# Lexical Translation Models II

**Machine Translation**
**Lecture 5**

**Instructor: Chris Callison-Burch**
**TAs: Mitchell Stern, Justin Chiu**

**Website: mt-class.org/penn**

# Last Time ...

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

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

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

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

Alternate ways of defining the translation probability

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

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

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

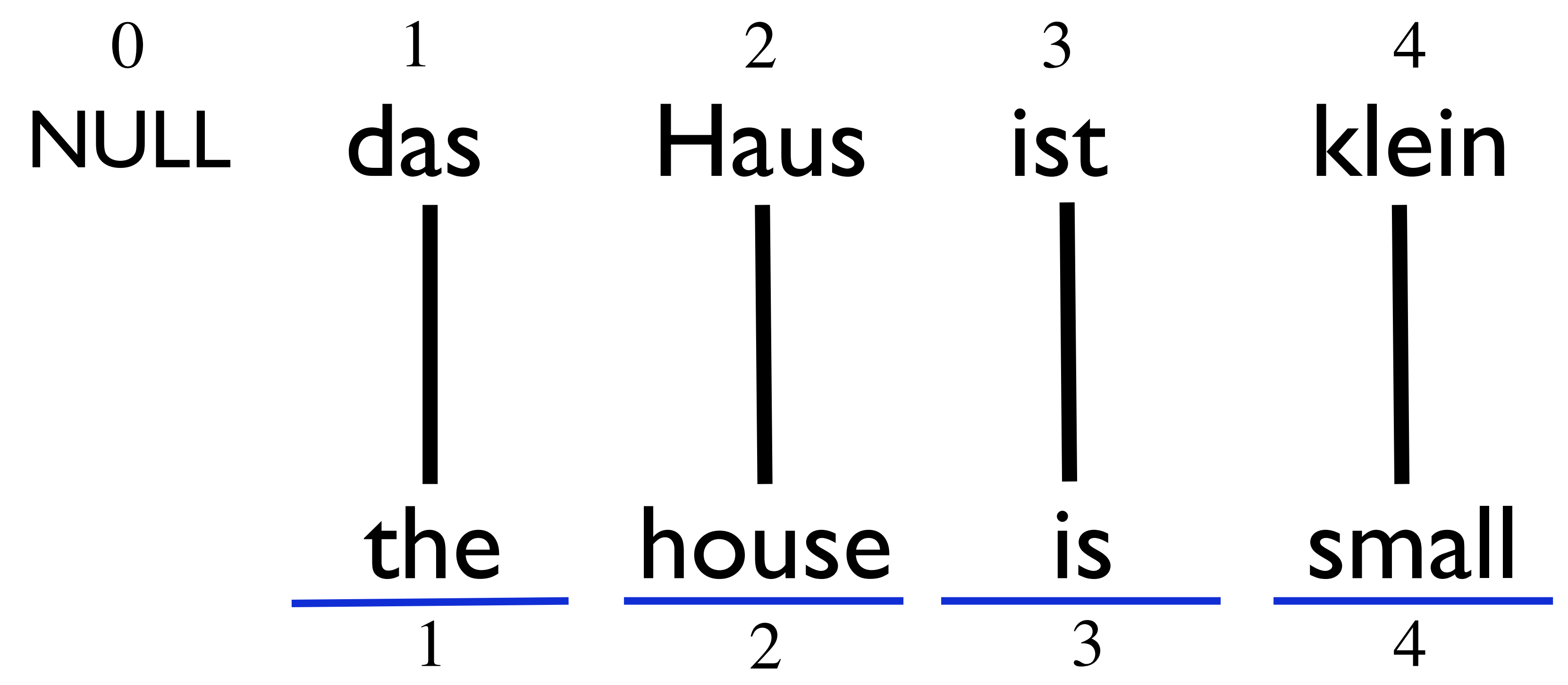
*What is the problem here?*

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

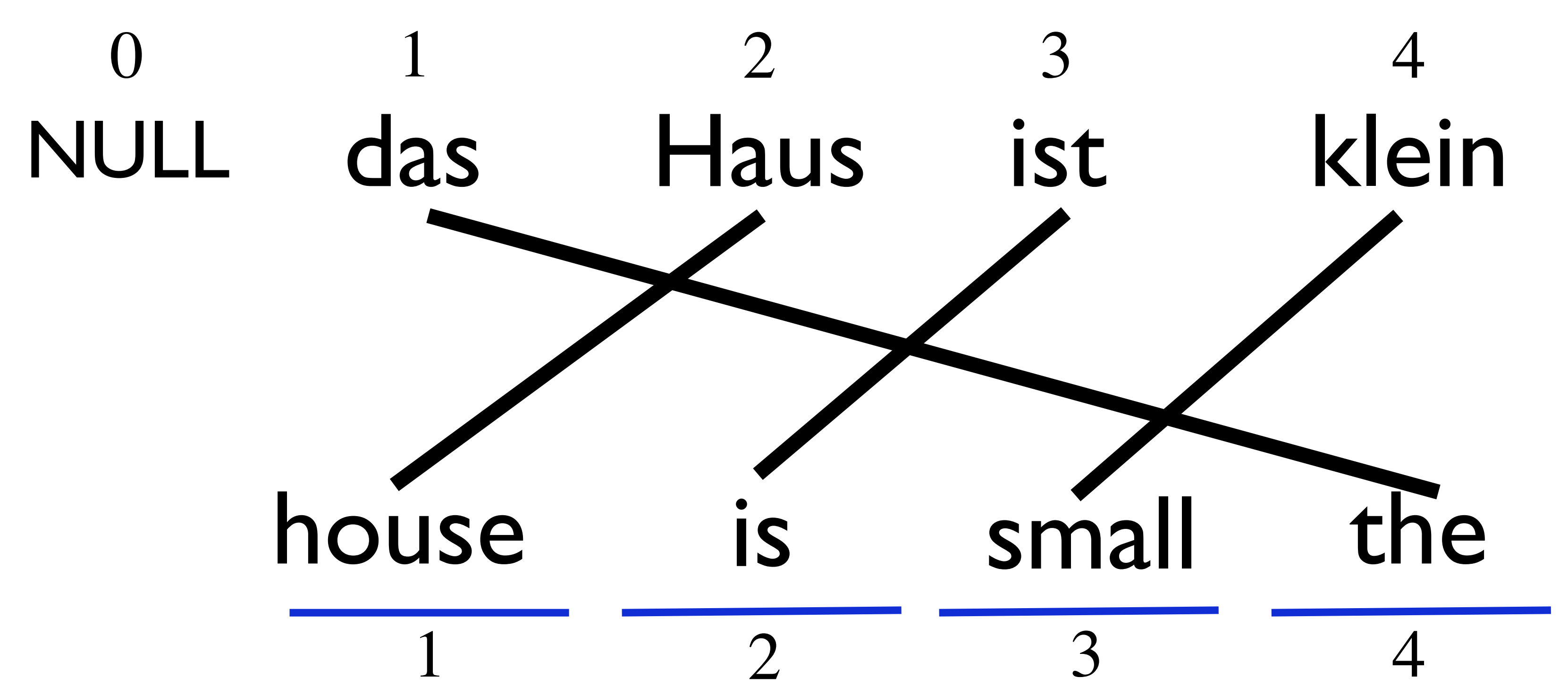
$$= \sum_{\mathbf{a} \in [0,n]^m} \underbrace{\prod_{i=1}^{m} \frac{1}{1+n}}_{p(\mathbf{a} \mid \mathbf{f}, m)} \times \prod_{i=1}^{m} p(e_i \mid f_{a_i})$$

