Algorithms for NLP

Machine Translation II

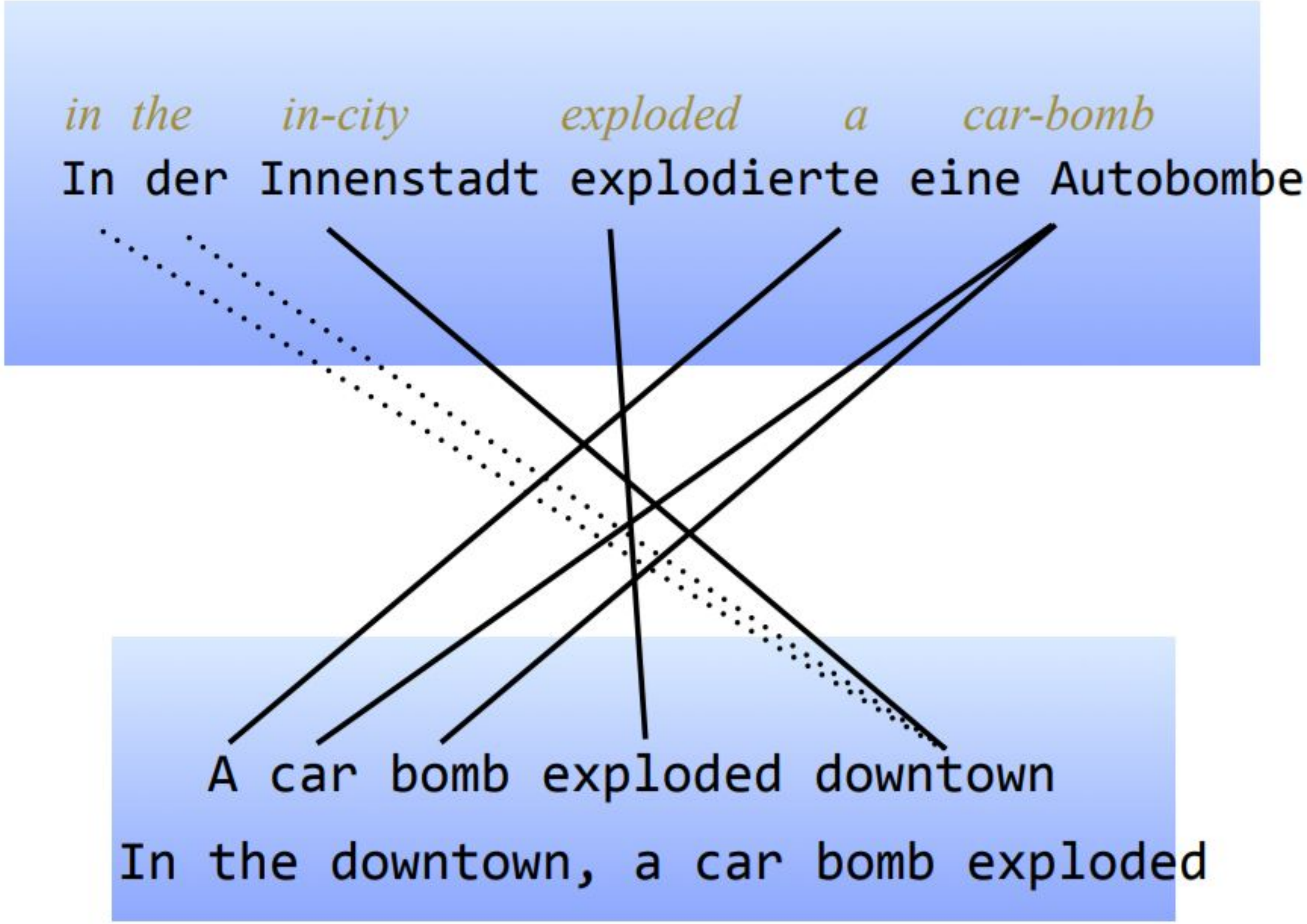
Yulia Tsvetkov – CMU

Slides: Philipp Koehn – JHU; Chris Dyer – DeepMind

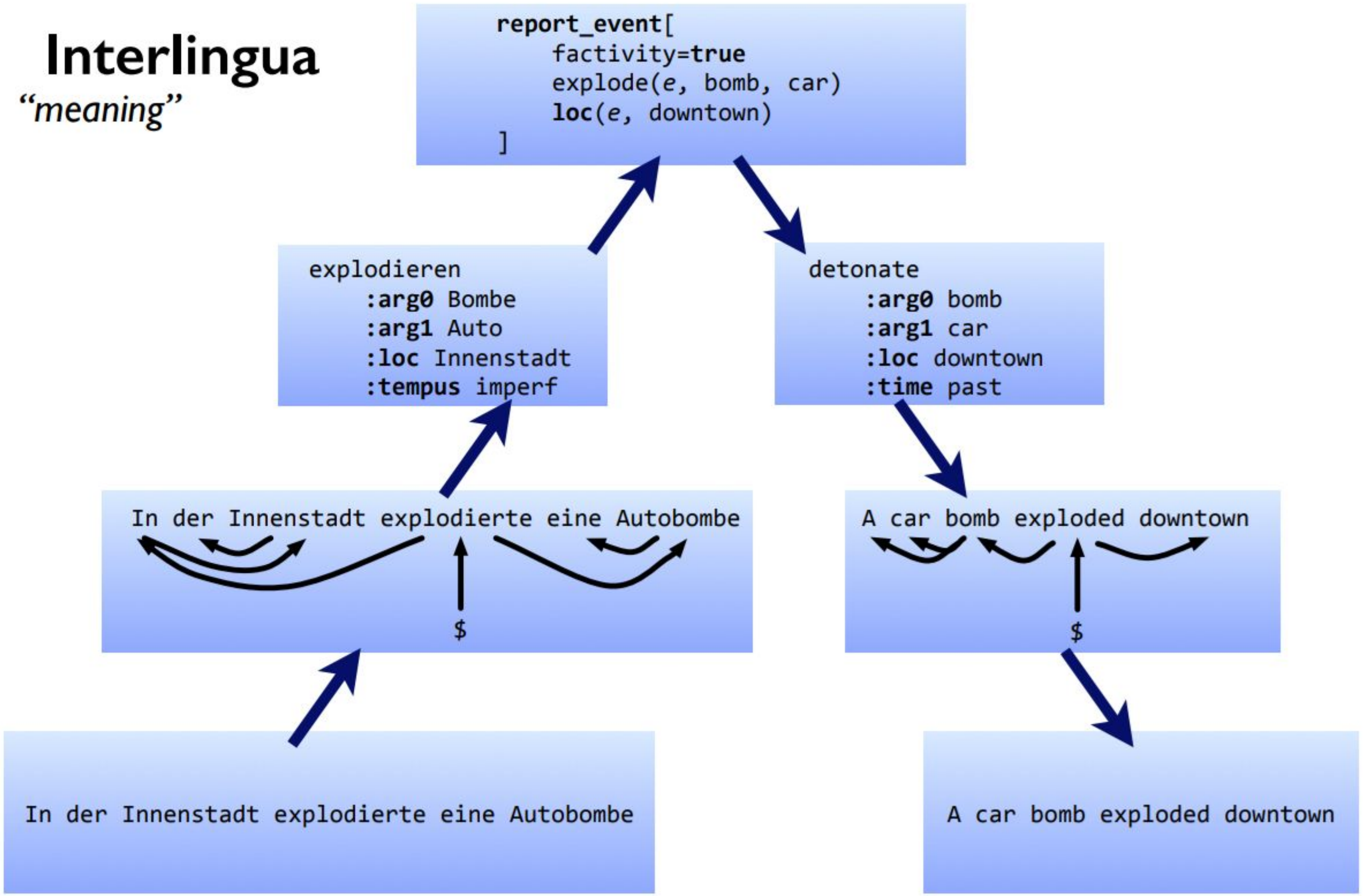
MT is Hard

Ambiguities

- words
- morphology
- syntax
- semantics
- pragmatics



Levels of Transfer

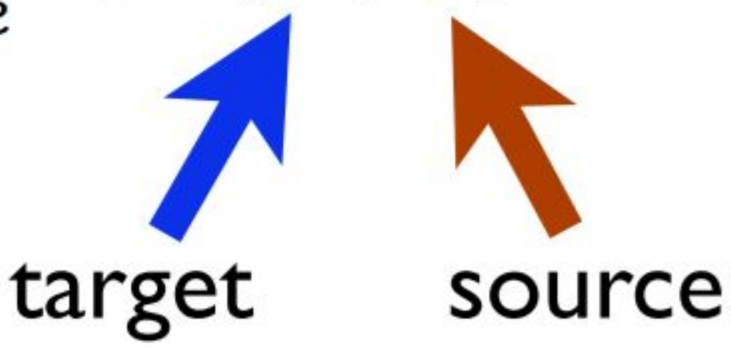


Two Views of Statistical MT

- Direct modeling (aka pattern matching)
  - I have really good learning algorithms and a bunch of example inputs (source language sentences) and outputs (target language translations)
- Code breaking (aka the noisy channel, Bayes rule)
  - I know the target language
  - I have example translations texts (example enciphered data)

