Word Alignment

Wednesday, February 18, 2015

Plan for Today:

- Wrap up EM for alignment
- Survey of alignment extensions

Training Without Alignments

Initially assume all p(f|e) are equally probable

Repeat:

- Enumerate all possible alignments
- Calculate how probable the alignments are under the current model (i.e. p(f|e))
- Recalculate p(f|e) using counts from **all** alignments, **weighted** by how probable they are

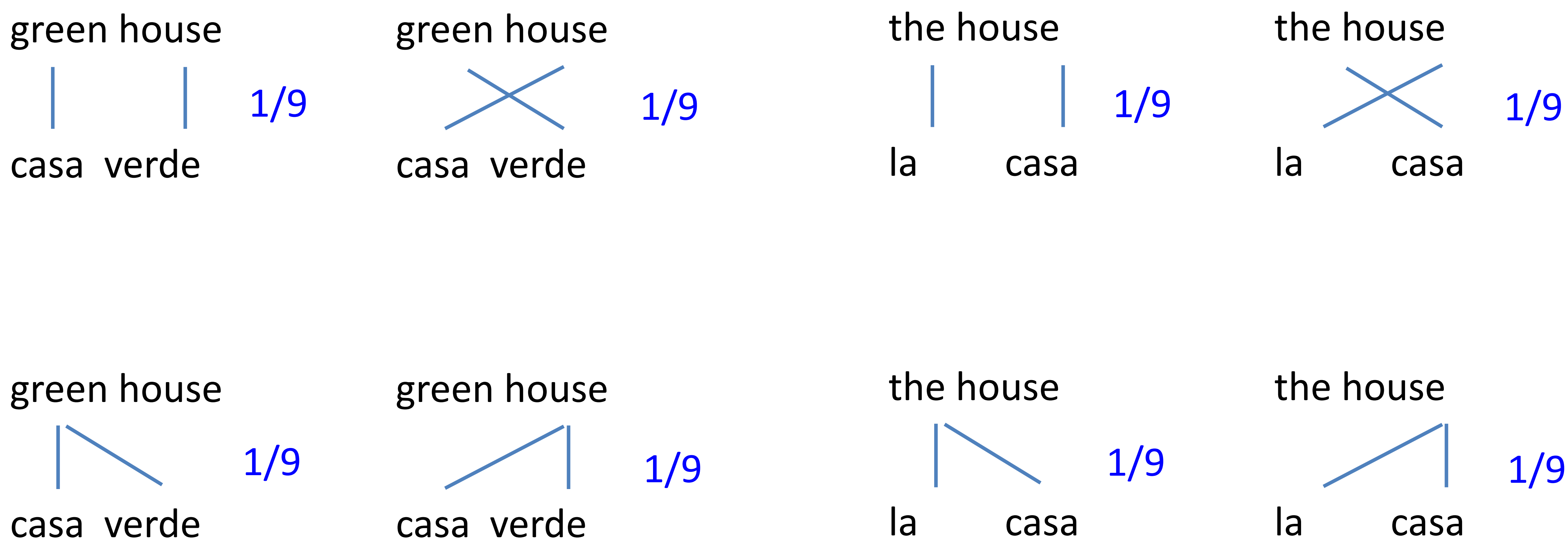
EM Alignment

E-step

- Enumerate all possible alignments
- Calculate how probable the alignments are under the current model (i.e. p(f|e))

M-step

- Recalculate p(f|e) using counts from **all** alignments, **weighted** by how probable they are



