

Lexical Translation Models II

February 4, 2014



Last Time ...

$$p(\text{Translation}) = \sum_{\text{Alignment}} p(\text{Alignment}, \text{Translation})$$

$$= \sum_{\text{Alignment}} \underbrace{p(\text{Alignment})}_{\text{Alignment}} \times \underbrace{p(\text{Translation} \mid \text{Alignment})}$$

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

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

Alternate ways of defining
the translation probability

$$\prod_{i=1}^m p(e_i \mid f_{a_i}, f_{a_{i-1}})$$

$$\prod_{i=1}^m p(e_i \mid f_{a_i}, e_{i-1})$$

$$\prod_{i=1}^m p(e_i, e_{i+1} \mid f_{a_i})$$

What is the problem here?

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

Can we do something better here?

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

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

$$\text{Model 2} = \sum_{\mathbf{a} \in [0, n]^m} \prod_{i=1}^m p(a_i \mid i, m, n) \times p(e_i \mid f_{a_i})$$

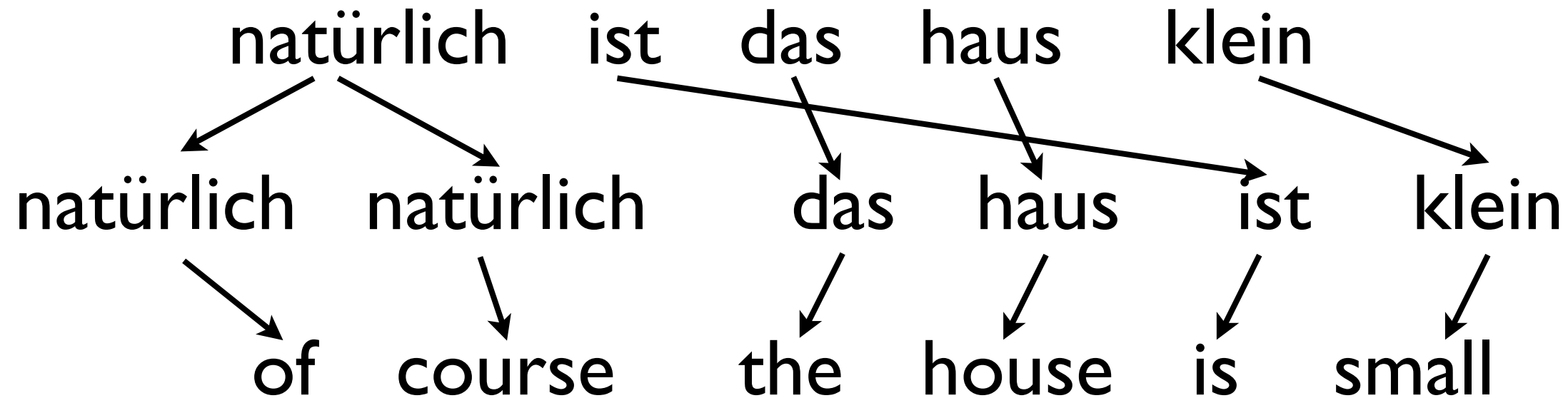
Model 2 $= \sum_{\mathbf{a} \in [0, n]^m} \prod_{i=1}^m p(a_i \mid i, m, n) \times p(e_i \mid f_{a_i})$

- Model alignment with an *absolute position distribution*
- Probability of translating a foreign word at position a_i to generate the word at position i (with target length m and source length n)

$$p(a_i \mid i, m, n)$$

- EM training of this model is almost the same as with Model 1 (same conditional independencies hold)

Model 2 $= \sum_{\mathbf{a} \in [0, n]^m} \prod_{i=1}^m p(a_i \mid i, m, n) \times p(e_i \mid f_{a_i})$