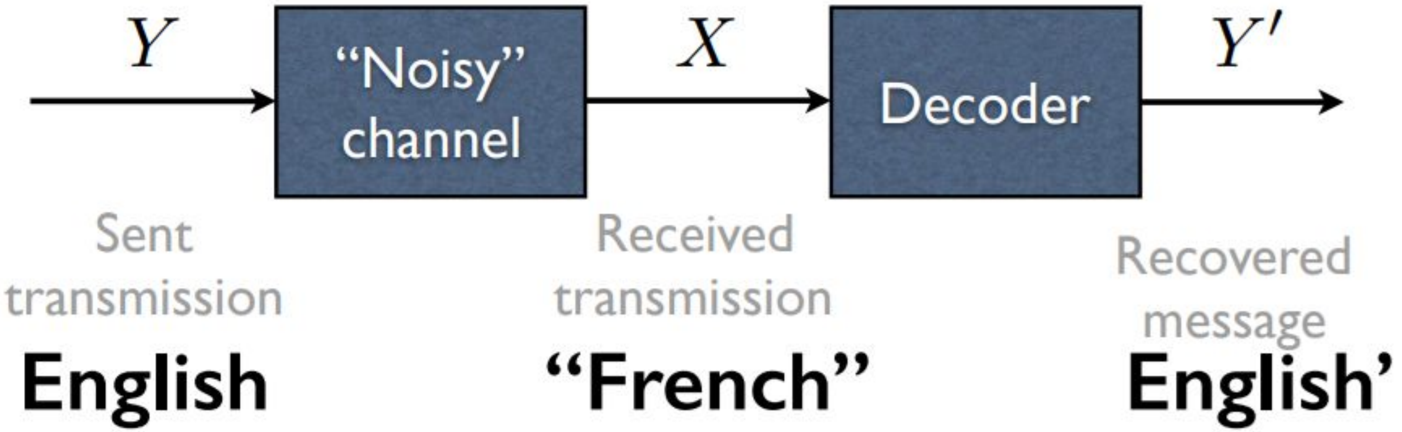
MT as Direct Modeling

$$\hat{e} = \arg\max_{\boldsymbol{e}} p_{\boldsymbol{\lambda}}(\boldsymbol{e} \mid \boldsymbol{f})$$



- one model does everything
- trained to reproduce a corpus of translations

Noisy Channel Model



$$\hat{e} = \arg\max_{\boldsymbol{e}} p_{\boldsymbol{\varphi}}(\boldsymbol{e}) \times p_{\boldsymbol{\theta}}(\boldsymbol{f} \mid \boldsymbol{e})$$

language model

translation model

Which is better?

- Noisy channel - $p_{\boldsymbol{\theta}}(\boldsymbol{e}) \times p_{\boldsymbol{\varphi}}(\boldsymbol{f} \mid \boldsymbol{e})$
  - easy to use monolingual target language data
  - search happens under a product of two models (individual models can be simple, product can be powerful)
  - obtaining probabilities requires renormalizing
- Direct model - $p_{\boldsymbol{\lambda}}(\boldsymbol{e} \mid \boldsymbol{f})$
  - directly model the process you care about
  - model must be very powerful

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 .

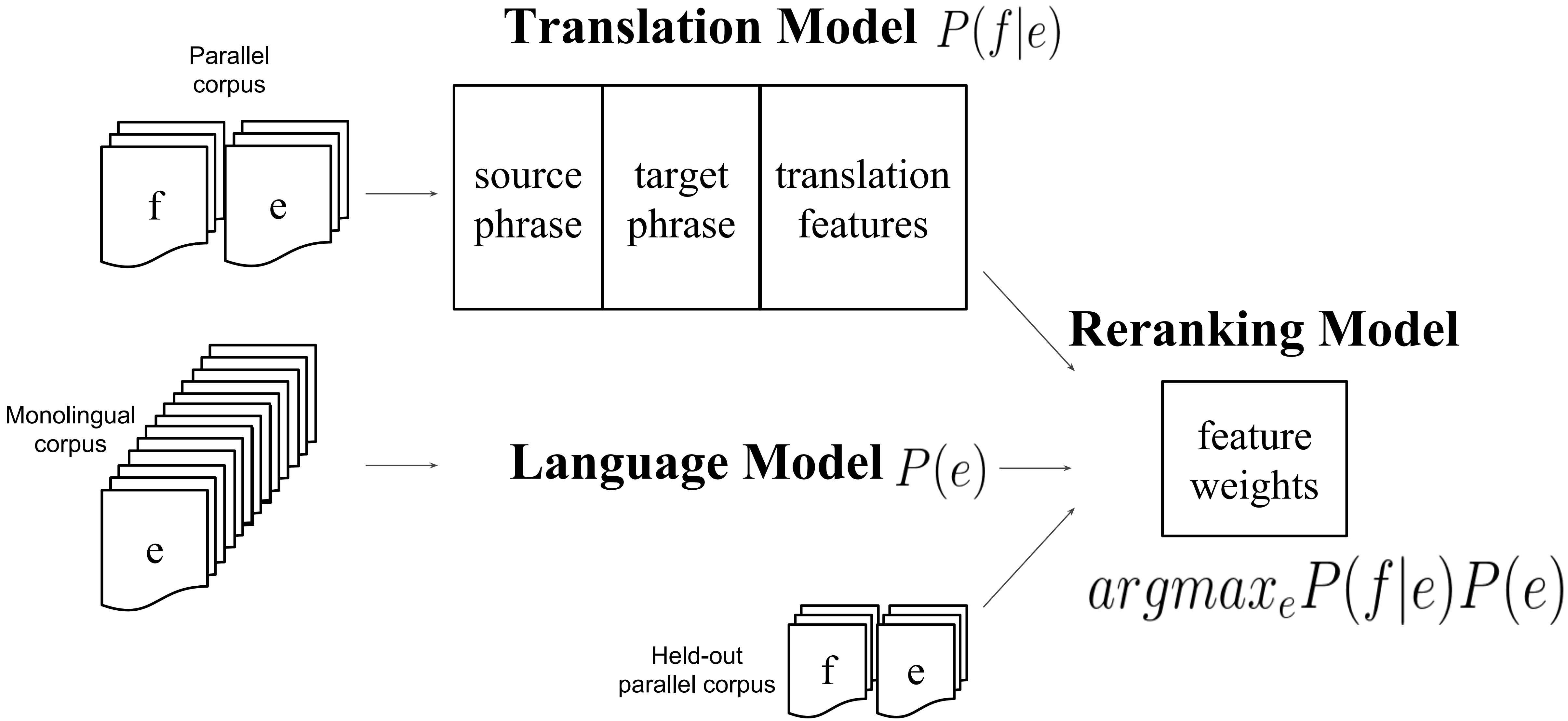
11a. lalok nok crrrok hihok yorok zanzanok .
11b. wat nnat arrat mat zanzanat .

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

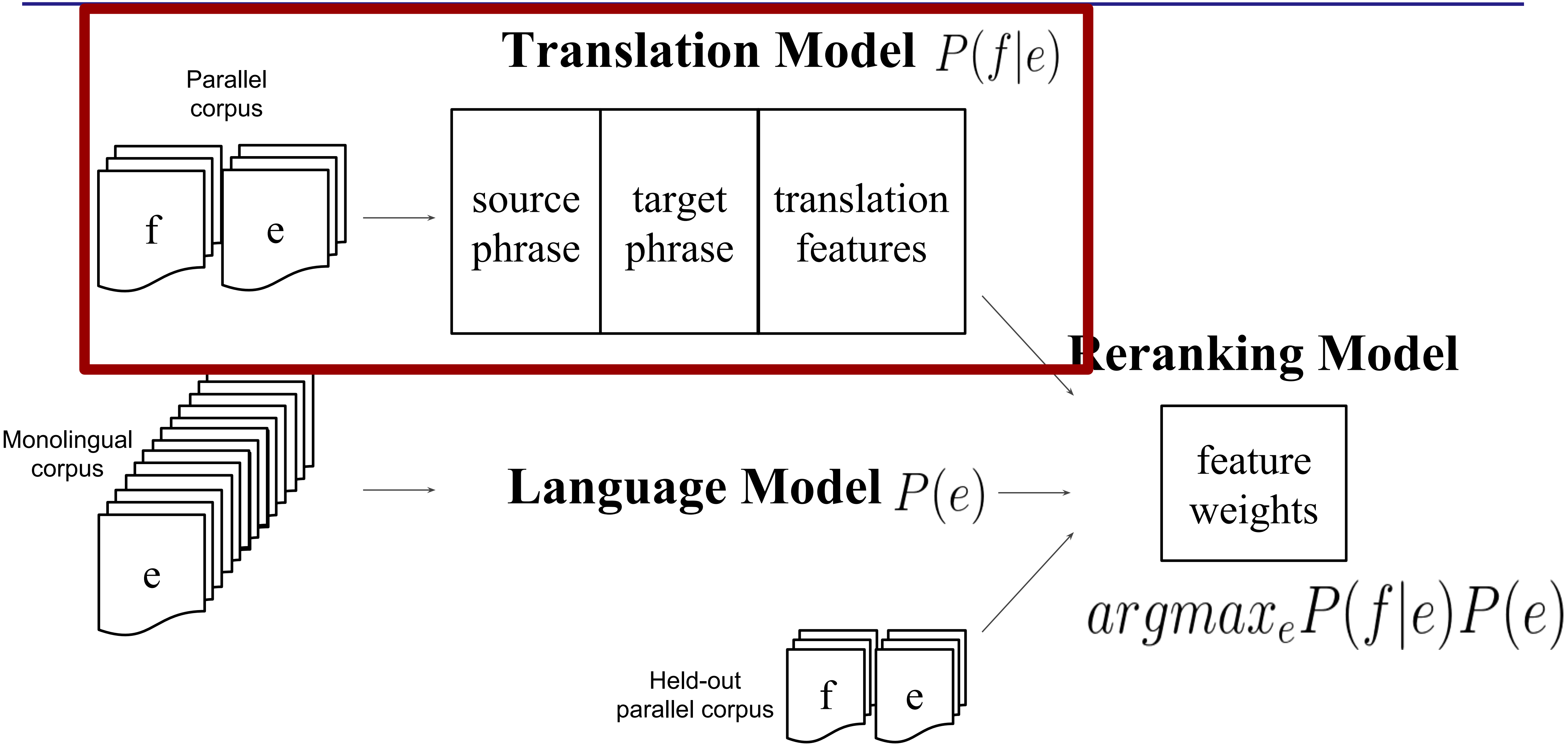
Translation challenge: **farok crrrok hihok yorok clok kantok ok-yurp**