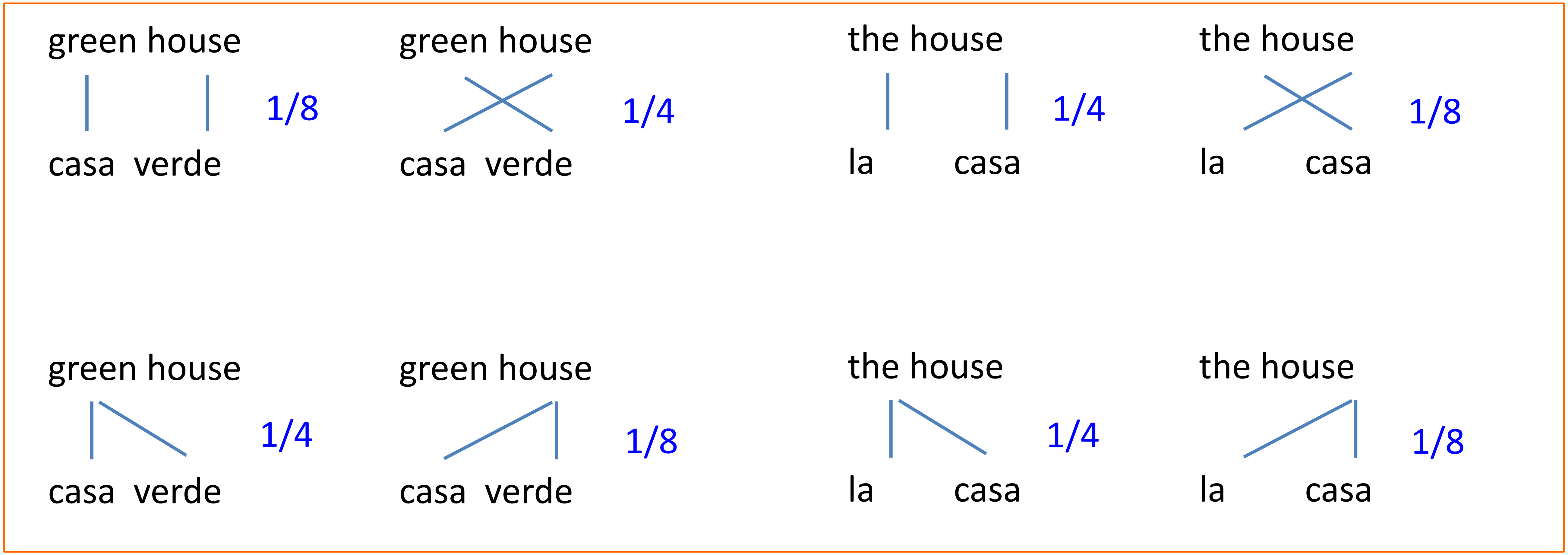
p(casa | green)	***1/3***
p(verde | green)	***1/3***
p(la | green)	***1/3***

p(casa | house)	***1/3***
p(verde | house)	***1/3***
p(la | house)	***1/3***

p(casa | the)	***1/3***
p(verde | the)	***1/3***
p(la | the)	***1/3***

E-step: What are the probabilities of the alignments?

$$p(f_1 f_2 \ldots f_{|F|}, a_1 a_2 \ldots a_{|F|} \mid e_1 e_2 \ldots e_{|E|}) = \prod_{i=1}^{|F|} p(f_i \mid e_{a_i})$$



p(casa / green)	***1/2***
p(verde / green)	***1/2***
p(la / green)	***0***

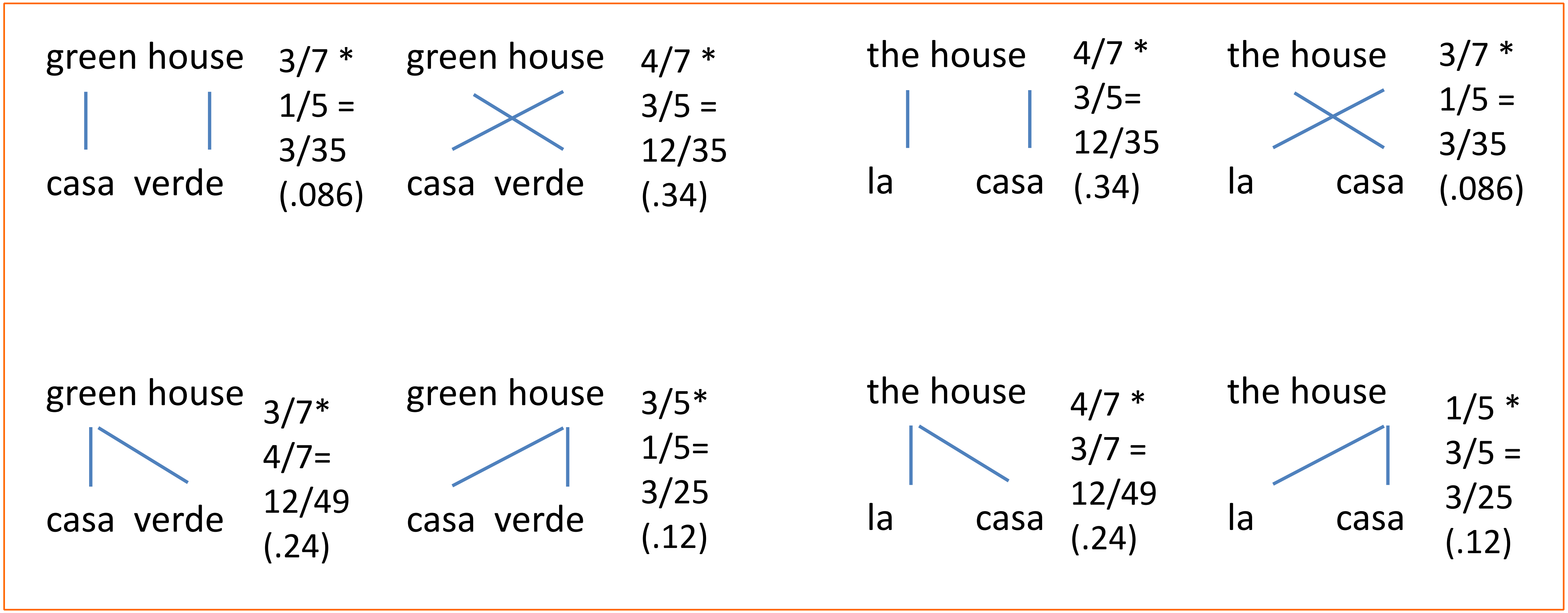
c(casa,green) = 1/9+1/9 = 1/3
c(verde,green) = 1/9+1/9 = 1/3
c(la, green) = 0

