: Mary did not
f: **--------
p: 0.154
```

better partial translation

```
e: the
f: -----**--
p: 0.354
```

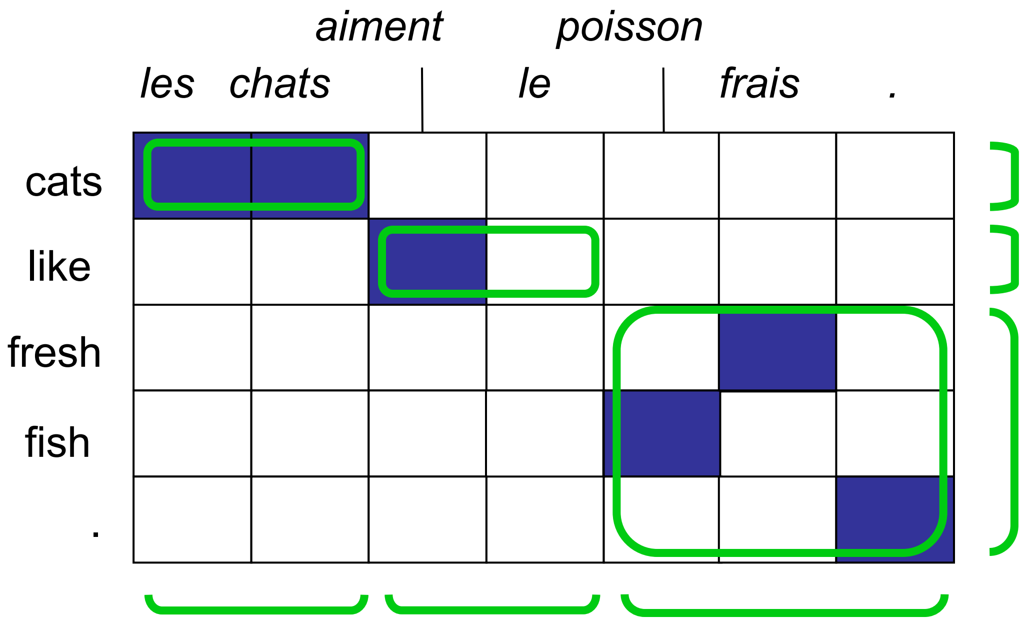
covers easier part --> lower cost

- Problem: easy partial analyses are cheaper
  - Solution 1: use beams per foreign subset
  - Solution 2: estimate forward costs (A*-like)

Parameter Tuning

Phrase Scoring

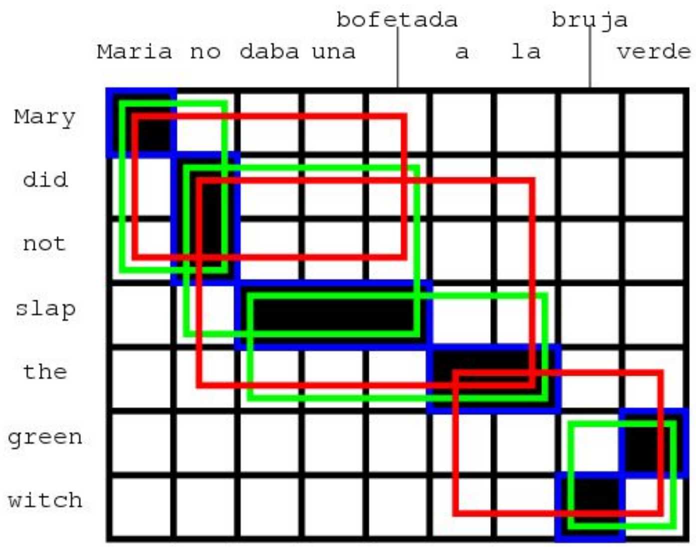
$$\phi_{new}(\bar{e}_j|\bar{f}_i) = \frac{c(\bar{f}_i, \bar{e}_j)}{c(\bar{f}_i)}$$