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

Noisy Channel Model : Phrase-Based MT



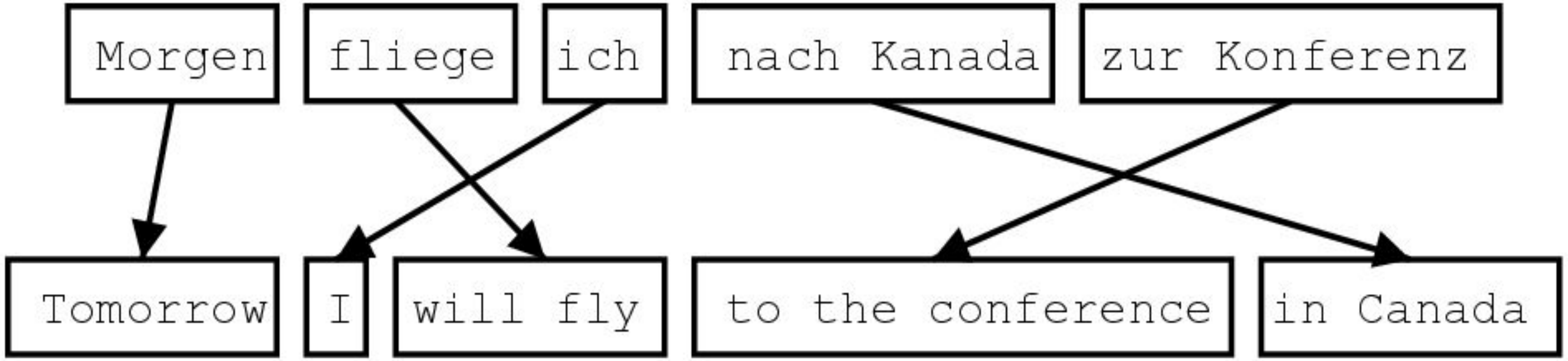
Phrase-Based MT

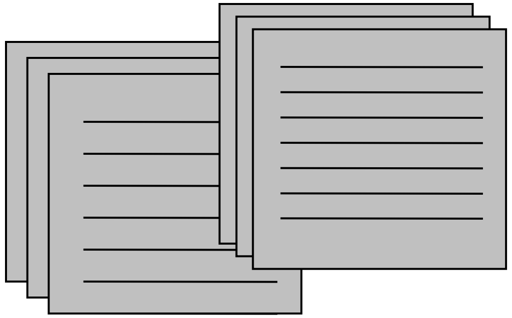


Phrase-Based Translation

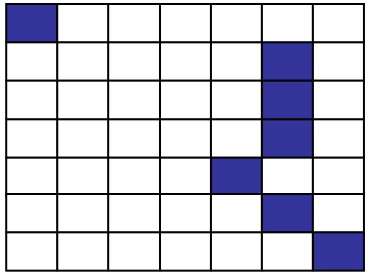
в	этом	смысле	подобные	действия	частично	дискредитируют	систему	американской	демократии
in	this	sense	such	actions	some	discredit	system	american	democracy
the	that	meaning	similar	action	partially		a system	u.s.	democracies
a	the	terms	these	the	part		systems	us	democratic
at	it	way	this	acts	in part		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									

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)

Lexical Translation

- How do we translate a word? Look it up in the dictionary

  Haus — house, building, home, household, shell

- Multiple translations
  - some more frequent than others
  - 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?

Look at a parallel corpus (German text along with English translation)

Translation of *Haus*	Count
house	8,000
building	1,600
home	200
household	150
shell	50

Estimate Translation Probabilities

Maximum likelihood estimation

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

Lexical Translation

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

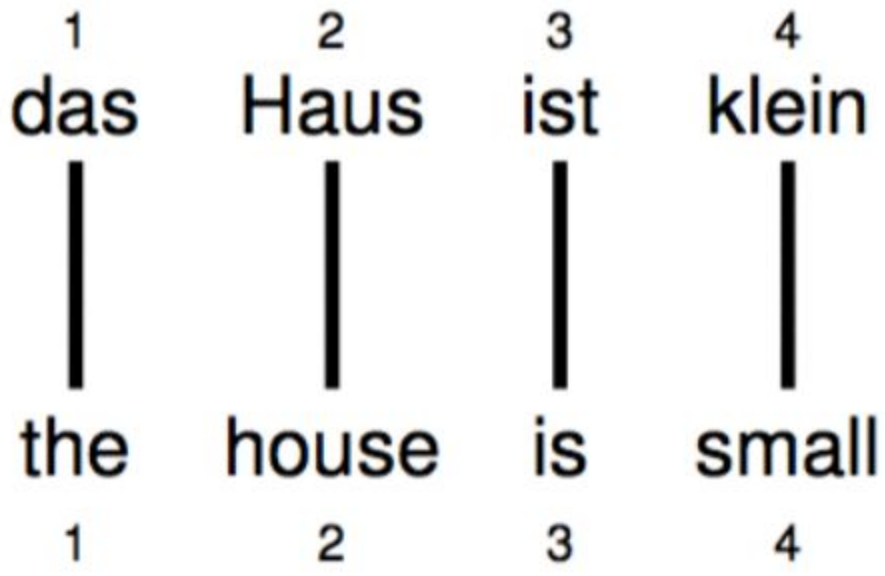
$$\mathbf{e} = \langle e_1, e_2, \ldots, e_m \rangle \qquad \mathbf{f} = \langle f_1, f_2, \ldots, f_n \rangle$$

Alignment Function

- In a parallel text (or when we translate), we align words in one language with the words in the other
- Alignments are represented as vectors of positions:

$$\mathbf{f} = \langle f_1, f_2, \ldots, f_n \rangle$$

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



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

Alignment Function

- Formalizing alignment with an alignment function

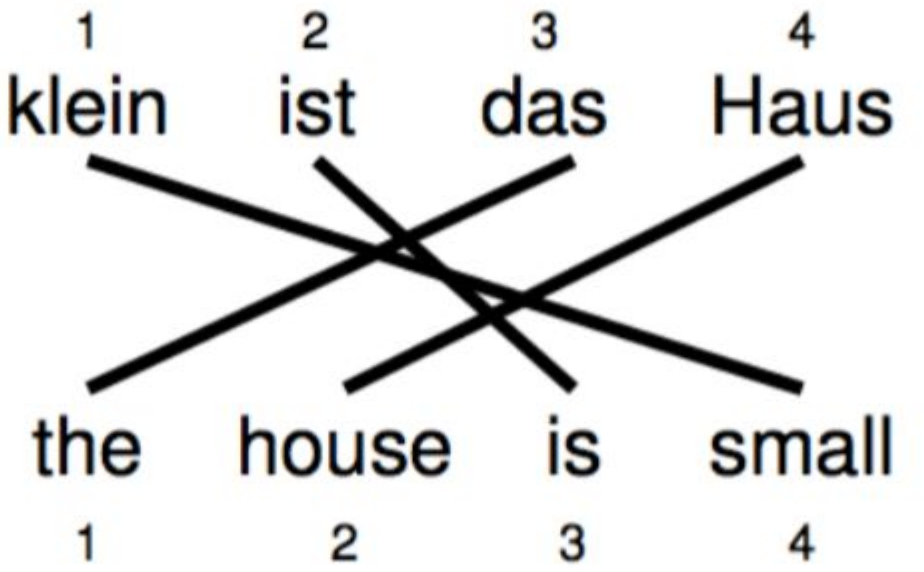
- Mapping an English target word at position *i* to a German source word at position *j* with a function $a : i \to j$

- Example

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

Reordering

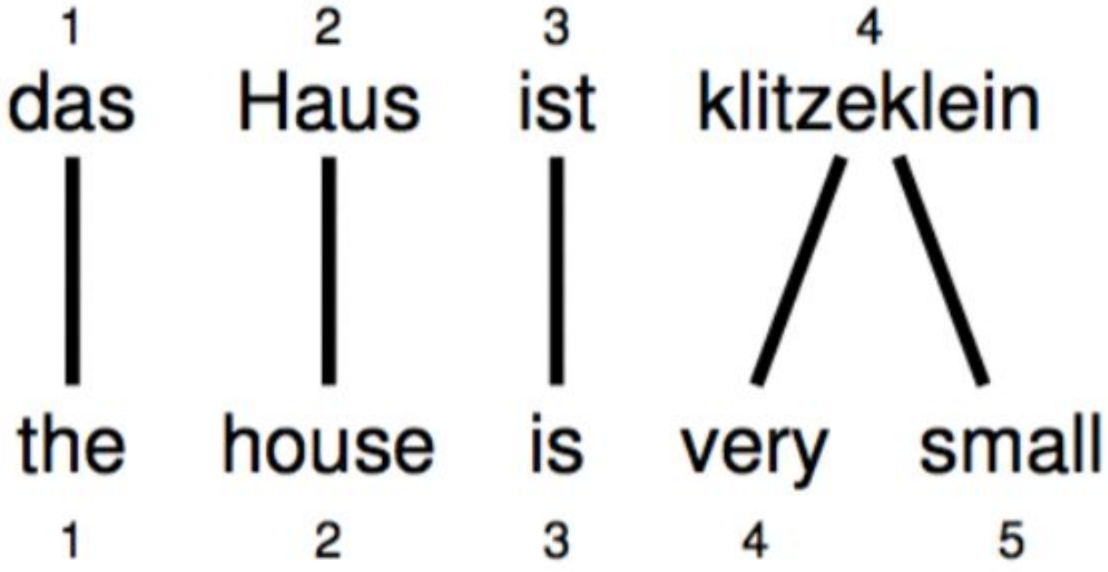
- Words may be reordered during translation.