p(casa / house)	***1/2***
p(verde / house)	***1/4***
p(la / house)	***1/4***

c(casa,house) = 1/9+1/9+ 1/9+1/9 = 2/3
c(verde,house) = 1/9+1/9 = 1/3
c(la,house) = 1/9+1/9 = 1/3

p(casa / the)	***1/2***
p(verde / the)	***0***
p(la / the)	***1/2***

c(casa,the) = 1/9+1/9 = 1/3
c(verde,the) = 0
c(la,the) = 1/9+1/9 = 1/3



p(casa | green)	***3/7***
p(verde | green)	***4/7***
p(la | green)	***0***

c(casa,green) = 1/8+1/4 = 3/8
c(verde,green) = 1/4+1/4 = 1/2
c(la, green) = 0

p(casa | house)	***3/5***
p(verde | house)	***1/5***
p(la | house)	***1/5***

c(casa,house) = 1/4+1/8+ 1/4+1/8 = 3/4
c(verde,house) = 1/8+1/8 = 1/4
c(la,house) = 1/8+1/8 = 1/4

p(casa | the)	***3/7***
p(verde | the)	***0***
p(la | the)	***4/7***

c(casa,the) = 1/8+1/4 = 3/8
c(verde,the) = 0
c(la,the) = 1/4+1/4 = 1/2

green house
casa verde
3/7 * 1/5 = 3/35 (.086)

green house
casa verde
4/7 * 3/5 = 12/35 (.343)

the house
la casa
4/7 * 3/5= 12/35 (.343)

the house
la casa
3/7 * 1/5 = 3/35 (.086)

green house
casa verde
3/7* 4/7= 12/49 (.245)

green house
casa verde
3/5* 1/5= 3/25 (.12)

the house
la casa
4/7 * 3/7 = 12/49 (.245)

the house
la casa
1/5 * 3/5 = 3/25 (.12)

p(casa | green)	***3/7***
p(verde | green)	***4/7***
p(la | green)	***0***

p(casa | house)	***3/5***
p(verde | house)	***1/5***
p(la | house)	***1/5***

p(casa | the)	***3/7***
p(verde | the)	***0***
p(la | the)	***4/7***

c(casa,green) = .086+.245=0.331
c(verde,green) = .343+0.245 = 0.588
c(la, green) = 0

c(casa,house) = .343+.12+ .343+.12=0.926
c(verde,house) = .086+.12=0.206
c(la,house) = .086+.12=0.206

c(casa,the) = .086+.245=0.331
c(verde,the) = 0
c(la,the) = .343+.245=0.588

Iterate…

5 iterations

p(casa | green)	*0.24*
p(verde | green)	*0.76*
p(la | green)	*0*

p(casa | house)	*0.84*
p(verde | house)	*0.08*
p(la | house)	*0.08*

p(casa | the)	*0.24*
p(verde | the)	*0*
p(la | the)	*0.76*

10 iterations

p(casa | green)	*0.1*
p(verde | green)	*0.9*
p(la | green)	*0*

p(casa | house)	*0.98*
p(verde | house)	*0.01*
p(la | house)	*0.01*

p(casa | the)	*0.1*
p(verde | the)	*0*
p(la | the)	*0.9*

100 iterations

p(casa | green)	*0.005*
p(verde | green)	*0.995*
p(la | green)	*0*

p(casa | house)	*~1.0*
p(verde | house)	*~0.0*
p(la | house)	*~0.0*

p(casa | the)	*0.005*
p(verde | the)	*0*
p(la | the)	*0.995*

EM Alignment

E-step
- En
- Ca (i.e. p(f|e))

M-step
- Re v probable they
  ar

Why does it work?