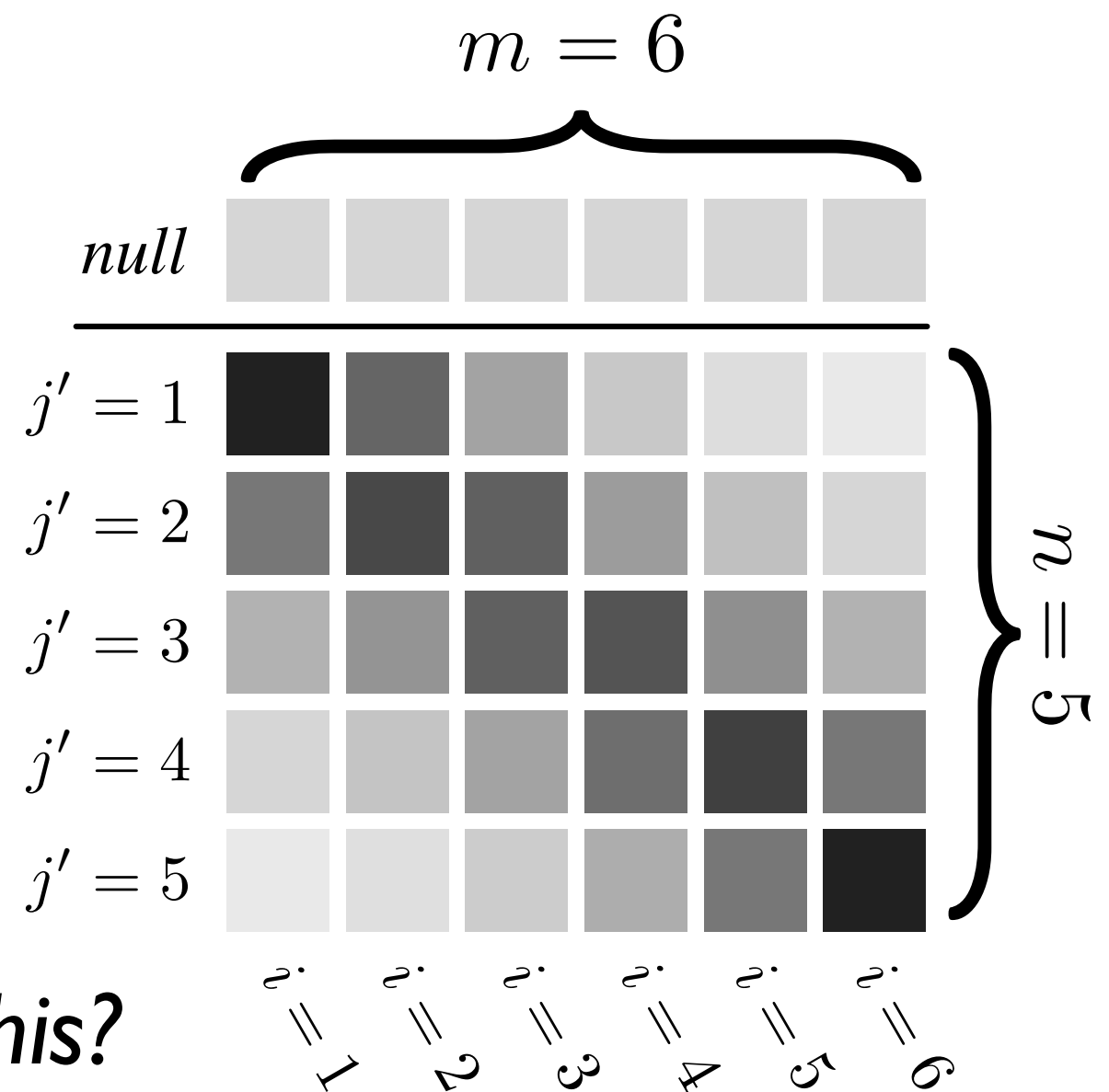
Model 2 $= \sum_{\mathbf{a} \in [0, n]^m} \prod_{i=1}^m p(a_i \mid i, m, n) \times p(e_i \mid f_{a_i})$

- **Pros**
 - Non-uniform alignment model
 - Fast EM training / marginal inference
- **Cons**
 - Absolute position is *very naive*
 - How many parameters to model $p(a_i \mid i, m, n)$

Model 2 $= \sum_{\mathbf{a} \in [0, n]^m} \prod_{i=1}^m p(a_i \mid i, m, n) \times p(e_i \mid f_{a_i})$

How much do we know when we only know the source & target lengths and the current position?

How many parameters do we actually need to model this?



Model 2 $= \sum_{\mathbf{a} \in [0, n]^m} \prod_{i=1}^m p(a_i \mid i, m, n) \times p(e_i \mid f_{a_i})$