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

One-to-many Translation

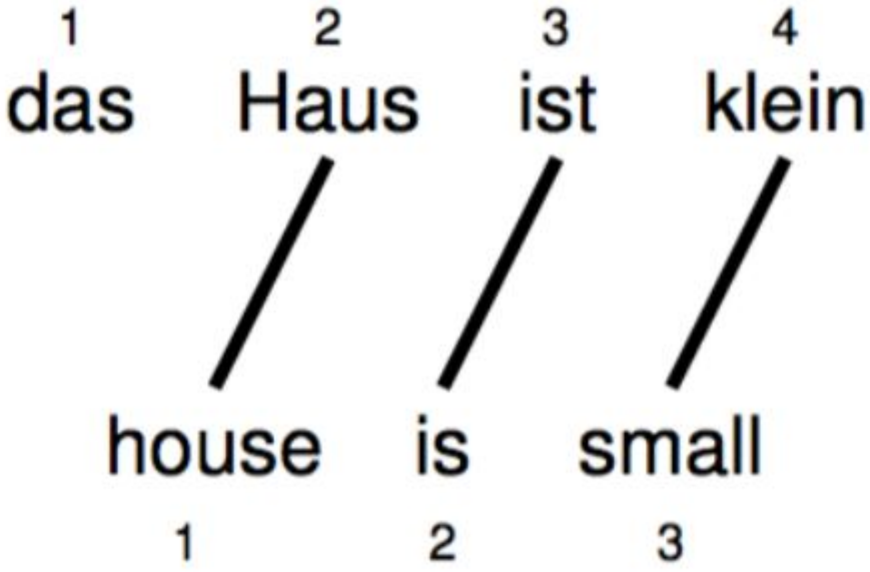
- A source word may translate into more than one target word
-



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

Word Dropping

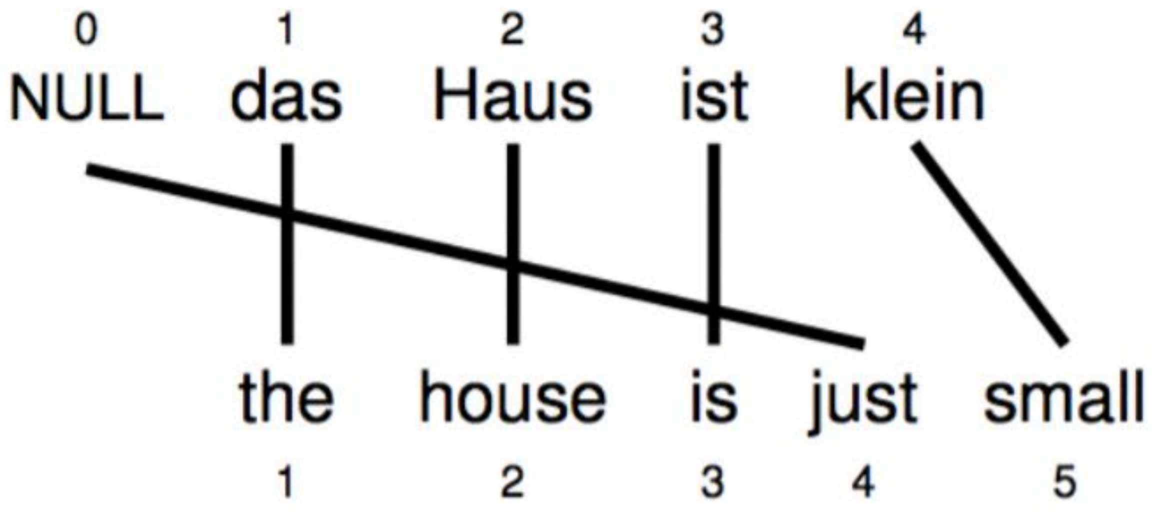
- A source word may not be translated at all



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

Word Insertion

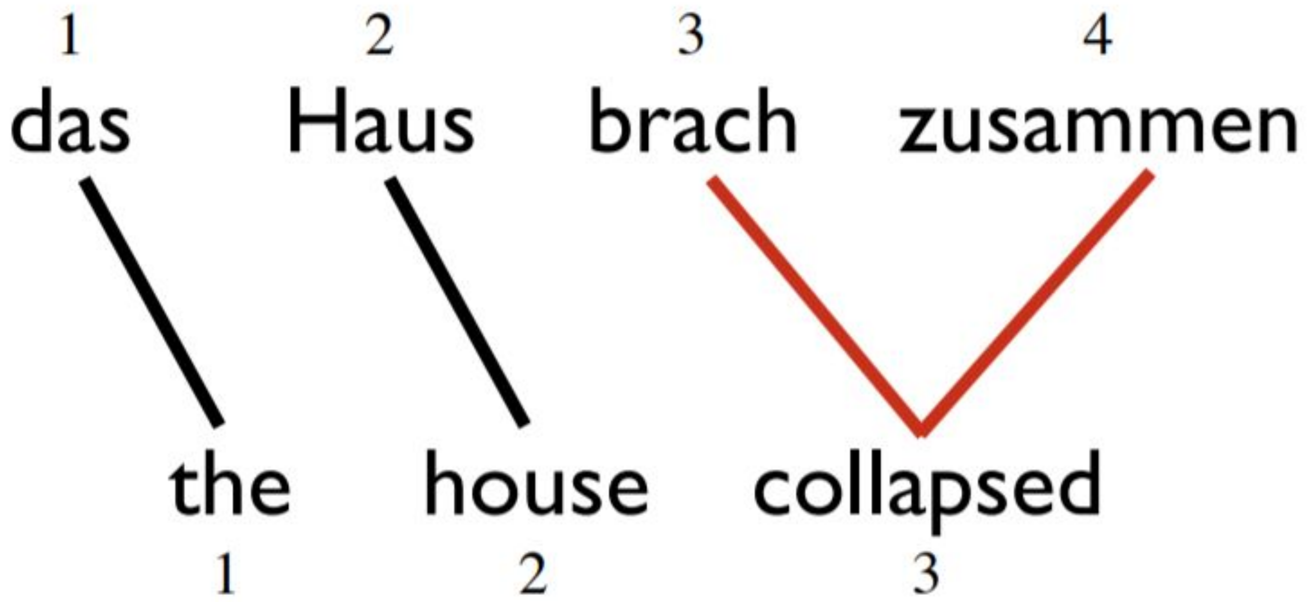
- Words may be inserted during translation
  - English *just* does not have an equivalent
  - But it must be explained - we typically assume every source sentence contains a NULL token



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

Many-to-one Translation

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



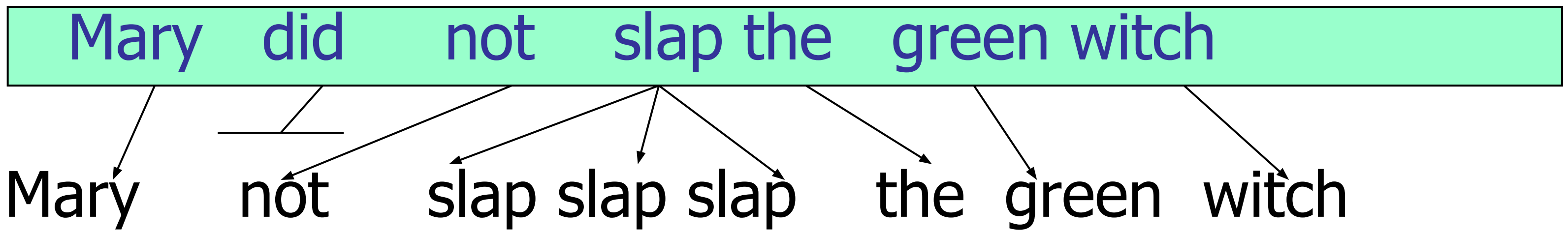
$\mathbf{a} = ???$ $\mathbf{a} = (1, 2, (3, 4) \quad)$?