***Can we do something better here?***

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



0
NULL
1
das
2
Haus
3
ist
4
klein
the
house
is
small
1
2
3
4

0
NULL
1
das
2
Haus
3
ist
4
klein
house
is
small
the
1
2
3
4

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

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

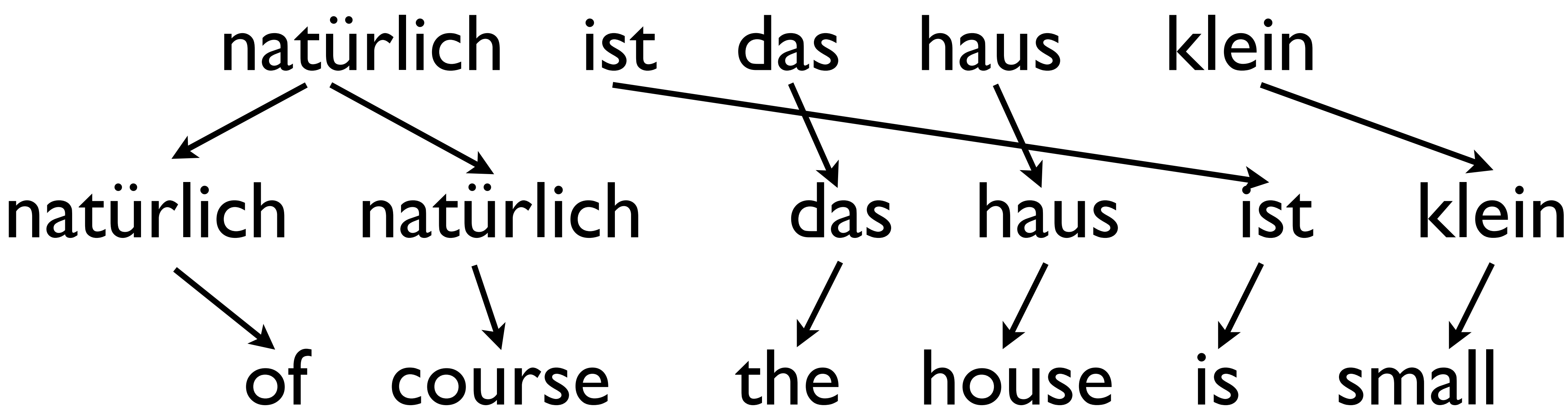
## Model 2 $= \sum_{\mathbf{a} \in [0,n]^m} \prod_{i=1}^{m} p(a_i \mid i, m, n) \times p(e_i \mid f_{a_i})$