pos in target

pos in source

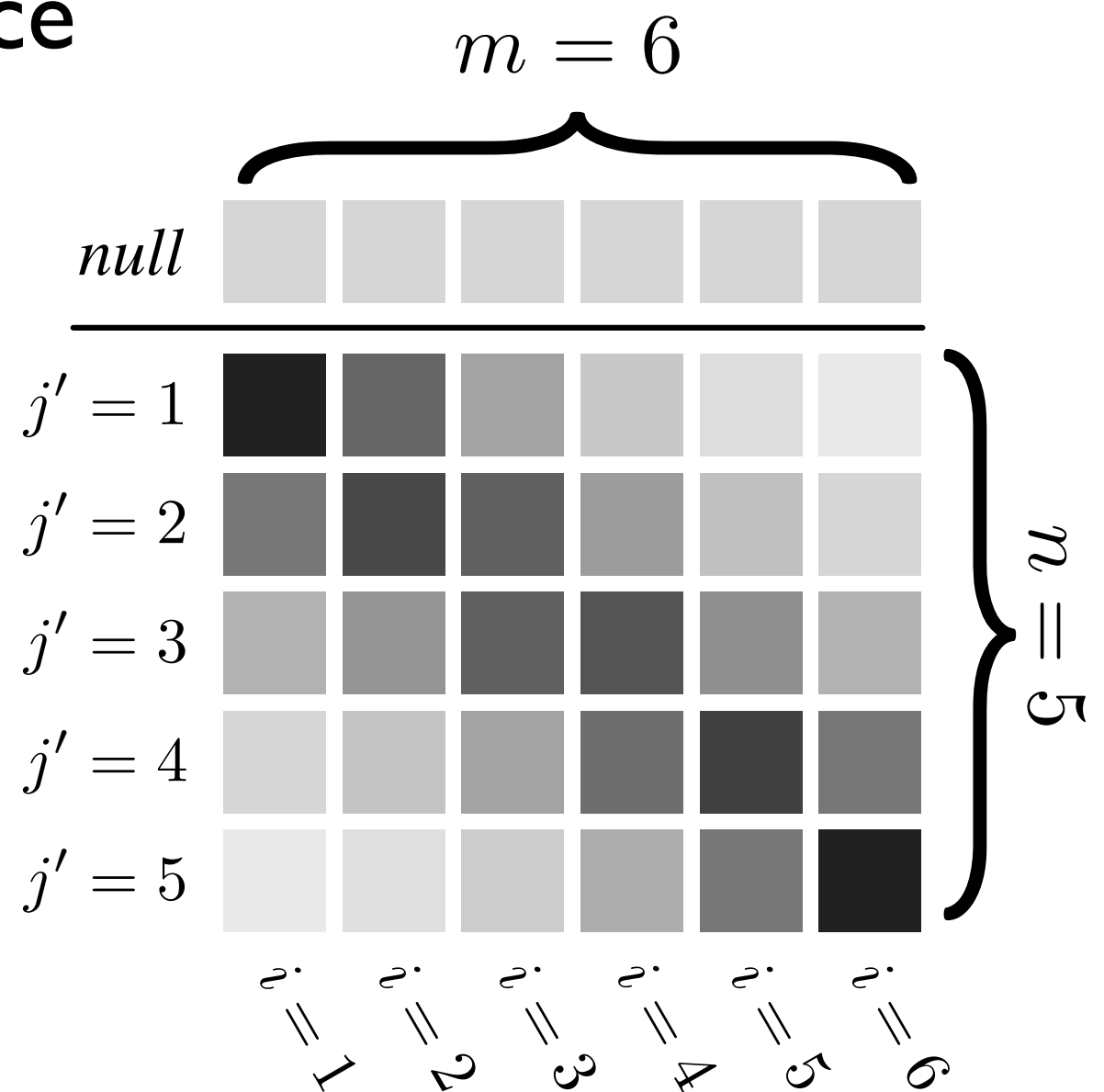
$$h(j, i, m, n) = - \left| \frac{i}{m} - \frac{j}{n} \right|$$

target len

source len

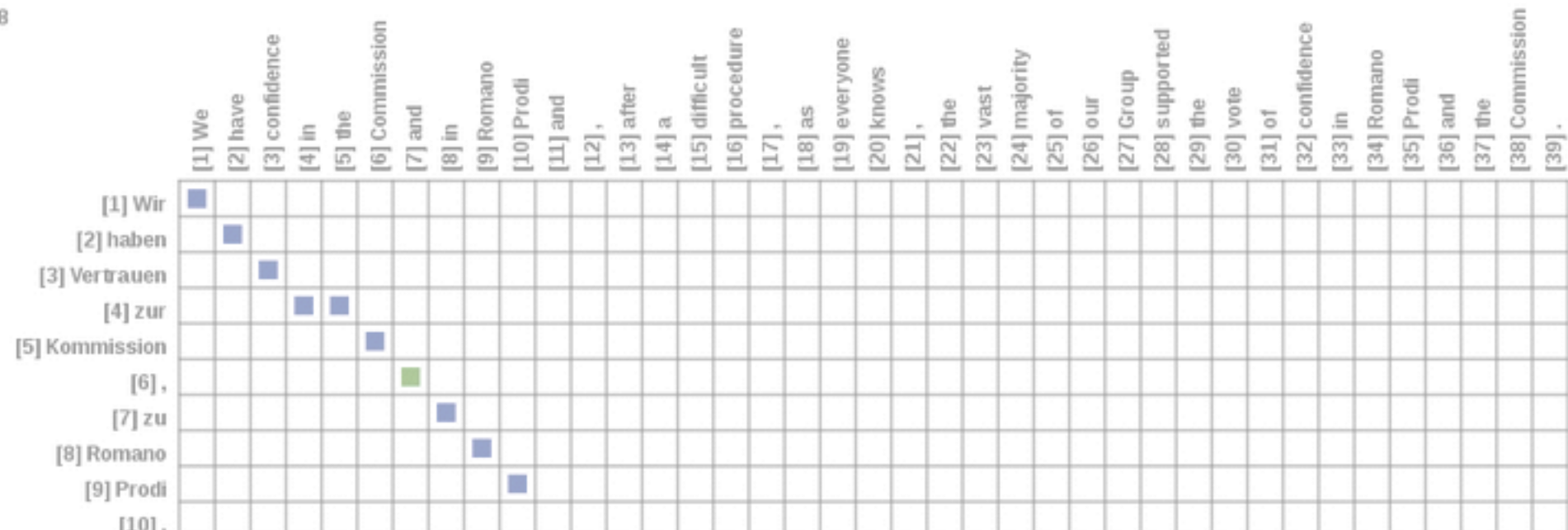
$$b(j \mid i, m, n) = \frac{\exp \lambda h(j, i, m, n)}{\sum_{j'} \exp \lambda h(j', i, m, n)}$$

$$p(a_i \mid i, m, n) = \begin{cases} p_0 & \text{if } a_i = 0 \\ (1 - p_0)b(a_i \mid i, m, n) & \text{otherwise} \end{cases}$$