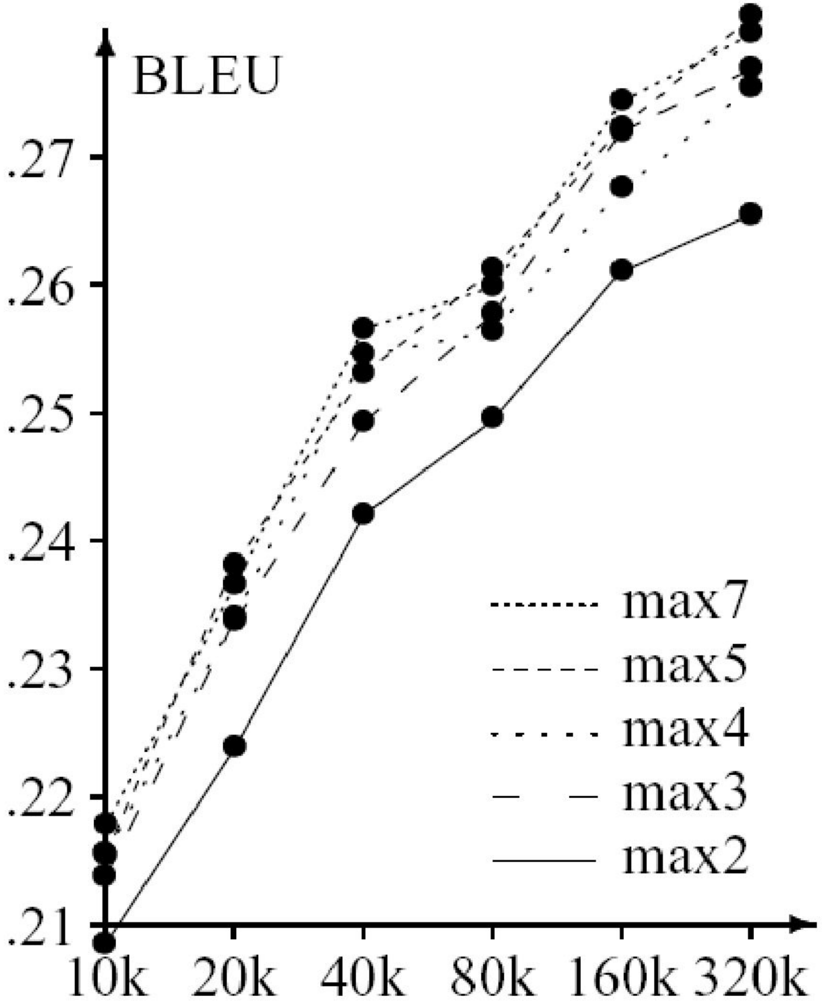


- Learning weights has been tried, several times:
  - [Marcu and Wong, 02]
  - [DeNero et al, 06]
  - … and others
- Seems not to work well, for a variety of partially understood reasons
- Main issue: big chunks get all the weight, obvious priors don't help
  - Though, [DeNero et al 08]

Phrase Size

- Phrases do help
  - But they don't need to be long
  - Why should this be?