Generative Story

$$p(\mathbf{e} \mid \mathbf{f}, m) \text{ ?}$$

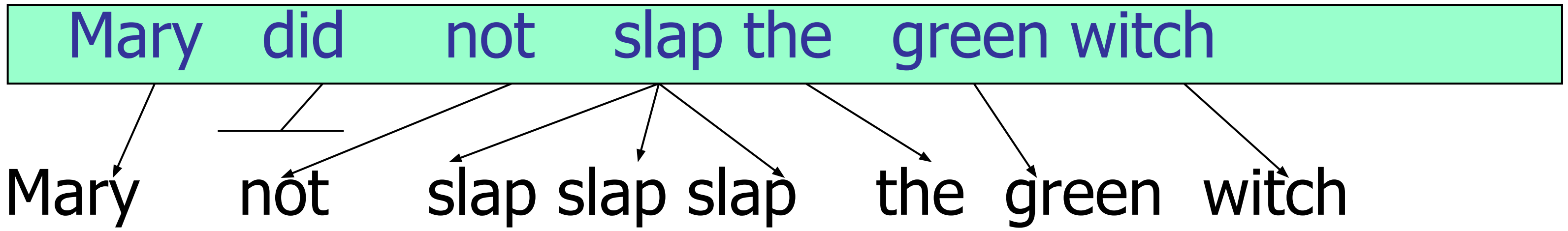
Mary did not slap the green witch

Generative Story



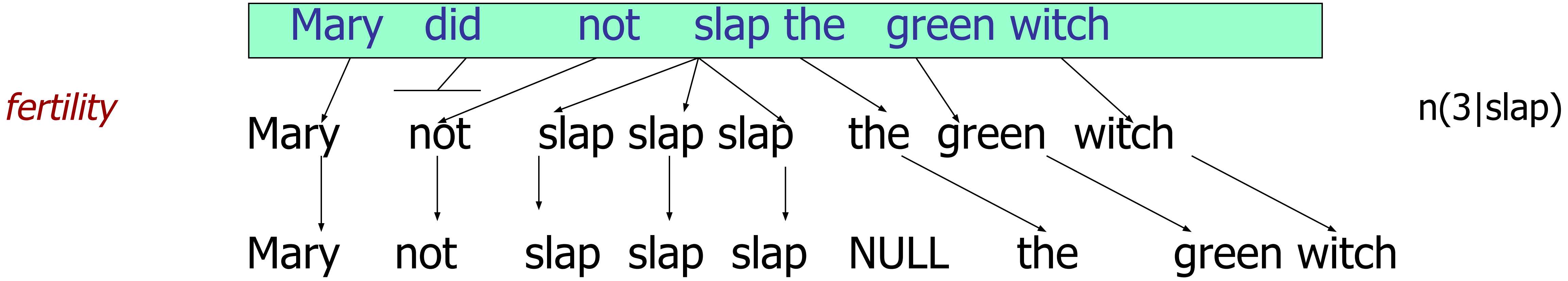
Generative Story

fertility

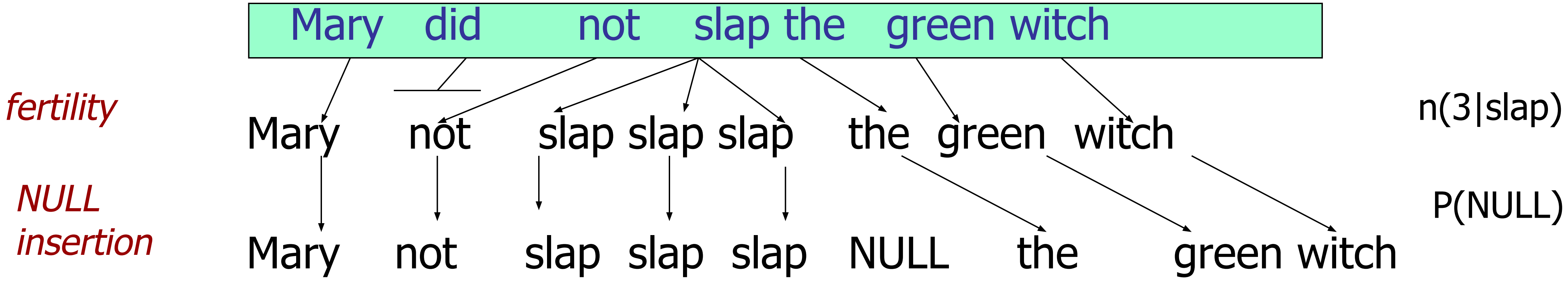


n(3|slap)