EM Alignment

Intuitively:

M-step

- Recalculate p(f|e) using counts from **all** alignments, **weighted** by how probable they are

Things that co-occur will have higher probabilities

E-step

- Calculate how probable the alignments are under the current model (i.e. p(f|e))

Alignments that contain things with higher p(f|e) will be scored higher

An Aside: Estimating Probabilities

What is the probability of "the" occurring in a sentence?

$$\frac{\text{number of sentences with “the”}}{\text{total number of sentences}}$$

Is this right?

Estimating Probabilities

- What is the probability of "the" occurring in a sentence? Maximum Likelihood Estimation (MLE)

$$\frac{\text{number of sentences with “the”}}{\text{total number of sentences}}$$

No. This is an *estimate* based on our data

This is the maximum likelihood estimation.

EM Alignment: The Math

The EM algorithm tries to find parameters to the model (in our case, p(f|e)) that maximize the likelihood of the data

In our case:

Each iteration, we increase (or keep the same) the likelihood of the data

$$p(f_1 f_2 \ldots f_{|F|} \mid e_1 e_2 \ldots e_{|E|}) = \sum_{a_1} \sum_{a_2} \ldots \sum_{a_{|F|}} p(f_i \mid e_{a_i})$$

Implementation Details

Any concerns/issues?
Anything underspecified?

Repeat:

E-step

- Enumerate all possible alignments
- Calculate how probable the alignments are under the current model (i.e. $p(f|e)$)

M-step

- Recalculate $p(f|e)$ using counts from **all** alignments, **weighted** by how probable they are

Implementation Details

When do we stop?

Repeat:

E-step

- Enumerate all possible alignments
- Calculate how probable the alignments are under the current model (i.e. p(f|e))

M-step

- Recalculate p(f|e) using counts from **all** alignments, **weighted** by how probable they are

Implementation Details

- Repeat for a fixed number of iterations
- Repeat until parameters don't change (much)
- Repeat until likelihood of (some) data doesn't change (much)

Repeat:

E-step

- Enumerate all possible alignments
- Calculate how probable the alignments are under the current model (i.e. p(f|e))

M-step

- Recalculate p(f|e) using counts from **all** alignments, **weighted** by how probable they are

Implementation Details

For |E| English words and |F| foreign words, how many alignments are there?

Repeat:

E-step

- Enumerate all possible alignments
- Calculate how probable the alignments are under the current model (i.e. p(f|e))

M-step

- Recalculate p(f|e) using counts from **all** alignments, **weighted** by how probable they are

Implementation Details

Each foreign word can be aligned to any of the English words (or NULL)

$$(|E|+1)^{|F|}$$

Repeat:

E-step

- Enumerate all possible alignments
- Calculate how probable the alignments are under the current model (i.e. p(f|e))

M-step

- Recalculate p(f|e) using counts from **all** alignments, **weighted** by how probable they are

Thought Experiment

The old man is happy. He has fished many times.

El viejo está feliz porque ha pescado muchos veces.

His wife talks to him.

Su mujer habla con él.

The sharks await.

Los tiburones esperan.

$$p(f_i \mid e_{a_i}) = \frac{count(f \ \ aligned\text{-}to \ e)}{count(e)}$$

p(el | the) = 0.5

p(Los | the) = 0.5

If we had Alignments…

Input: corpus of English/Foreign sentence pairs along with alignment

```
for (E, F) in corpus:
    for aligned words (e, f) in pair (E,F):
        count(e,f) += 1
        count(e) += 1

for all (e,f) in count:
    p(f|e) = count(e,f) / count(e)
```

Without the Alignments

Input: corpus of English/Foreign sentence pairs along with alignment

```
for (E, F) in corpus:
    for e in E:
        for f in F:
            p(f -> e): probability that f is aligned to e in this pair
            count(e,f) += p( f -> e)
            count(e) += p(f -> e)

for all (e,f) in count:
    p(f|e) = count(e,f) / count(e)
```

Without the Alignments

p(f -> e): probability that f is aligned to e *in this pair*

a b c

y z

What is p(y -> a)?

Put another way, of all things that y could align to, how likely is it to be a?

Without the Alignments

p(f -> e): probability that f is aligned to e *in this pair*

a b c

y z

Of all things that y could align to, how likely is it to be a:

$$p(y \mid a)$$

Does that do it?

No! p(y | a) is how likely y is to align to a over the whole data set.