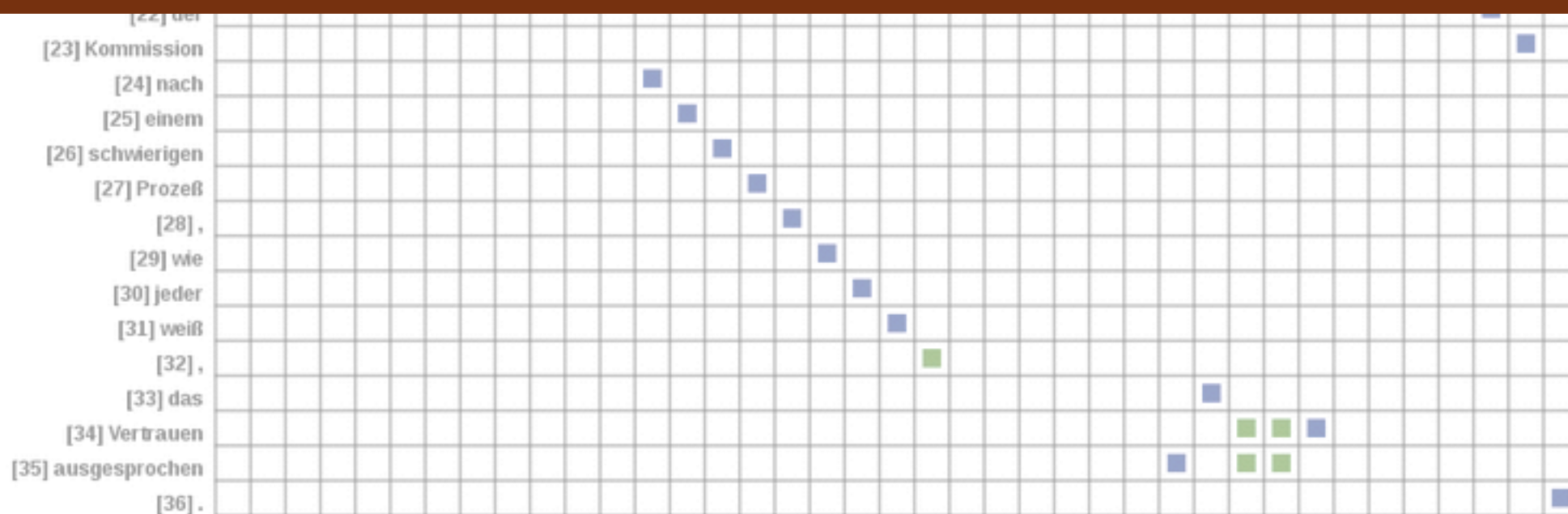


[illegible]

[illegible]



Words reorder in groups. Model this!



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

$$\text{Model 2} = \sum_{\mathbf{a} \in [0, n]^m} \prod_{i=1}^m p(a_i \mid i, m, n) \times p(e_i \mid f_{a_i})$$

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

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

- Insight: words translate in groups
- Condition on previous alignment position
- Probability of translating a foreign word at position a_i given that the previous position translated was a_{i-1}

$$p(a_i \mid a_{i-1})$$

- EM training of this model using forward-backward algorithm (dynamic programming)

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

- Improvement: model “jumps” through the source sentence

$$p(a_i \mid a_{i-1}) = j(a_i - a_{i-1})$$

-4	0.0008
-3	0.0015
-2	0.08
-1	0.18
0	0.0881
1	0.4
2	0.16
3	0.064
4	0.0256

- Relative position model rather than absolute position model

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

- Be careful! NULLs must be handled carefully.
Here is one option (due to Och):

$$p(a_i \mid a_{i-n_i}) = \begin{cases} p_0 & \text{if } a_i = 0 \\ (1 - p_0)j(a_i - a_{i-n_i}) & \text{otherwise} \end{cases}$$

n_i is the index of the first non-null aligned word in the alignment to the left of i .