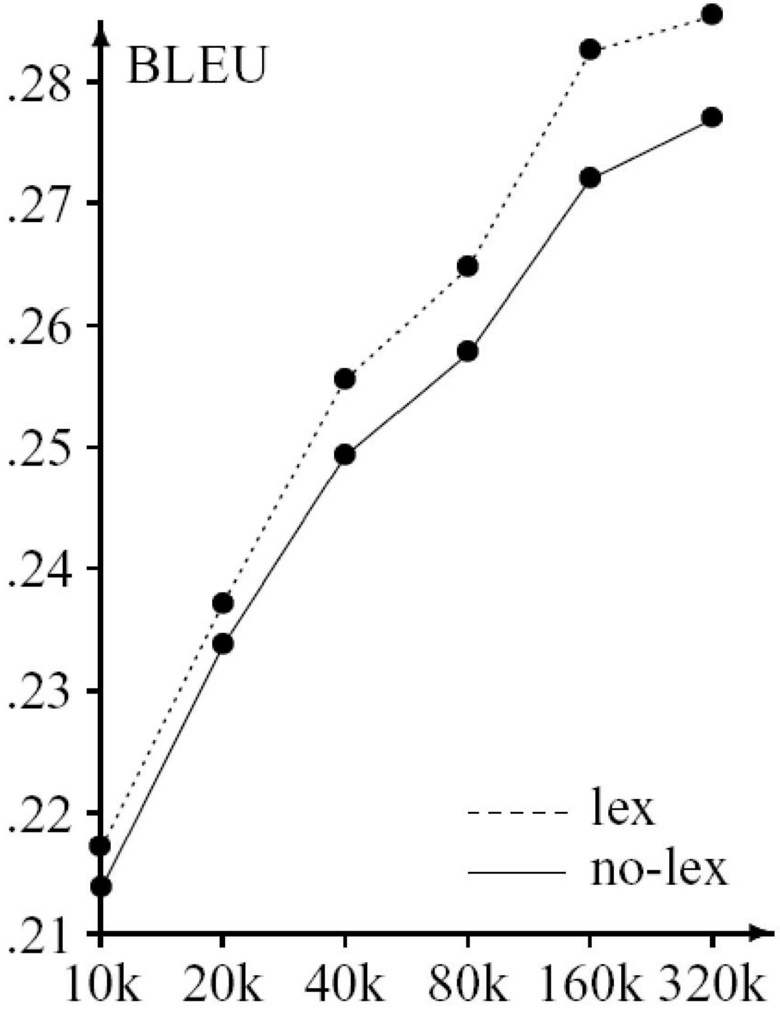


Lexical Weighting

$$\phi(\bar{f}_i|\bar{e}_i) = \frac{count(\bar{f}_i, \bar{e}_i)}{count(\bar{e}_i)} p_w(\bar{f}_i|\bar{e}_i)$$

```
      f1 f2 f3
NULL  -- -- ##
  e1  ## -- --
  e2  -- ## --
  e3  -- ## --
```

$$\begin{aligned} p_w(\bar{f}|\bar{e}, a) &= p_w(f_1 f_2 f_3 | e_1 e_2 e_3, a) \\ &= w(f_1|e_1) \\ &\quad \times \frac{1}{2}(w(f_2|e_2) + w(f_2|e_3)) \\ &\quad \times w(f_3|\text{NULL}) \end{aligned}$$

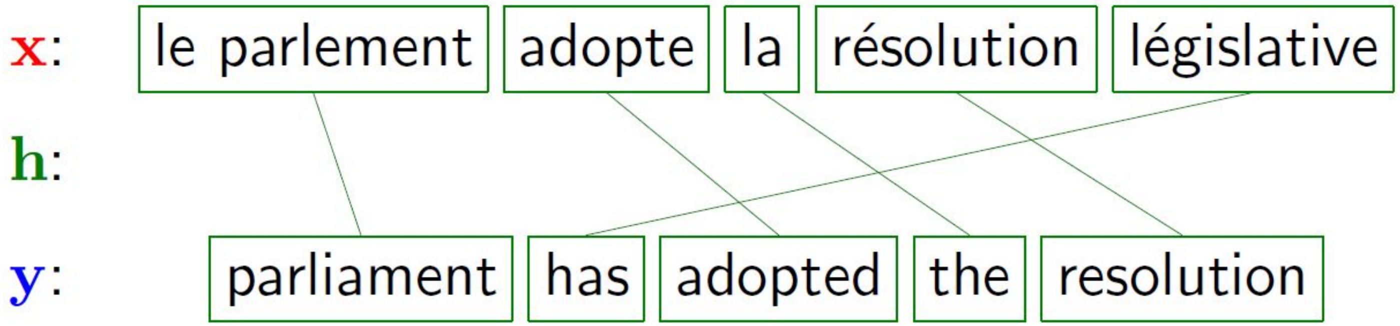


Tuning for MT

- Features encapsulate lots of information
  - Basic MT systems have around 6 features
  - P(e|f), P(f|e), lexical weighting, language model
- How to tune feature weights?