Without the Alignments

p(f -> e): probability that f is aligned to e *in this pair*

a b c

y z

Of all things that y could align to, how likely is it to be a:

$$\frac{p(y \mid a)}{p(y \mid a) + p(y \mid b) + p(y \mid c)}$$

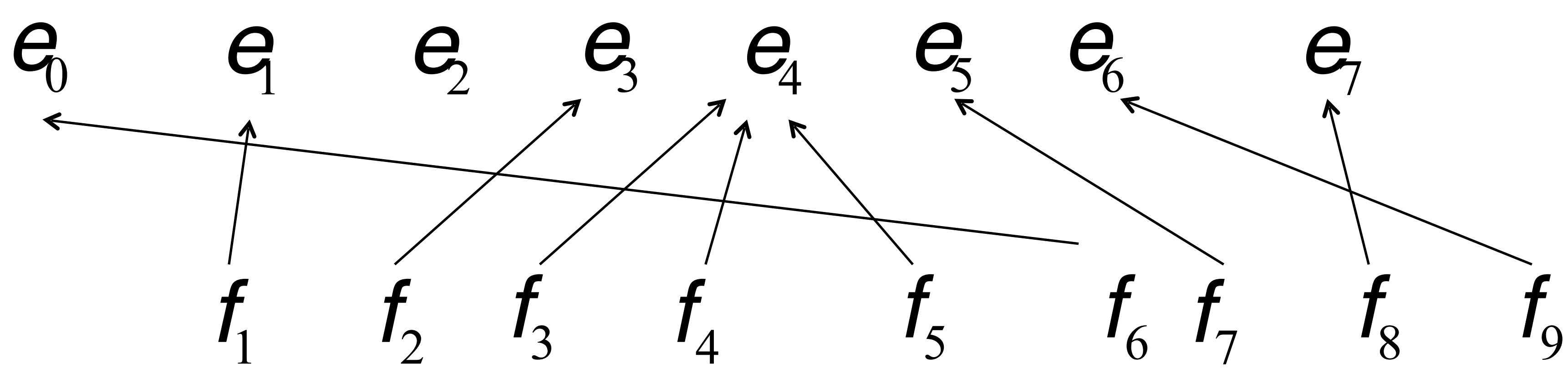
Without the Alignments

```
Input: corpus of English/Foreign sentence pairs along with alignment

for (E, F) in corpus:
    for e in E:
        for f in F:
            p(f -> e) = p(f | e) / ( sum_(e in E) p( f | e) )
            count(e,f) += p( f -> e)
            count(e) += p(f -> e)

for all (e,f) in count:
    p(f|e) = count(e,f) / count(e)
```

Good/Bad of Word-Level Models

Rarely used in practice for modern MT system

Mary did not slap the green witch



Maria no dió una botefada a la bruja verde

Two key side effects of training a word-level model:

- Word-level alignment
- p(f | e): translation dictionary

How do I get this?

Word alignment

100 iterations

p(casa | green)	***0.005***
p(verde | green)	***0.995***
p(la | green)	***0***

p(casa | house)	***~1.0***
p(verde | house)	***~0.0***
p(la | house)	***~0.0***

p(casa | the)	***0.005***
p(verde | the)	***0***
p(la | the)	***0.995***

green house

casa verde

How should these be aligned?

the house

la casa

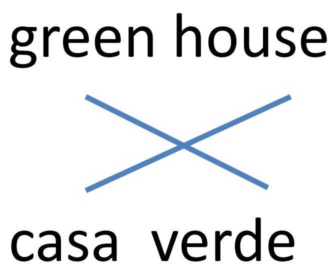
Word Alignment

100 iterations

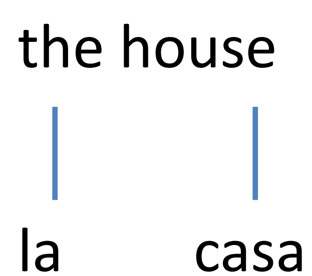
p(casa | green)	***0.005***
p(verde | green)	***0.995***
p(la | green)	***0***

p(casa | house)	***~1.0***
p(verde | house)	***~0.0***
p(la | house)	***~0.0***

p(casa | the)	***0.005***
p(verde | the)	***0***
p(la | the)	***0.995***



Why?



Word Alignment

$$alignment(E,F) = \arg_A \max p(A,F \mid E)$$

Which for IBM model 1 is:

$$alignment(E,F) = \arg_A \max \prod_{i=1}^{|F|} p(f_i \mid e_{a_i})$$

Given a model (i.e. trained p(f|e)), how do we find this?

Align each foreign word (f in F) to the English word (e in E) with highest p(f|e)

$$a_i = \arg_{j:1-|E|} \max p(f_i \mid e_j)$$

Word Alignment Evaluation

The old man is happy. He has fished many times.

El viejo está feliz porque ha pescado muchos veces.

How good of an alignment is this?
How can we quantify this?

Word Alignment Evaluation

Hypothesis (generated by the system):

The old man is happy. He has fished many times.

El viejo está feliz porque ha pescado muchos veces.

Reference (generated by a human):

The old man is happy. He has fished many times.

El viejo está feliz porque ha pescado muchos veces.