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

- Other extensions: certain word-types are more likely to be reordered

$$\cancel{j(\delta \mid f)} \quad j(\delta \mid \mathcal{C}(f))$$

Condition the jump probability on the previous word translated

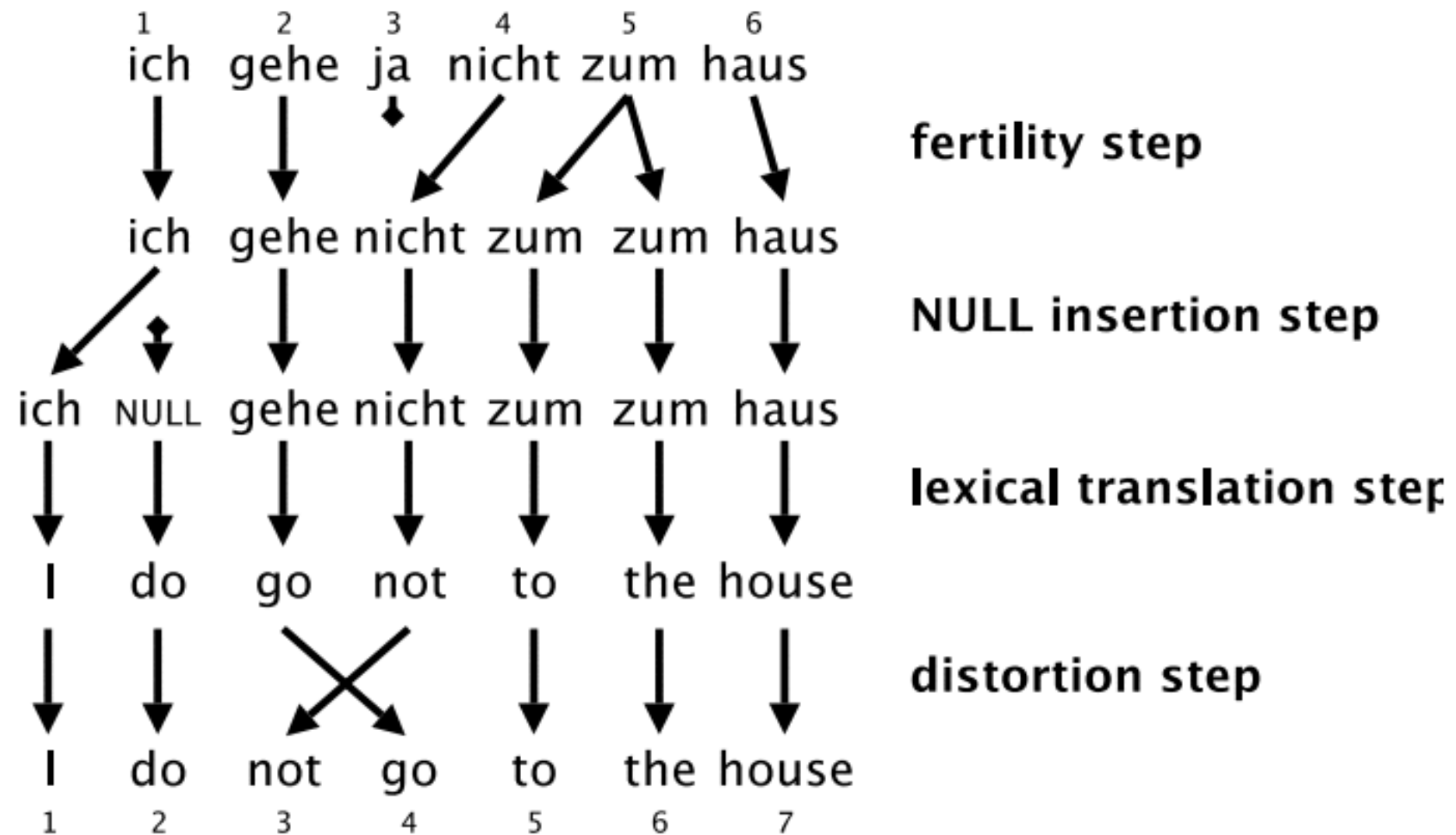
$$\cancel{j(\delta \mid f, e)} \quad j(\delta \mid \mathcal{A}(f), \mathcal{B}(e))$$

Condition the jump probability on the previous word translated, and **how** it was translated

Fertility Models

- The models we have considered so far have been efficient
- This efficiency has come at a modeling cost:
 - What is to stop the model from “translating” a word 0, 1, 2, or 100 times?
- We introduce *fertility models* to deal with this

IBM Model 3



Fertility

- Fertility: the number of English words generated by a foreign word
- Modeled by categorical distribution $n(\phi \mid f)$
- Examples:

Unabhaengigkeitserklaerung

0	0.00008
1	0.1
2	0.0002
3	0.8
4	0.009
5	0

zum = (zu + dem)

0	0.01
1	0
2	0.9
3	0.0009
4	0.0001
5	0

Haus

0	0.01
1	0.92
2	0.07
3	0
4	0
5	0

Fertility

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

- Fertility models mean that we can no longer exploit conditional independencies to write $p(\mathbf{a} \mid \mathbf{f}, m)$ as a series of local alignment decisions.
- *How do we compute the statistics required for EM training?*

EM Recipe reminder

- If alignment points were visible, training fertility models would be easy
- We would _____ and _____

$$n(\phi = 3 \mid f = \textit{Unabhaenigkeitserklaerung}) = \frac{\text{count}(3, \textit{Unabhaenigkeitserklaerung})}{\text{count}(\textit{Unabhaenigkeitserklaerung})}$$