- Idea 1: Use your favorite classifier

Why Tuning is Hard

- Problem 1: There are latent variables
  - Alignments and segmentations

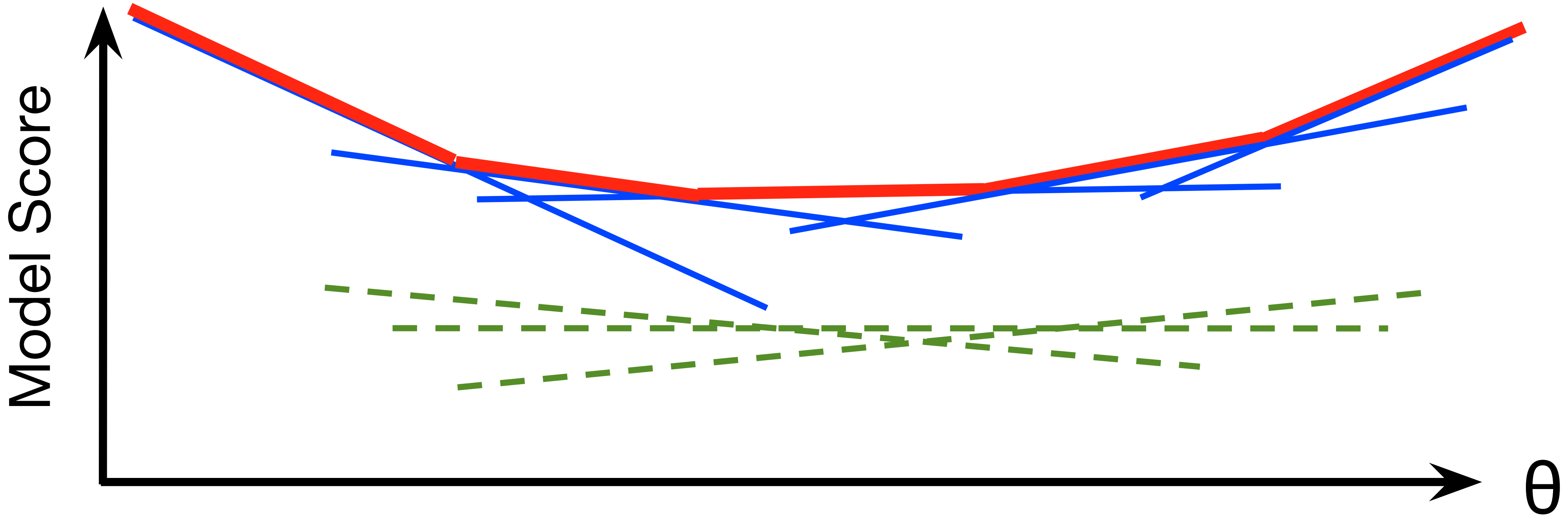


Why Tuning is Hard

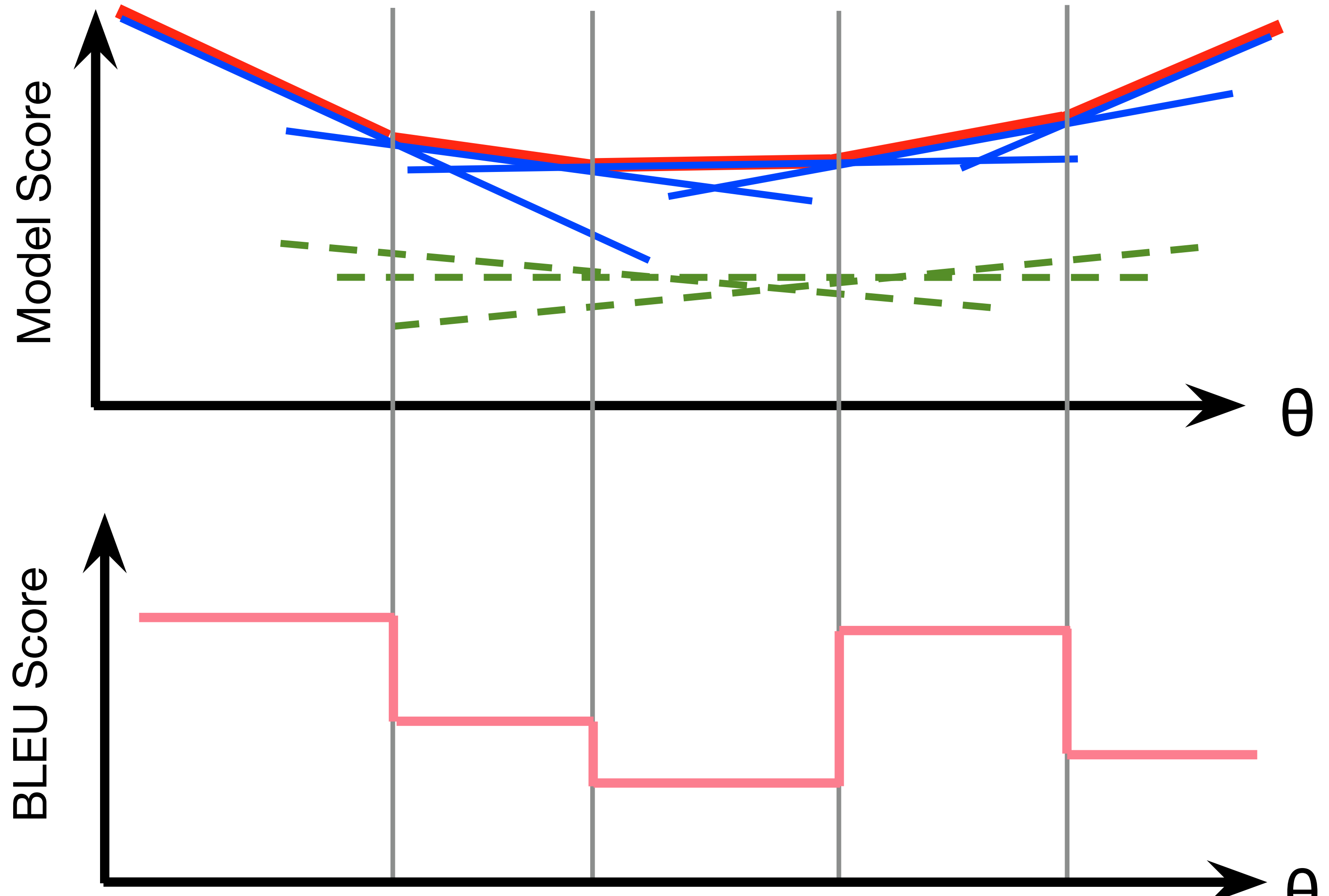
- Problem 3: Computational constraints
  - Discriminative training involves repeated decoding