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

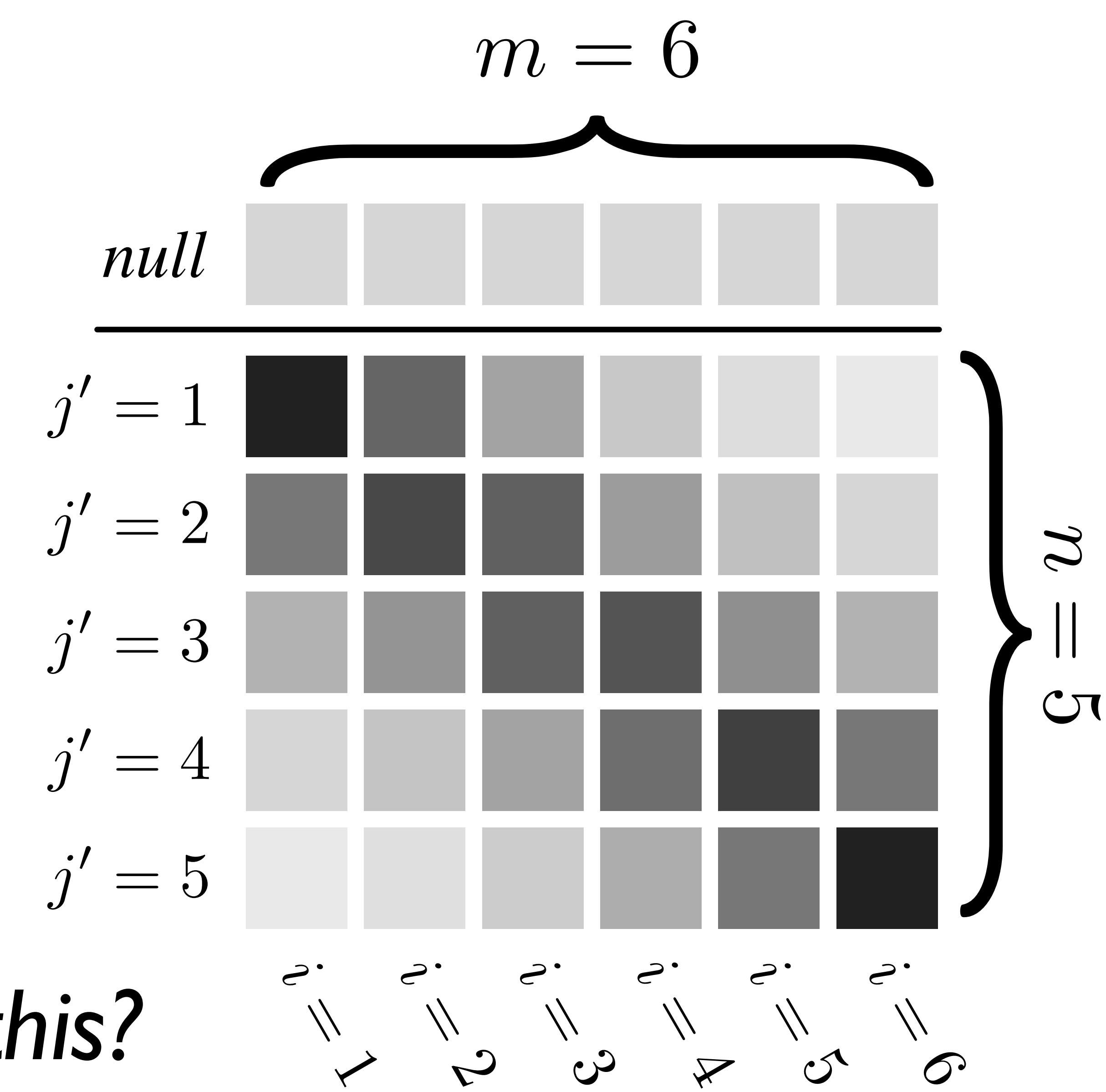


**Model 2** $= \sum_{\mathbf{a} \in [0,n]^m} \prod_{i=1}^{m} p(a_i \mid i, m, n) \times p(e_i \mid f_{a_i})$

- **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

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