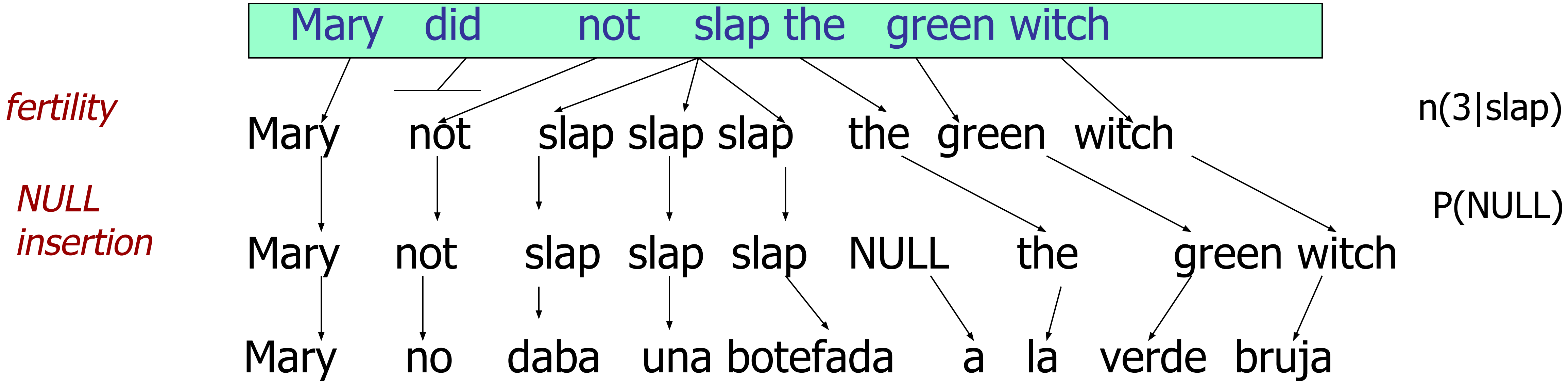
Generative Story



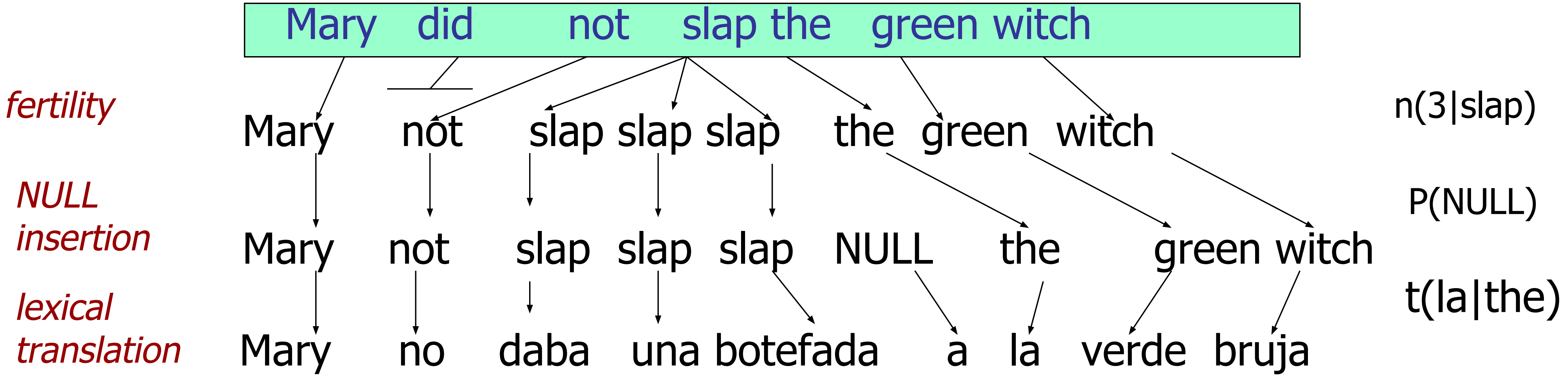
Generative Story



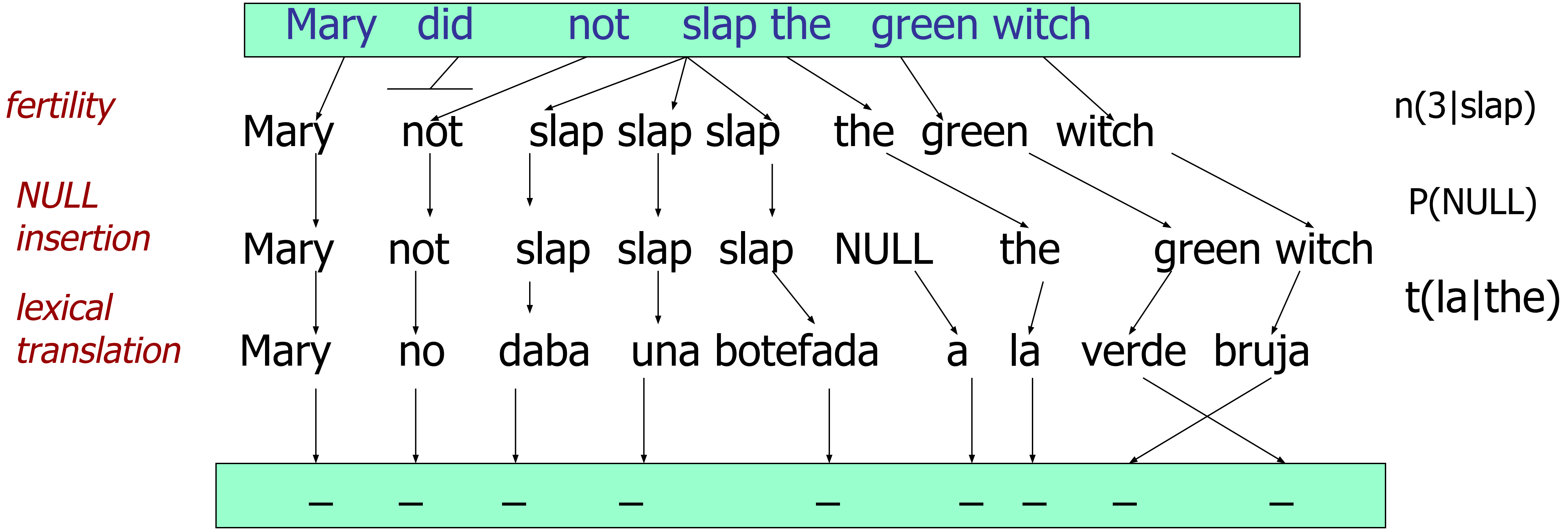
Generative Story



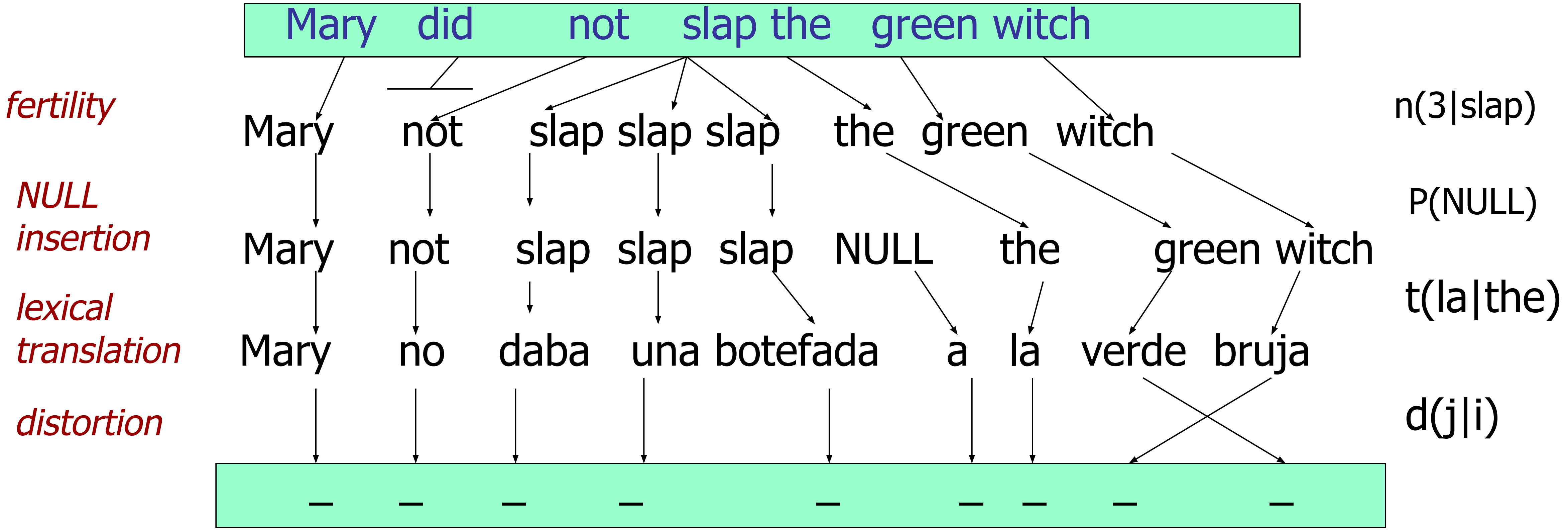
Generative Story



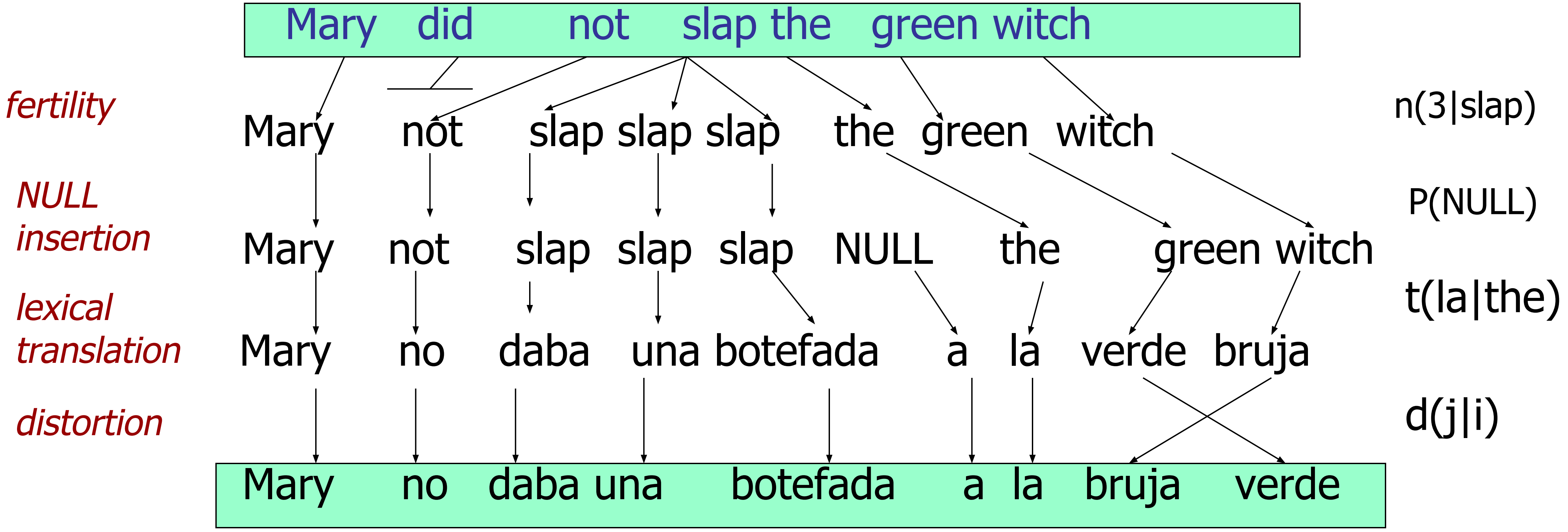
Generative Story



Generative Story



The IBM Models 1--5 (Brown et al. 93)

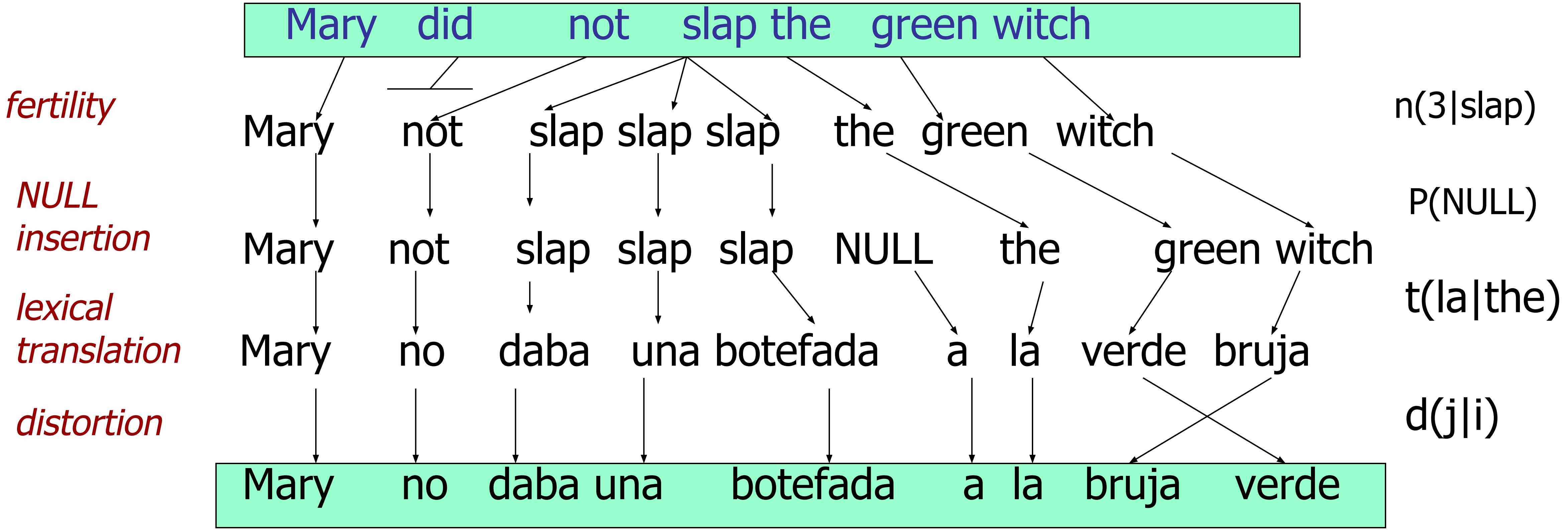


[from Al-Onaizan and Knight, 1998]

Alignment Models

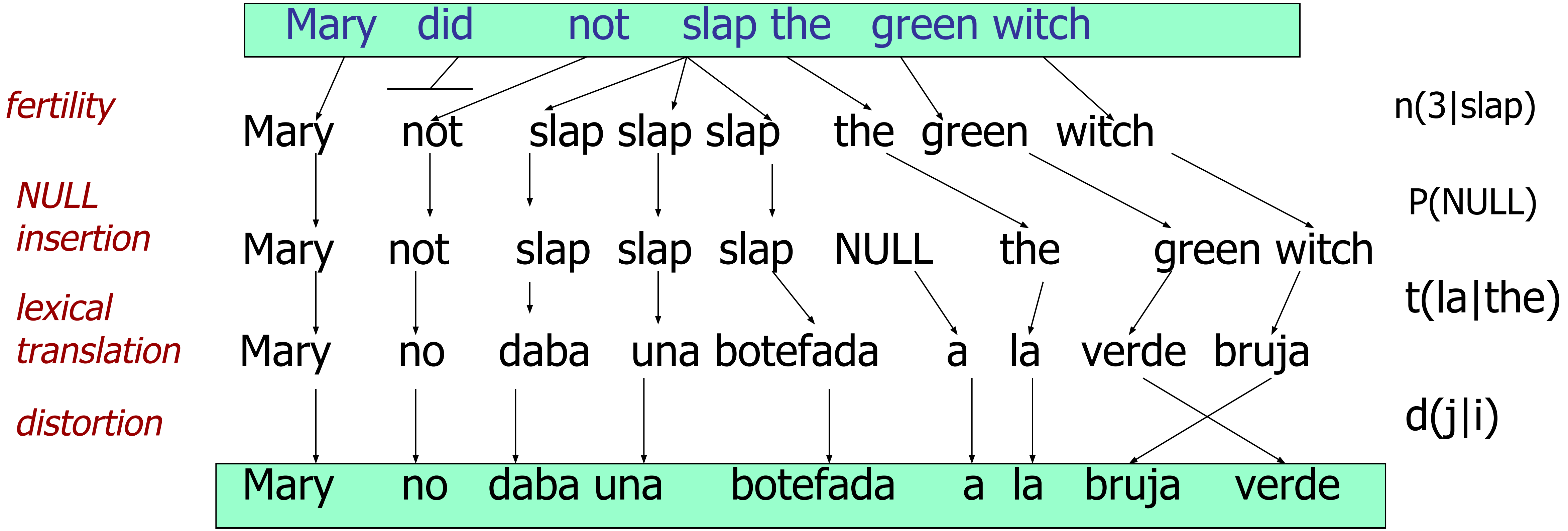
- IBM Model 1: lexical translation
- IBM Model 2: alignment model, global monotonicity
- HMM model: local monotonicity
- *fastalign:* efficient reparametrization of Model 2
- IBM Model 3: fertility
- IBM Model 4: relative alignment model
- IBM Model 5: deficiency
- +many more

P(e,a|f)



$$P(e, alignment|f) = \prod p_f \prod p_t \prod p_d$$

P(e|f)



$$P(e|f) = \sum_{all\_possible\_alignments} \prod p_f \prod p_t \prod p_d$$

IBM Model 1

- Generative model: break up translation process into smaller steps
- Simplest possible lexical translation model
- Additional assumptions
  - All alignment decisions are independent
  - The alignment distribution for each a_i is uniform over all source words and NULL

IBM Model 1

- Translation probability
  - for a foreign sentence **f** = (f_1, ..., f_{lf}) of length l_f
  - to an English sentence **e** = (e_1, ..., e_{le}) of length l_e
  - with an alignment of each English word e_j to a foreign word f_i according to the alignment function $a : j \longrightarrow i$