How can we quantify this?

Characterizing Human Alignments

S(ure) alignments

(casa -> house, la -> the)

P(ossible) alignments

(viejo -> old, viejo -> man)

In evaluation, we want to:

- Not penalize our system if it finds a "possible" alignment
- Penalize our system if it doesn't find a "sure" alignment

Quantifying Alignment Success

Precision: $|A \cap P| \,/\, |A|$

Recall: $|A \cap S| \,/\, |S|$

Alignment Error Rate:

$$AER = 1 - (|A \cap S| + |A \cap P|) \,/\, (|A| + |S|)$$

(For comtrans data, Possible=Sure)

Quantifying Alignment Success

Hypothesis (generated by the system):

The old man is happy. He has fished many times.

El viejo está feliz porque ha pescado muchos veces.

Reference (generated by a human):

The old man is happy. He has fished many times.

El viejo está feliz porque ha pescado muchos veces.

Precision: $|A \cap P| / |A|$

Recall: $|A \cap S| / |S|$

Alignment Error Rate: $AER = 1 - (|A \cap S| + |A \cap P|) / (|A| + |S|)$

Which Alignment is Better?

Reference (generated by a human):

The old man is happy. He has fished many times.

El viejo está feliz porque ha pescado muchos veces.

Hypothesis 1 (generated by System 1):

The old man is happy. He has fished many times.

El viejo está feliz porque ha pescado muchos veces.

Hypothesis 2 (generated by System 2):

The old man is happy. He has fished many times.

El viejo está feliz porque ha pescado muchos veces.

Getting Better Alignments…

IBM Model 2: Some alignments are more likely than others.

- Especially for similar languages, words near the beginning will align to words near the beginning
- Completely jumbled alignments are unlikely (though not impossible)
- In math:

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

m=length of French sentence

n=length of English sentence

i=index of English word

a_i=index of French word

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)