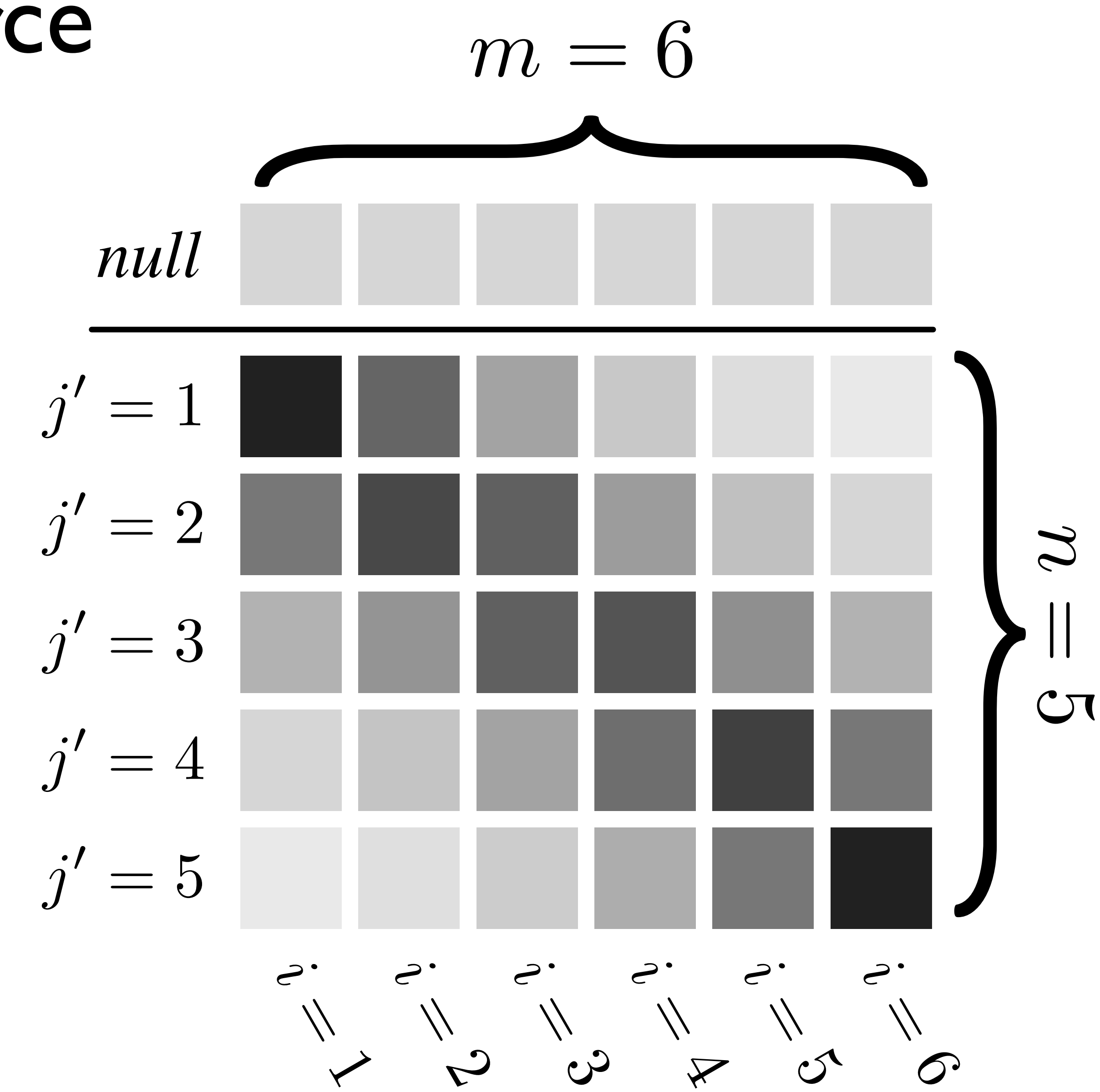
pos in target

pos in source

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

target len

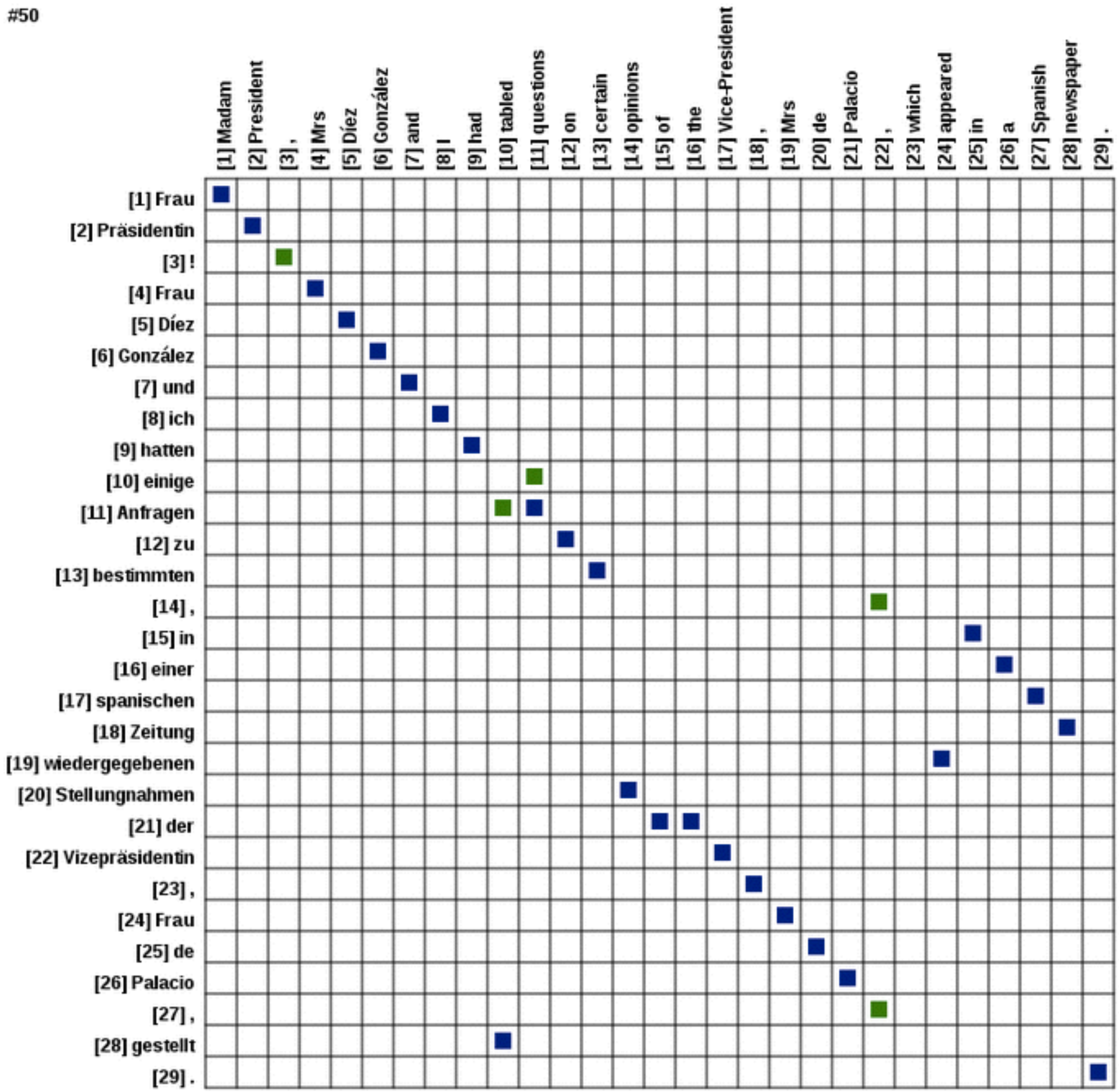
source len



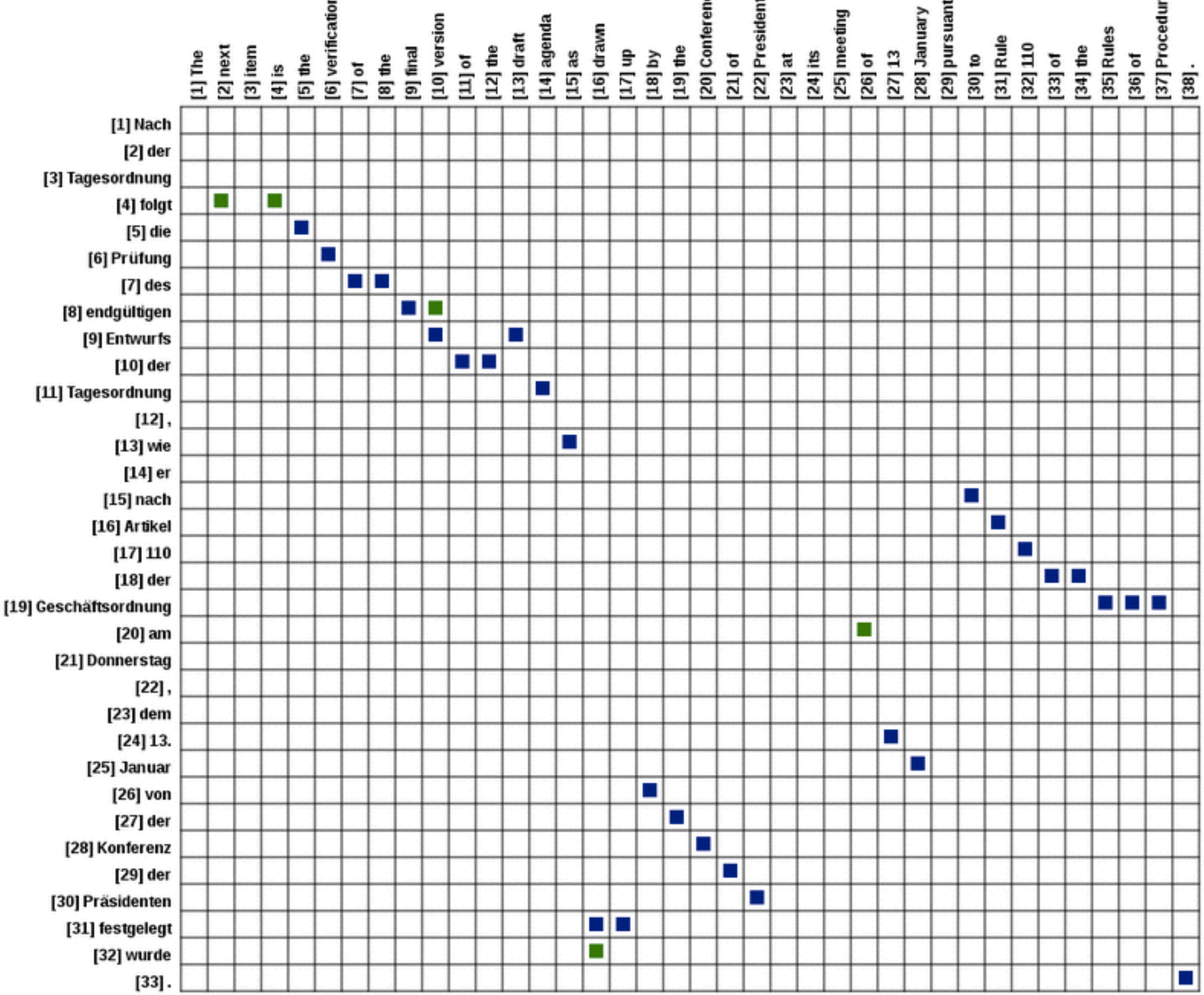

$$b(j \mid i, m, n) = \frac{\exp \lambda h(j, i, m, n)}{\sum_{j'} \exp \lambda h(j', i, m, n)}$$