  - Very slow! So people tune on sets much smaller than those used to build phrase tables

Minimum Error Rate Training

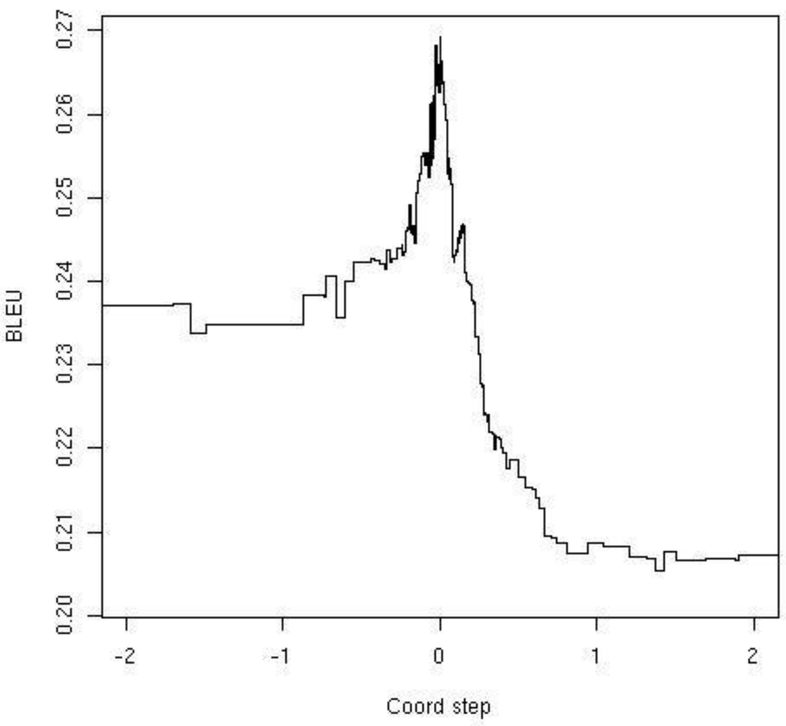
- Standard method: minimize BLEU directly (Och 03)
  - MERT is a discontinuous objective
  - Only works for max ~10 features, but works very well then