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

- parameter ϵ is a normalization constant

Example

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

$$p(e,a|f) = \frac{\epsilon}{4^3} \times t(\text{the}|\text{das}) \times t(\text{house}|\text{Haus}) \times t(\text{is}|\text{ist}) \times t(\text{small}|\text{klein})$$

$$= \frac{\epsilon}{4^3} \times 0.7 \times 0.8 \times 0.8 \times 0.4$$

$$= 0.0028\epsilon$$

Learning Lexical Translation Models

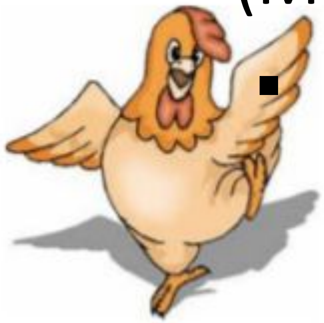
We would like to estimate the lexical translation probabilities $t(e|f)$ from a parallel corpus

- ... 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



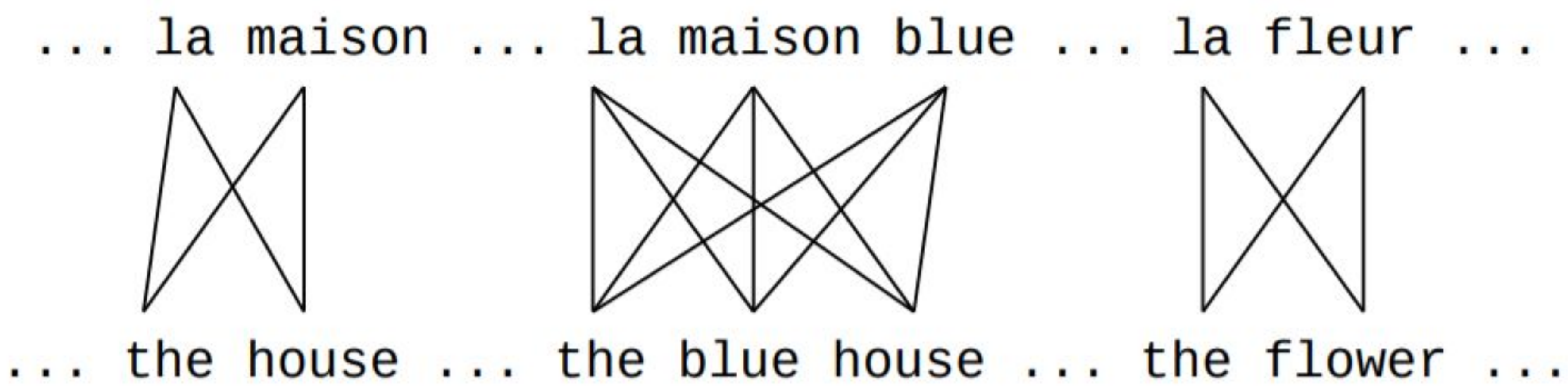
EM Algorithm

- 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

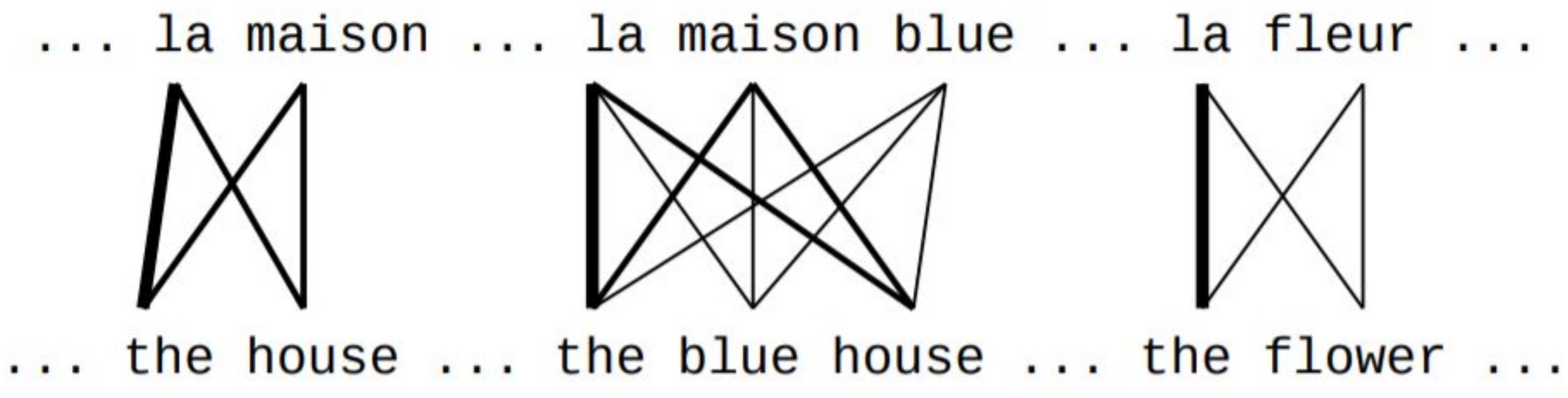
1. initialize model parameters (e.g. uniform, random)
2. assign probabilities to the missing data
3. estimate model parameters from completed data
4. iterate steps 2–3 until convergence

EM Algorithm



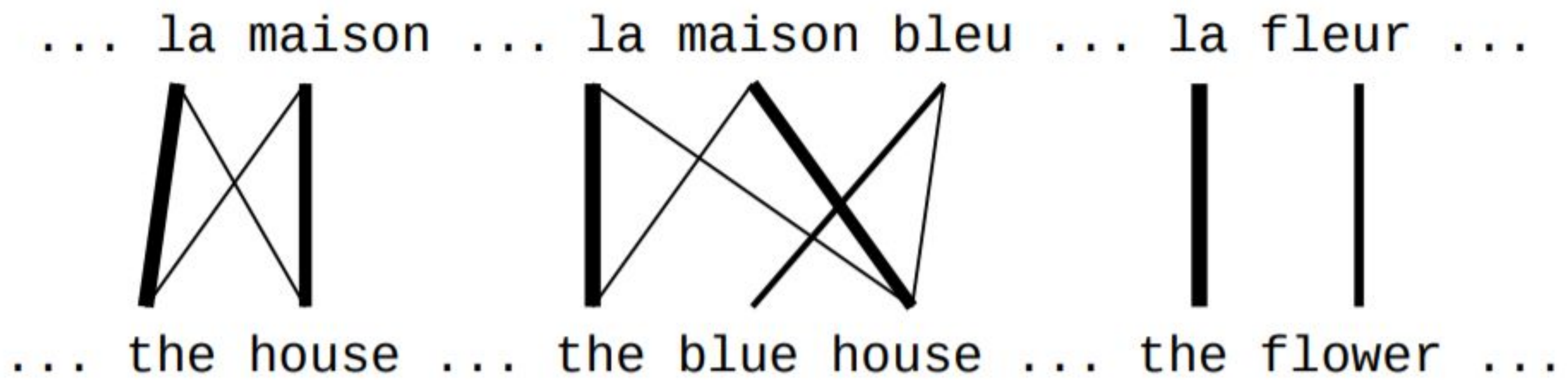
- Initial step: all alignments equally likely
- Model learns that, e.g., *la* is often aligned with *the*

EM Algorithm



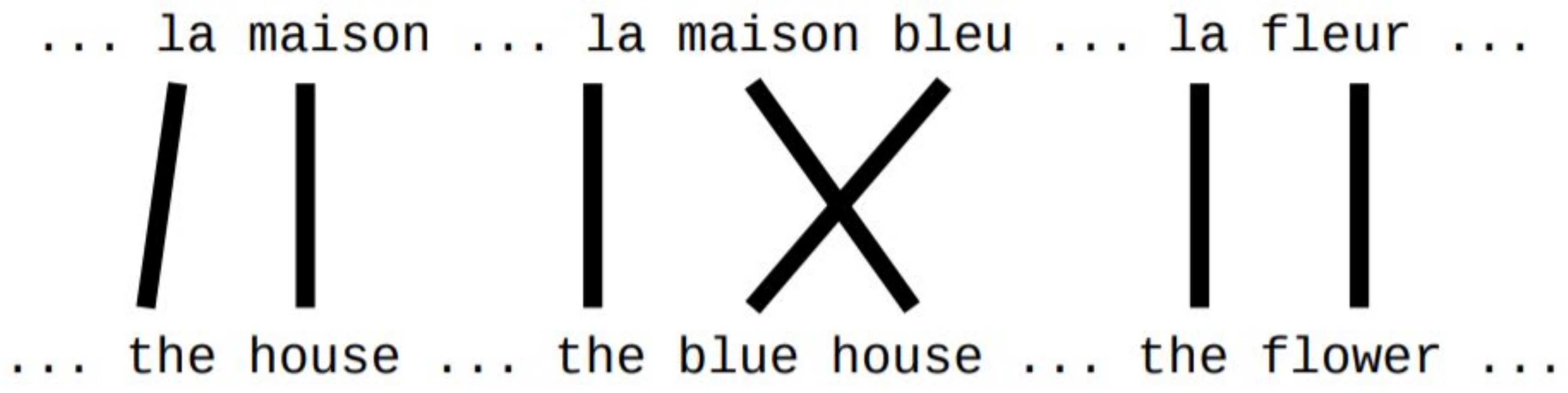
- After one iteration
- Alignments, e.g., between *la* and *the* are more likely