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

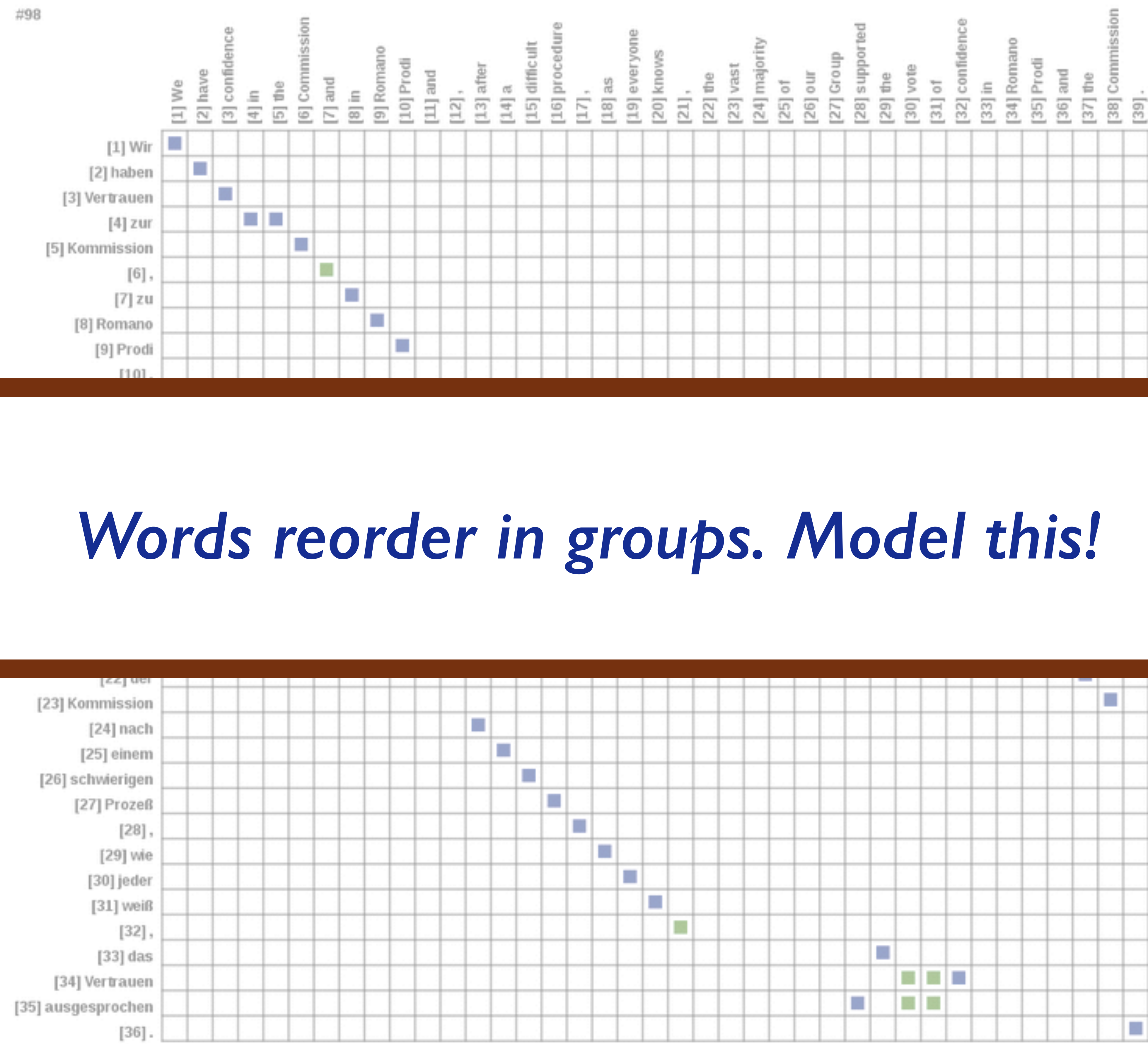
#50


[1] Madam
[2] President
[3] ,
[4] Mrs
[5] Díez
[6] González
[7] and
[8] I
[9] had
[10] tabled
[11] questions
[12] on
[13] certain
[14] opinions
[15] of
[16] the
[17] Vice-President
[18] ,
[19] Mrs
[20] de
[21] Palacio
[22] ,
[23] which
[24] appeared
[25] in
[26] a
[27] Spanish
[28] newspaper
[29] .
[1] Frau
[2] Präsidentin
[3] !
[4] Frau
[5] Díez
[6] González
[7] und
[8] ich
[9] hatten
[10] einige
[11] Anfragen
[12] zu
[13] bestimmten
[14] ,
[15] in
[16] einer
[17] spanischen
[18] Zeitung
[19] wiedergegebenen
[20] Stellungnahmen
[21] der
[22] Vizepräsidentin
[23] ,
[24] Frau
[25] de
[26] Palacio
[27] ,
[28] gestellt
[29] .

#67



[1] The
[2] next
[3] item
[4] is
[5] the
[6] verification
[7] of
[8] the
[9] final
[10] version
[11] of
[12] the
[13] draft
[14] agenda
[15] as
[16] drawn
[17] up
[18] by
[19] the
[20] Conference
[21] of
[22] Presidents
[23] at
[24] its
[25] meeting
[26] of
[27] 13
[28] January
[29] pursuant
[30] to
[31] Rule
[32] 110
[33] of
[34] the
[35] Rules
[36] of
[37] Procedure
[38] .
[1] Nach
[2] der
[3] Tagesordnung
[4] folgt
[5] die
[6] Prüfung
[7] des
[8] endgültigen
[9] Entwurfs
[10] der
[11] Tagesordnung
[12] ,
[13] wie
[14] er
[15] nach
[16] Artikel
[17] 110
[18] der
[19] Geschäftsordnung
[20] am
[21] Donnerstag
[22] ,
[23] dem
[24] 13.
[25] Januar
[26] von
[27] der
[28] Konferenz
[29] der
[30] Präsidenten
[31] festgelegt
[32] wurde
[33] .




*Words reorder in groups. Model this!*

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

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

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

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

- 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)

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

- 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

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

- 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 $i$.

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

- 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

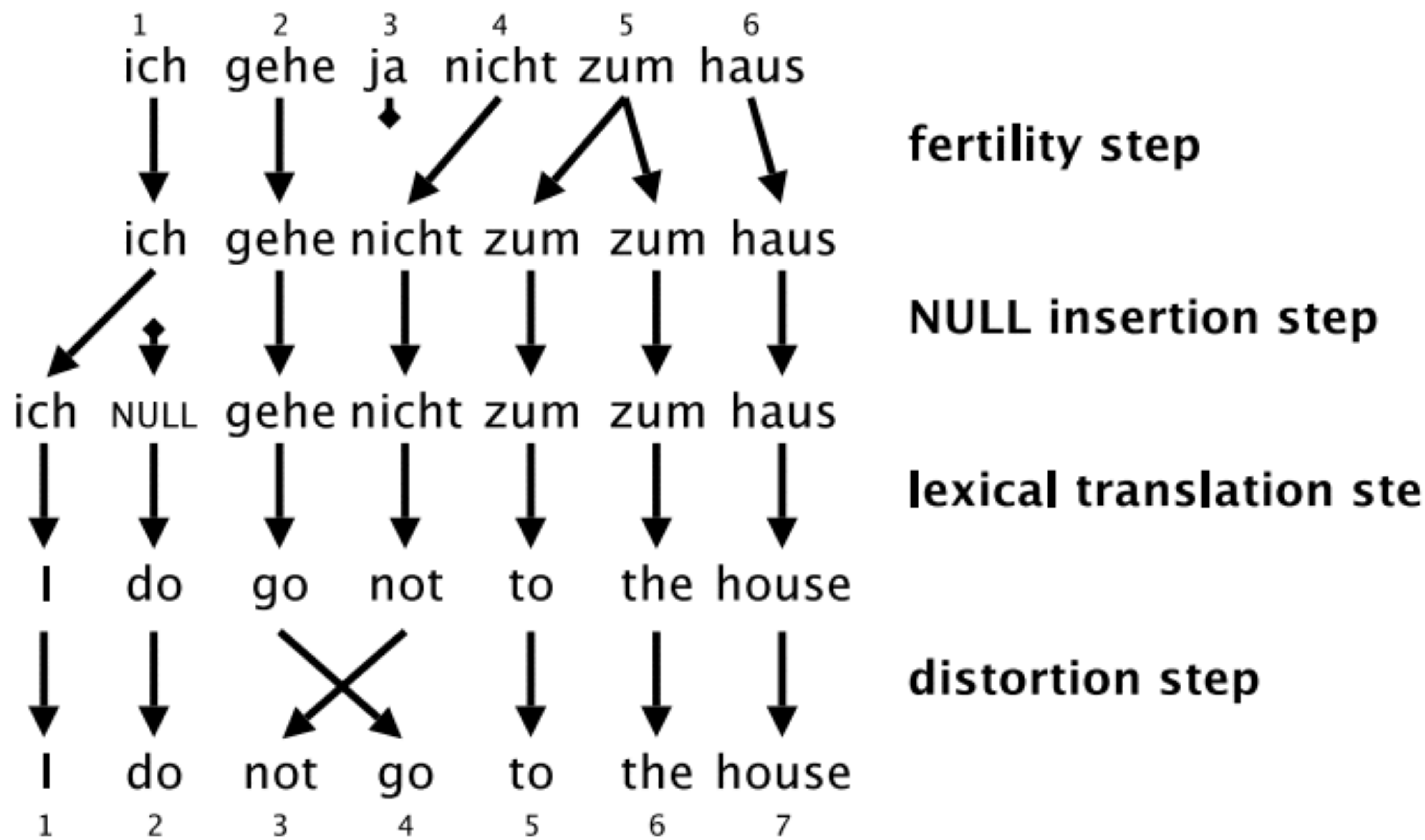
$\cancel{j(\delta \mid f, e)}$ $\quad j(\delta \mid \mathcal{A}(f), \mathcal{B}(e))$