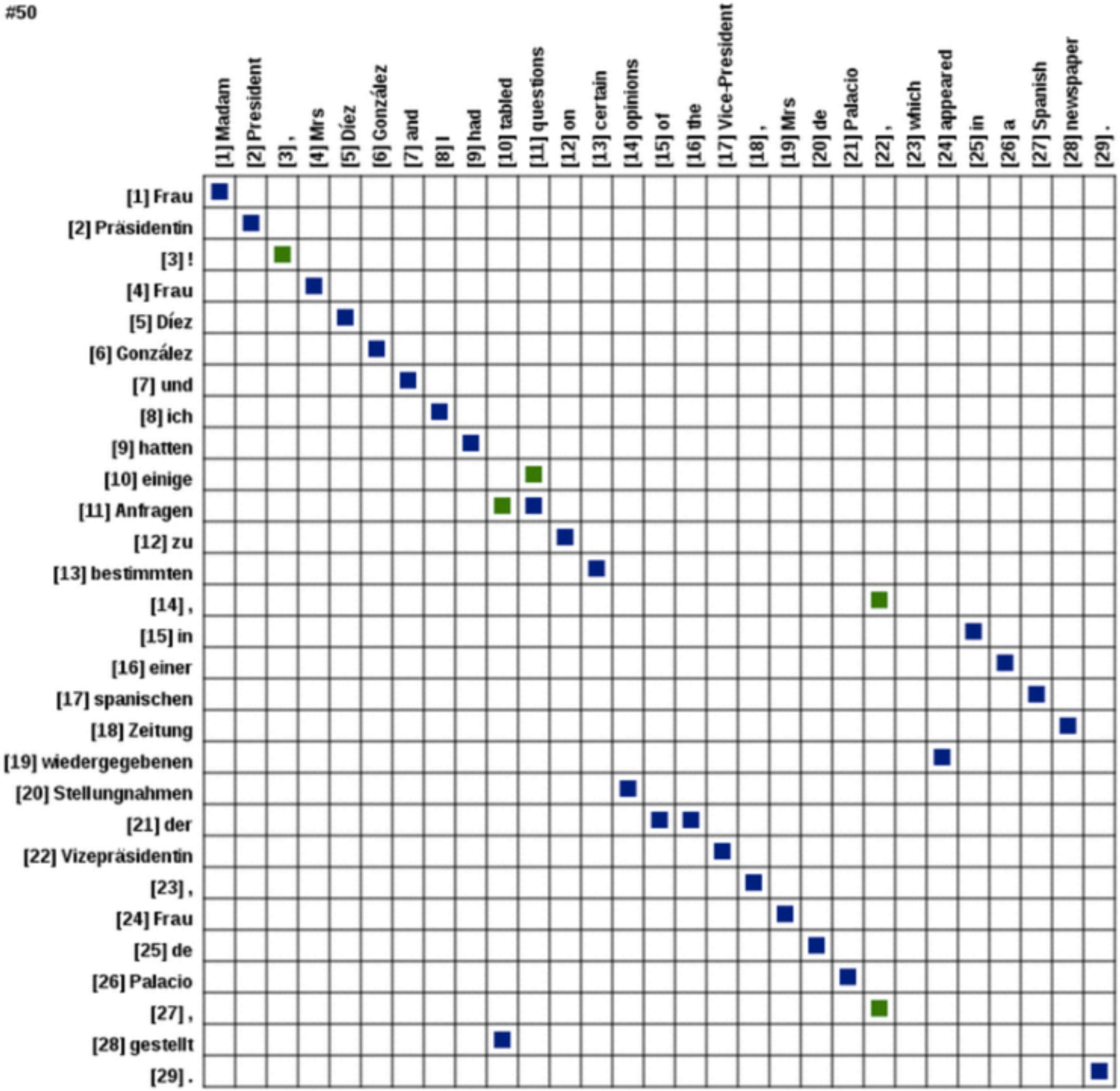
$$p(a_i \mid i, m, n)$$

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

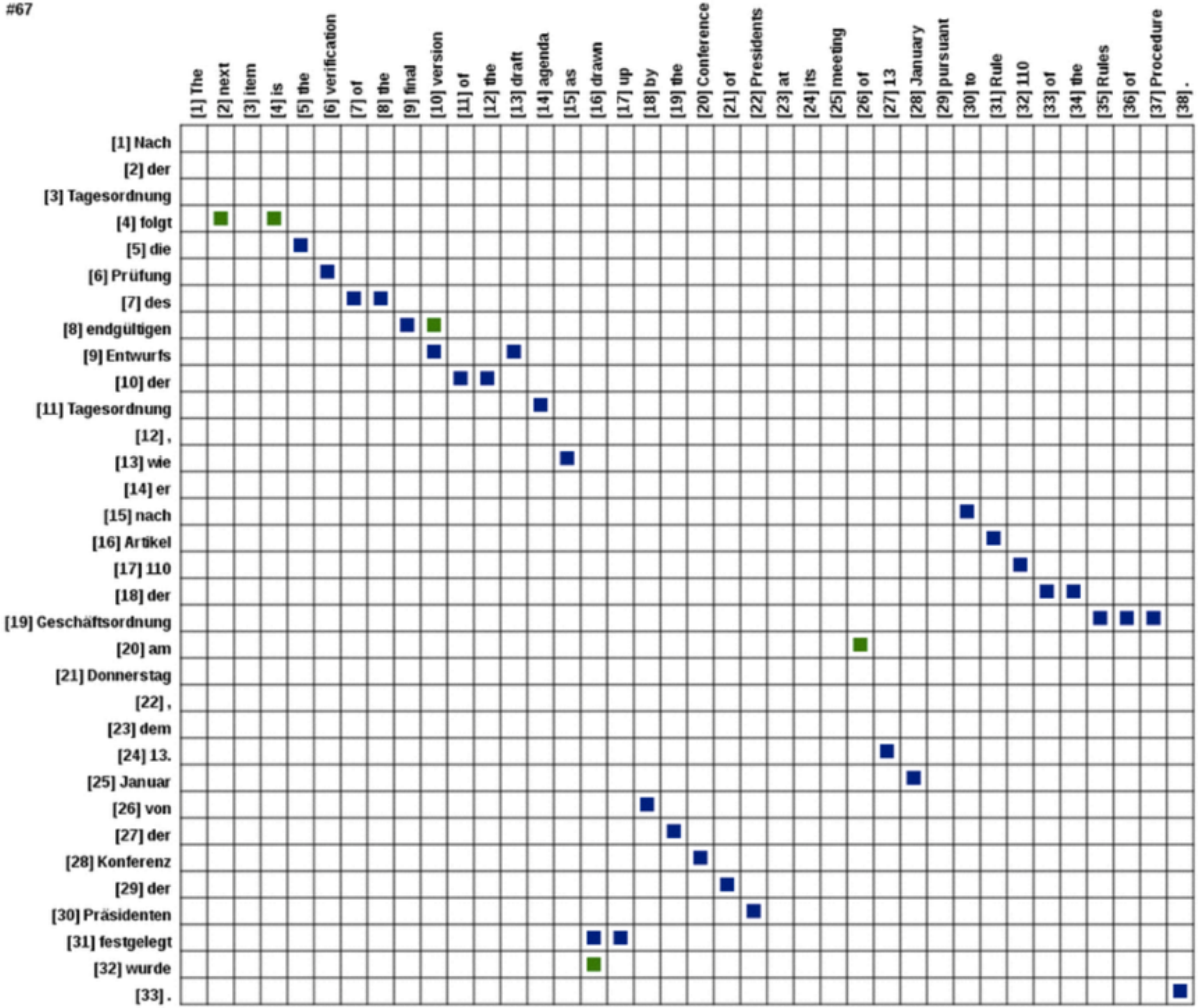
Model 2 $= \sum_{\mathbf{a} \in [0,n]^m} \prod_{i=1}^{m} p(a_i \mid i, m, n) \times p(e_i \mid f_{a_i})$