- But, alignments are not visible

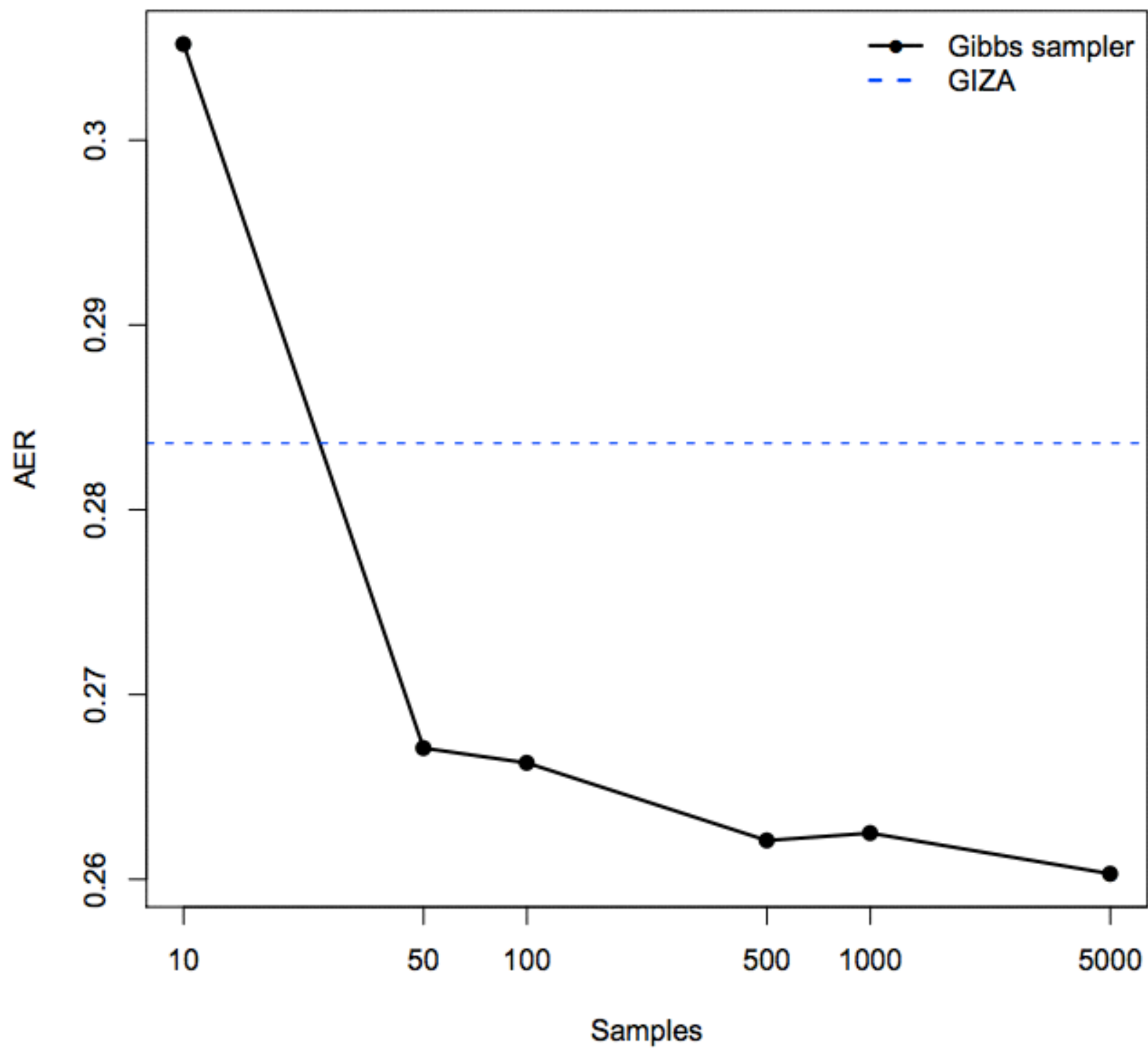
$$n(\phi = 3 \mid f = \textit{Unabhaenigkeitserklaerung}) = \frac{\mathbb{E}[\text{count}(3, \textit{Unabhaenigkeitserklaerung})]}{\mathbb{E}[\text{count}(\textit{Unabhaenigkeitserklaerung})]}$$

Expectation & Fertility

- We need to compute expected counts under $p(a \mid f, e, m)$
- Unfortunately $p(a \mid f, e, m)$ doesn't factorize nicely. :(
- Can we sum exhaustively? How many different a 's are there?
 - What to do?