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


1 2 3 4 5 6
ich gehe ja nicht zum haus
fertility step
ich gehe nicht zum zum haus
NULL insertion step
ich NULL gehe nicht zum zum haus
lexical translation step
I do go not to the house
distortion step
I do not go to the house
1 2 3 4 5 6 7

# 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 __________

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

- But, alignments are not visible

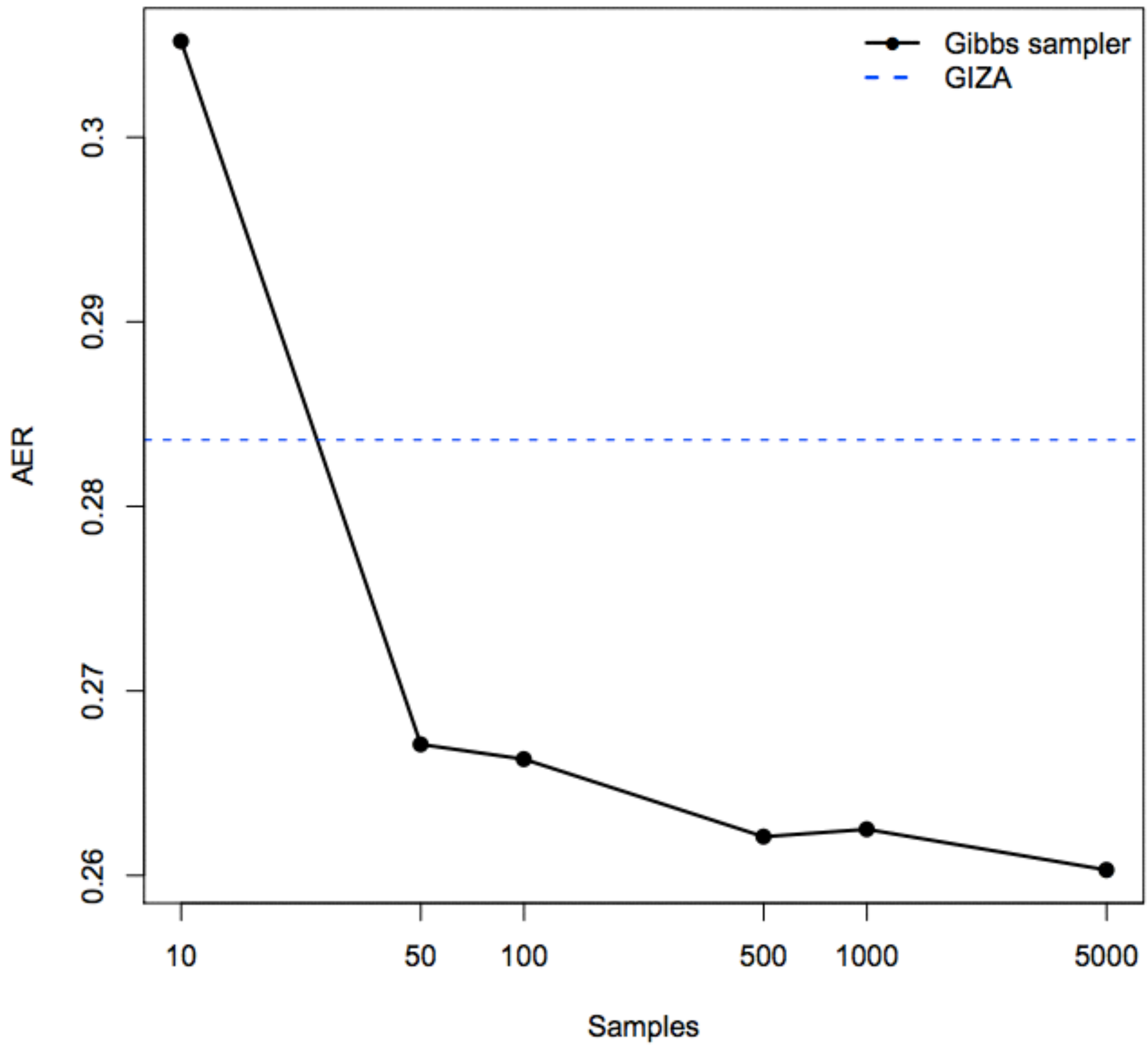
$$n(\phi = 3 \mid f = Unabhaenigkeitserklaerung) = \frac{\mathbb{E}[\text{count}(3, Unabhaenigkeitserklaerung)]}{\mathbb{E}[\text{count}(Unabhaenigkeitserklaerung)]}$$

# Expectation & Fertility

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

# Sampling Alignments

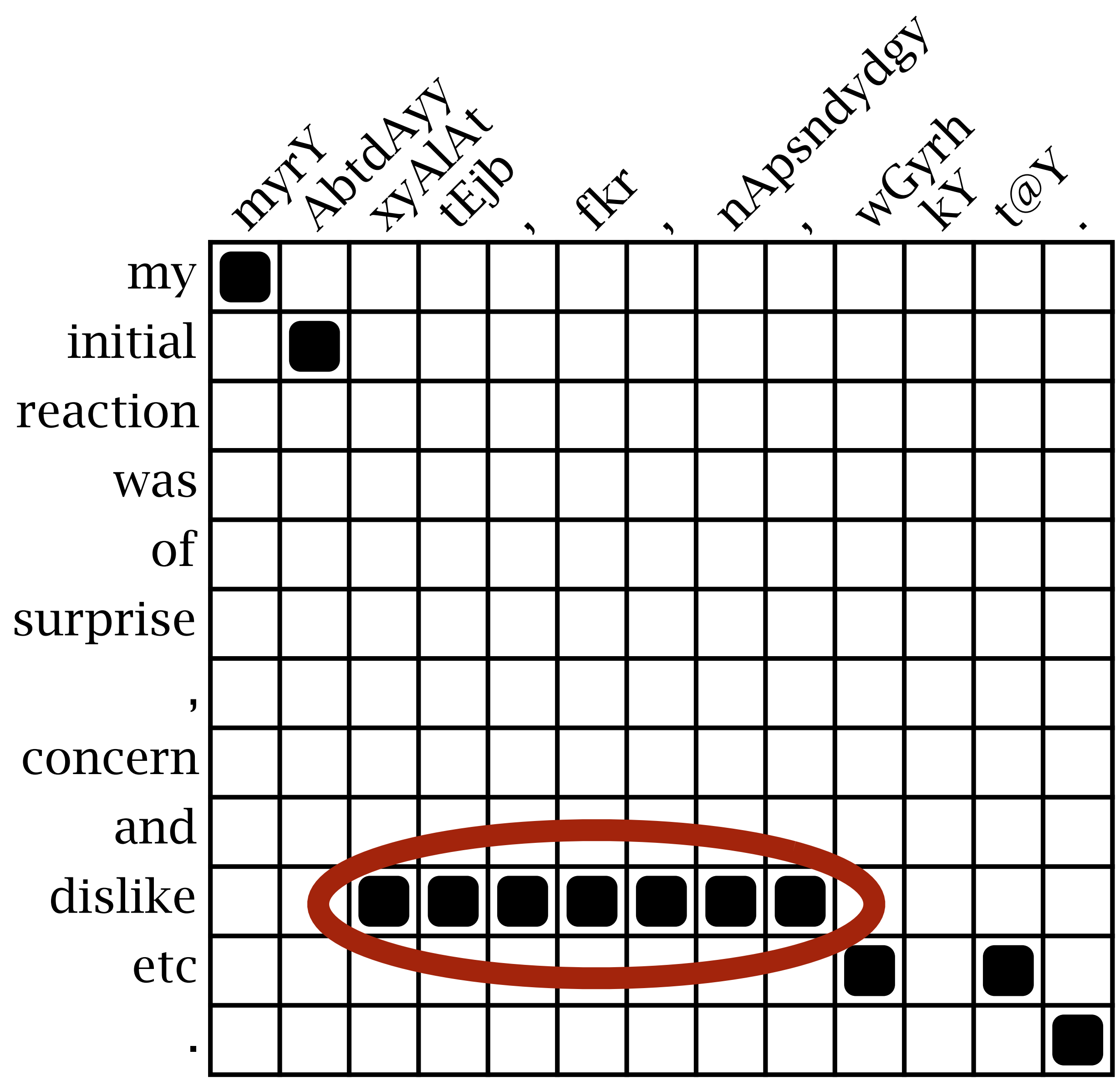
- Monte-Carlo methods
  - Gibbs sampling
  - Importance sampling
  - Particle filtering
- For historical reasons
  - Use model 2 alignment to start (easy!)
  - Weighted sum over all alignment configurations that are "close" to this alignment configuration
  - Is this correct? No! Does it work? Sort of.