  - Here: k-best lists, but forest methods exist (Machery et al 08)
  - Recently, lots of alternatives being explored for more features

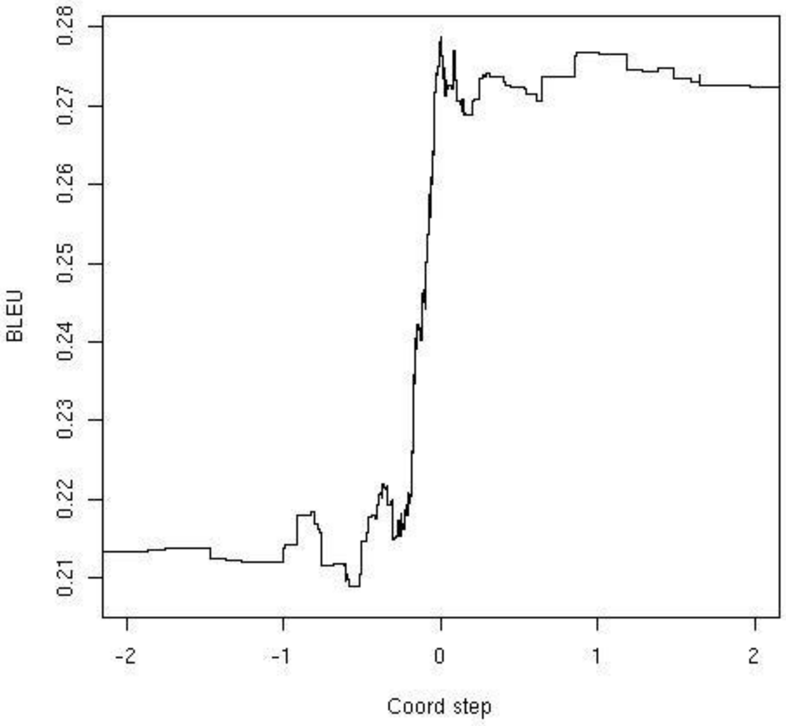


MERT



MERT


BLEU
Coord step
0.20
0.21
0.22
0.23
0.24
0.25
0.26
0.27
-2
-1
0
1
2


BLEU
Coord step
0.21
0.22
0.23
0.24
0.25
0.26
0.27
0.28
-2
-1
0
1
2