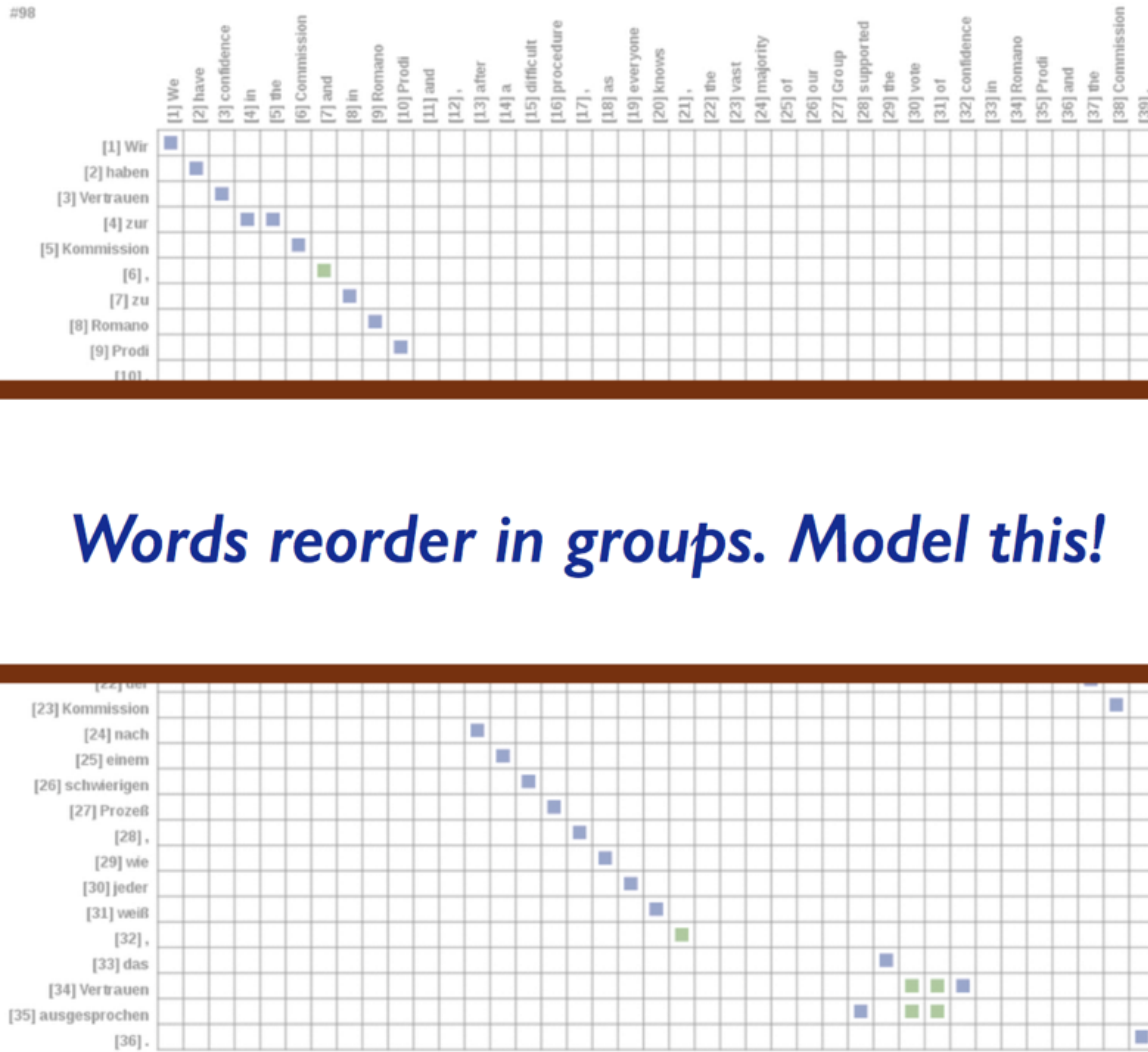
- Pros
  - Non-uniform alignment model
  - Fast EM training / marginal inference
- Cons
  - Absolute position is *very naive*
  - How many parameters to model $p(a_i \mid i, m, n)$

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

We'll hear more about this method next week!