Gibbs sampler
GIZA
AER
0.26
0.27
0.28
0.29
0.3
10
50
100
500
1000
5000
Samples

# 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

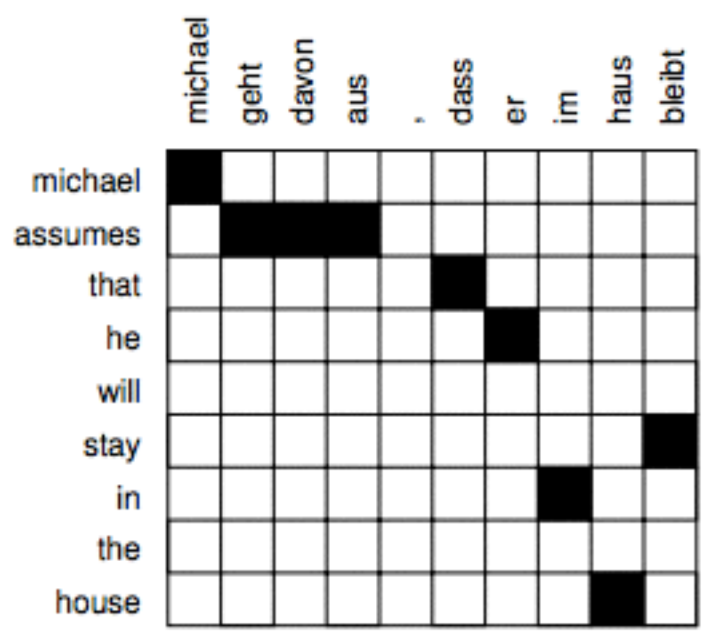
# Pitfalls of Conditional Models




IBM Model 4 alignment

# A few tricks...

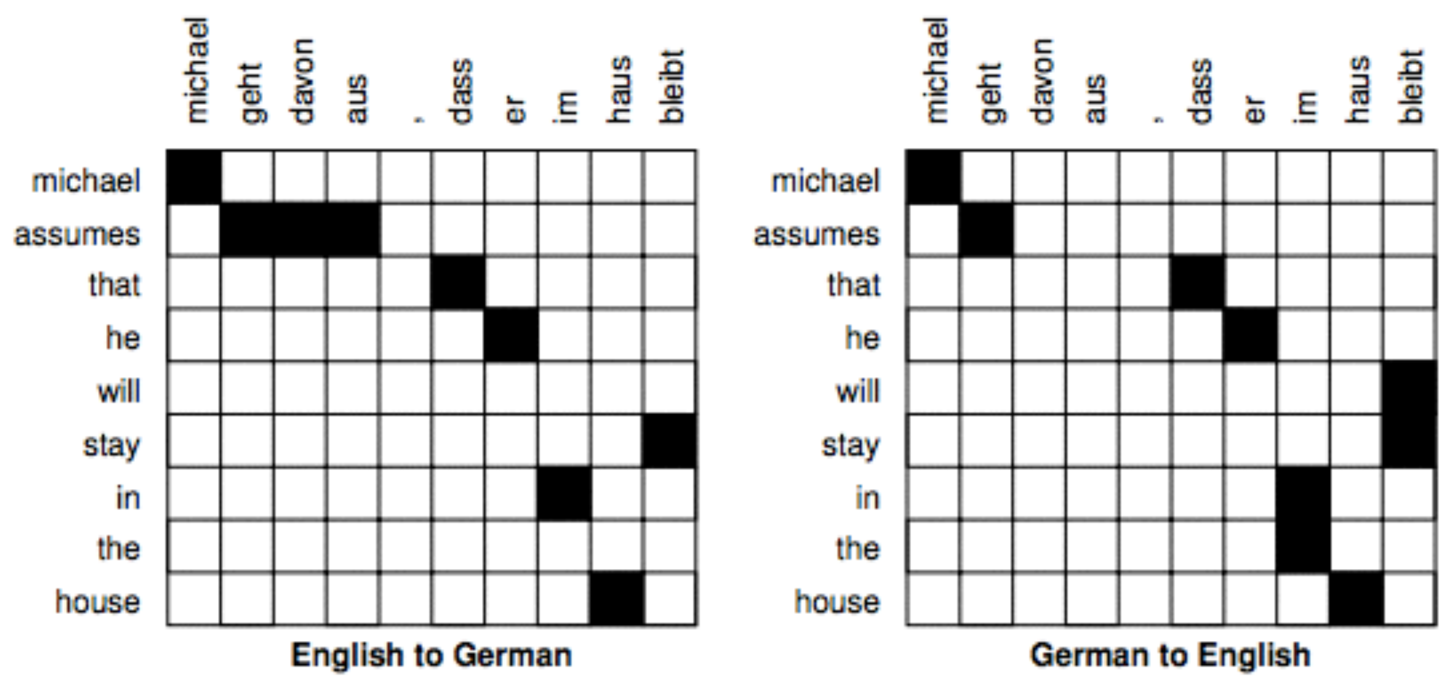
p(f|e)




English to German

# A few tricks...

p(f|e)