Sampling Alignments

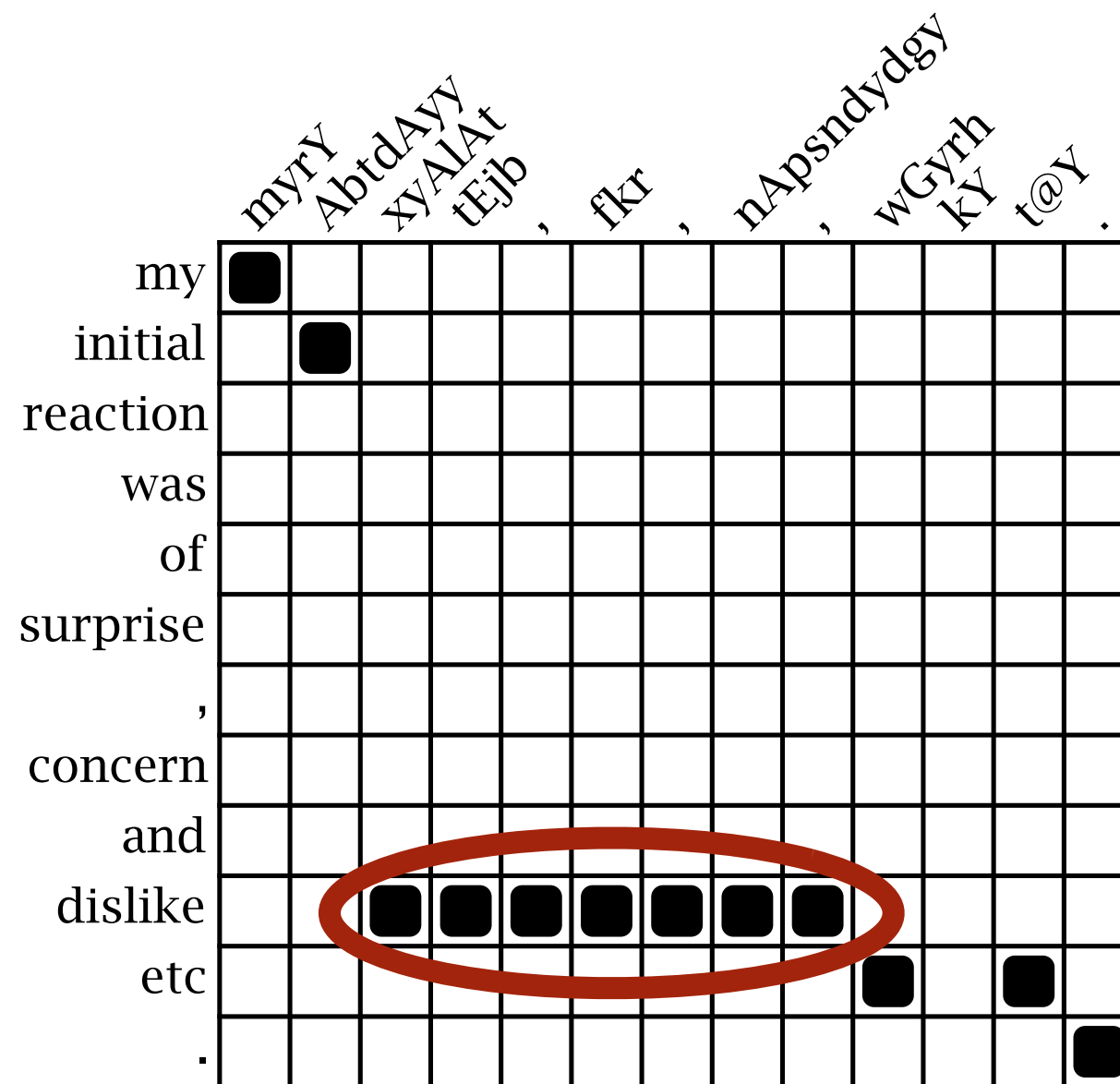
- Monte-Carlo methods
 - Gibbs sampling
 - Importance sampling
 - Particle filtering
- For historical reasons
 - Use model 2 alignment to start (easy!)
 - Weighted sum over all alignment configurations that are “close” to this alignment configuration
 - Is this correct? No! Does it work? Sort of.



Lexical Translation

- IBM Models 1-5 [Brown et al., 1993]
 - Model 1: lexical translation, uniform alignment
 - Model 2: absolute position model
 - Model 3: fertility
 - Model 4: relative position model (jumps in target string)
 - Model 5: non-deficient model
- HMM translation model [Vogel et al., 1996]
 - Relative position model (jumps in source string)
- Latent variables are more useful these days than the translations
- Widely used Giza++ toolkit

Pitfalls of Conditional Models



IBM Model 4 alignment

A few tricks...

$$p(f|e)$$

	michael	geht	davon	aus	,	dass	er	im	haus	bleibt
michael	1									
assumes		1	1	1						
that						1				
he							1			
will										
stay										1
in								1		
the										
house									1	

English to German

A few tricks...

$p(f|e)$

	michael	geht	davon	aus	.	dass	er	im	haus	bleibt
michael	■									
assumes		■	■	■						
that						■				
he							■			
will										
stay										■
in							■			
the										
house									■	

English to German

	michael	geht	davon	aus	.	dass	er	im	haus	bleibt
michael	■									
assumes		■								
that						■				
he							■			
will										■
stay										
in							■			
the										
house									■	

German to English

$p(e|f)$

A few tricks...

$p(f|e)$

	michael	geht	davon	aus	.	dass	er	im	haus	bleibt
michael	■									
assumes		■	■	■						
that						■				
he							■			
will										
stay									■	
in							■			
the										
house									■	

English to German

	michael	geht	davon	aus	.	dass	er	im	haus	bleibt
michael	■									
assumes		■								
that						■				
he							■			
will										■
stay										
in							■			
the										
house									■	

German to English

$p(e|f)$

	michael	geht	davon	aus	.	dass	er	im	haus	bleibt
michael	■									
assumes		■	■	■						
that						■				
he							■			
will										■
stay									■	
in							■			
the										
house									■	

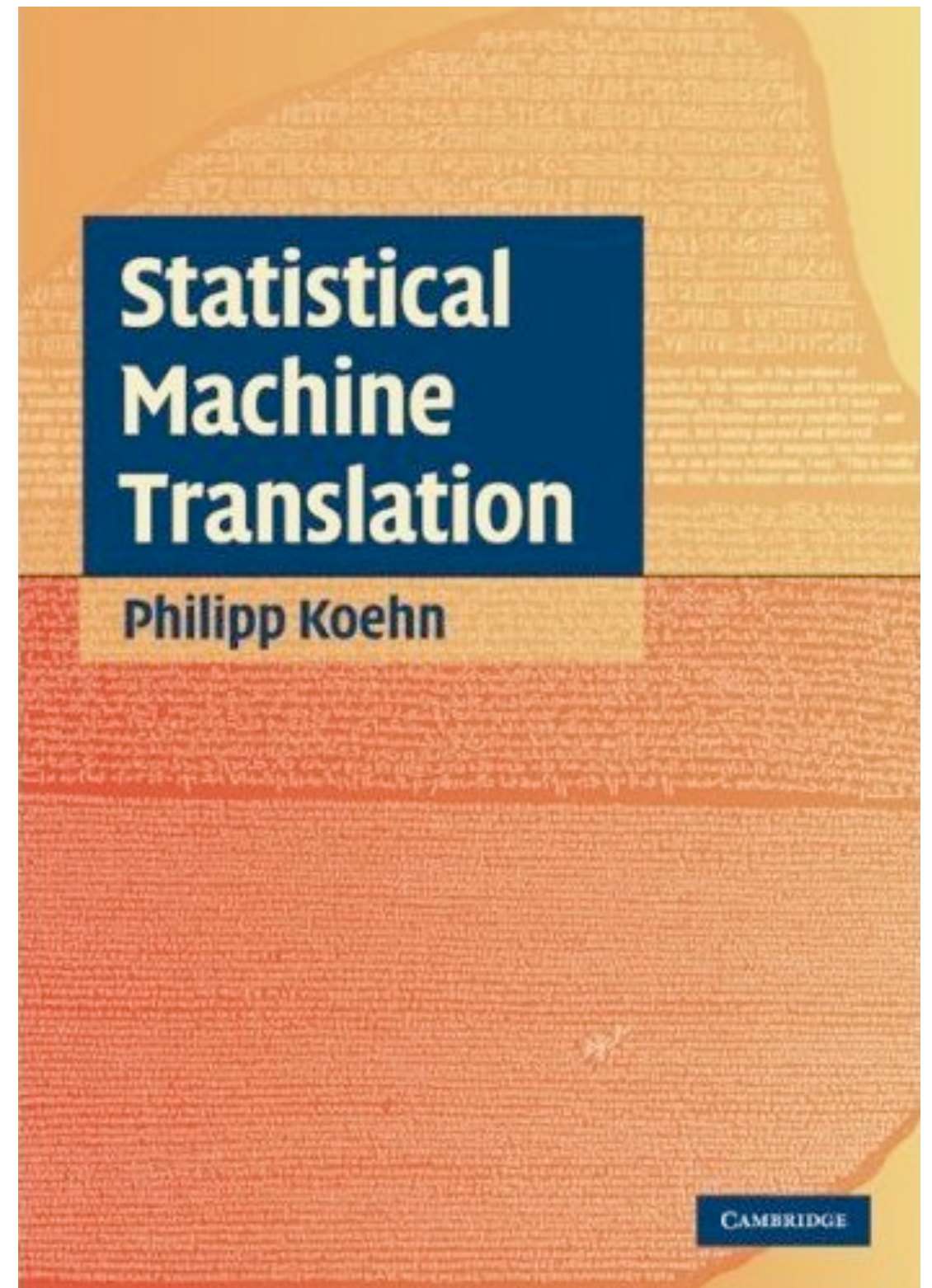
Intersection / Union

Suggestions for HWI

- Implement IBM Model 2 in addition to IBM Model 1
- Try the heuristics for merging the many-to-one and one-to-many alignments
- Try to reduce sparse counts by pre-processing your training data
- Other ideas?

Reading

- Read Chapter 4 from the textbook (today we covered 4.4 through 4.6)



Announcements

- HW1 due on Thursday by 11:59pm
- To get an A you'll need to beat the baseline IBM Model 1 performance
- Jonny will hold extended office hours on Wednesday (1-3, and by appointment if you can't make that time)
- Upcoming language-in-10
 - Thursday: Emily - Swedish
 - Tuesday: Mary - Irish/Gaelic