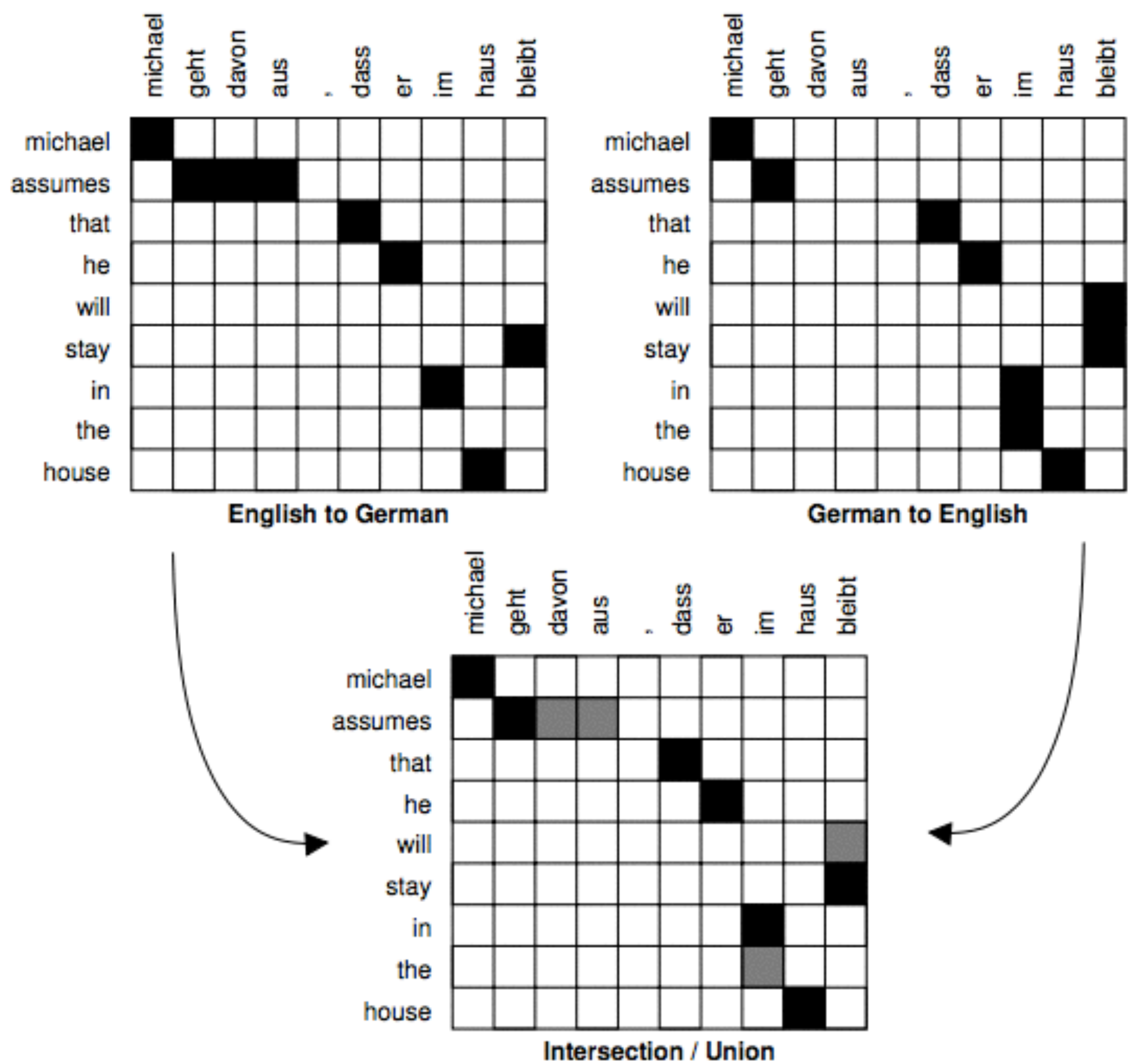



p(e|f)

# A few tricks...

p(f|e)

p(e|f)

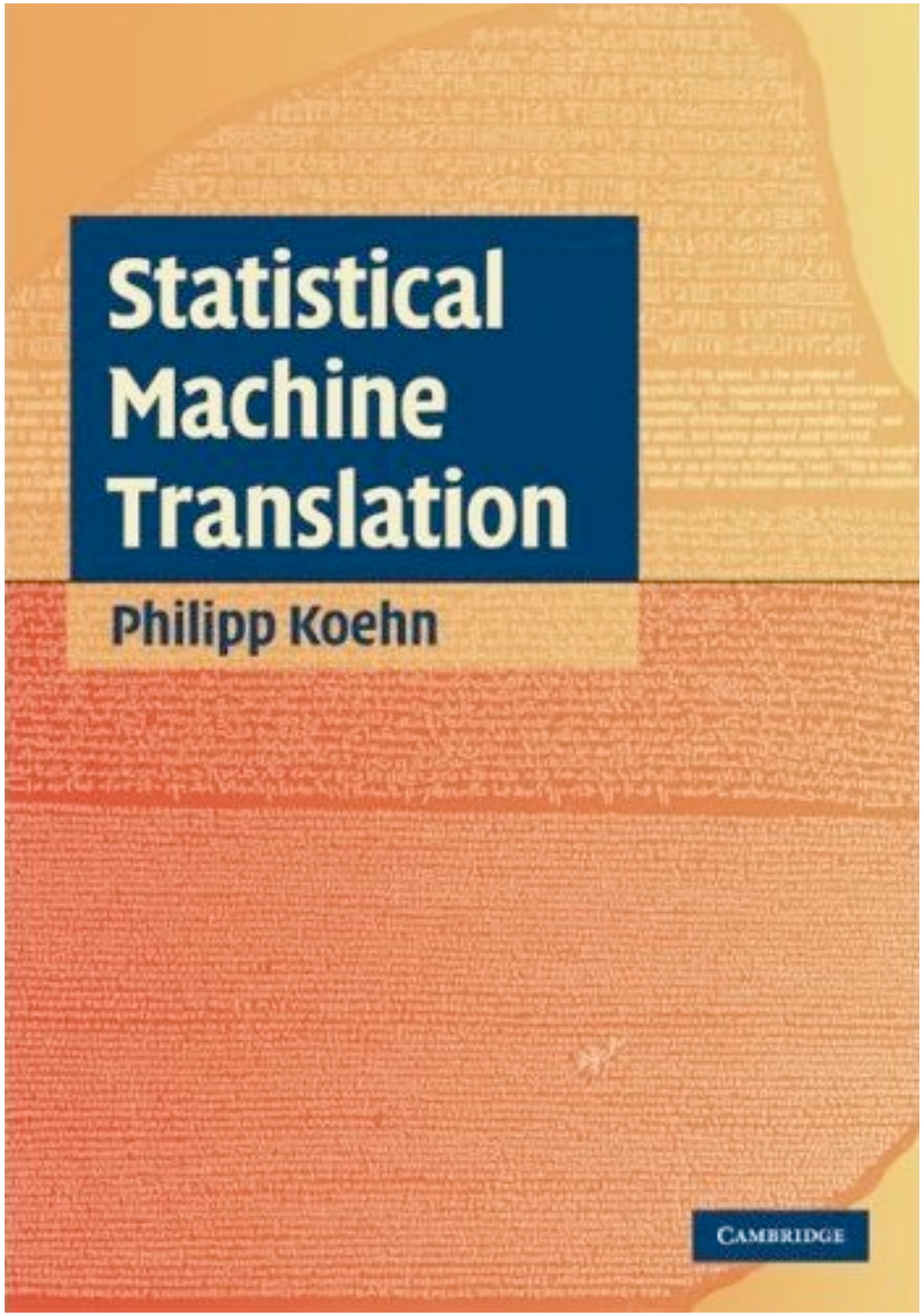


# Suggestions for HW1

- Matching the baseline will get you a B
- 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 leaderboard submissions are due by Tuesday at 11:59pm**
- **HW1 write ups and code are due 24 hours later**