EM Algorithm



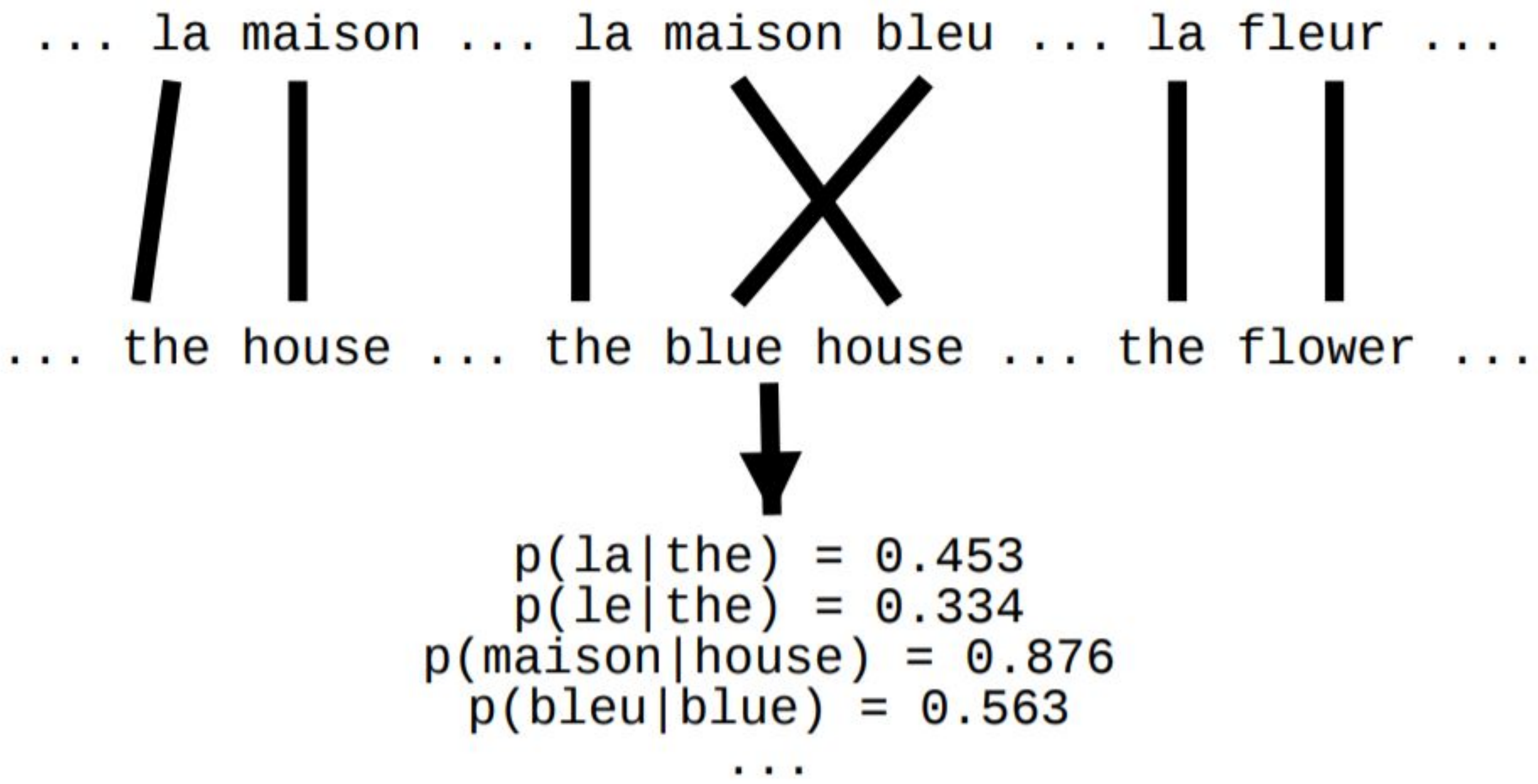
- After another iteration
- It becomes apparent that alignments, e.g., between *fleur* and *flower* are more likely (pigeon hole principle)

EM Algorithm



- Convergence
- Inherent hidden structure revealed by EM

EM Algorithm



- Parameter estimation from the aligned corpus

IBM Model 1 and EM

EM Algorithm consists of two steps

- Expectation-Step: Apply model to the data
    - parts of the model are hidden (here: alignments)
    - using the model, assign probabilities to possible values
- Maximization-Step: Estimate model from data
    - take assigned values as fact
    - collect counts (weighted by lexical translation probabilities)
    - estimate model from counts
- Iterate these steps until convergence

IBM Model 1 and EM

- We need to be able to compute:
  - Expectation-Step: probability of alignments
  - Maximization-Step: count collection

IBM Model 1 and EM

t-table **Probabilities**

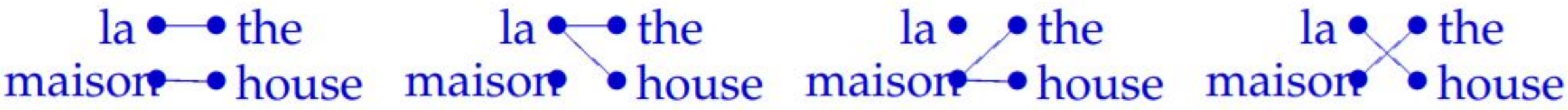
$$p(\text{the}|\text{la}) = 0.7 \qquad p(\text{house}|\text{la}) = 0.05$$
$$p(\text{the}|\text{maison}) = 0.1 \qquad p(\text{house}|\text{maison}) = 0.8$$

IBM Model 1 and EM

t-table **Probabilities**

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

Alignments



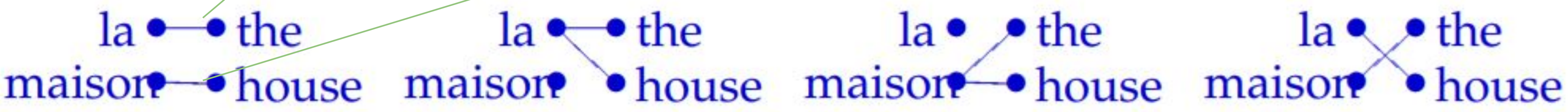
IBM Model 1 and EM

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



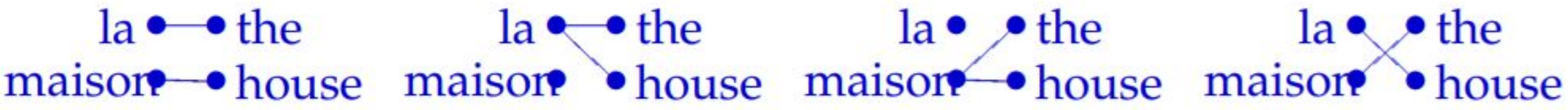
$p(\mathbf{e}, a|\mathbf{f}) = 0.56$ $p(\mathbf{e}, a|\mathbf{f}) = 0.035$ $p(\mathbf{e}, a|\mathbf{f}) = 0.08$ $p(\mathbf{e}, a|\mathbf{f}) = 0.005$

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



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

Applying the chain rule:

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

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

IBM Model 1 and EM: Expectation Step

We need to compute $p(\mathbf{e}|\mathbf{f})$

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

$$= \sum_{a(1)=0}^{l_f} \cdots \sum_{a(l_e)=0}^{l_f} p(\mathbf{e}, a|\mathbf{f})$$

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

IBM Model 1 and EM: Expectation Step

$$p(\mathbf{e}|\mathbf{f}) = \sum_{a(1)=0}^{l_f} \dots \sum_{a(l_e)=0}^{l_f} \frac{\epsilon}{(l_f+1)^{l_e}} \prod_{j=1}^{l_e} t(e_j|f_{a(j)})$$

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

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

- Note the trick in the last line
  - removes the need for an exponential number of products

→ this makes IBM Model 1 estimation tractable

The Trick

(case $l_e = l_f = 2$)

$$\sum_{a(1)=0}^{2} \sum_{a(2)=0}^{2} = \frac{\epsilon}{3^2} \prod_{j=1}^{2} t(e_j|f_{a(j)}) =$$

$$= t(e_1|f_0)\, t(e_2|f_0) + t(e_1|f_0)\, t(e_2|f_1) + t(e_1|f_0)\, t(e_2|f_2) +$$

$$+ t(e_1|f_1)\, t(e_2|f_0) + t(e_1|f_1)\, t(e_2|f_1) + t(e_1|f_1)\, t(e_2|f_2) +$$

$$+ t(e_1|f_2)\, t(e_2|f_0) + t(e_1|f_2)\, t(e_2|f_1) + t(e_1|f_2)\, t(e_2|f_2) =$$

$$= t(e_1|f_0)\,(t(e_2|f_0) + t(e_2|f_1) + t(e_2|f_2)) +$$

$$+ t(e_1|f_1)\,(t(e_2|f_1) + t(e_2|f_1) + t(e_2|f_2)) +$$

$$+ t(e_1|f_2)\,(t(e_2|f_2) + t(e_2|f_1) + t(e_2|f_2)) =$$

$$= (t(e_1|f_0) + t(e_1|f_1) + t(e_1|f_2))\,(t(e_2|f_2) + t(e_2|f_1) + t(e_2|f_2))$$

IBM Model 1 and EM: Expectation Step

Combine what we have:

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

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

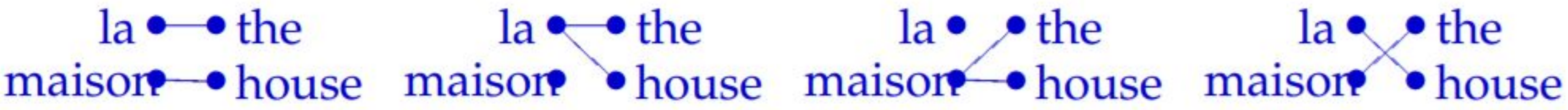
$$= \prod_{j=1}^{l_e} \frac{t(e_j|f_{a(j)})}{\sum_{i=0}^{l_f} t(e_j|f_i)}$$

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

Now we have to collect counts

Evidence from a sentence pair **e,f** that word e is a translation of word f:

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

With the same simplication as before:

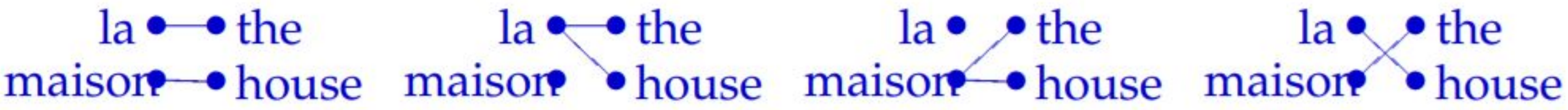
$$c(e|f;\mathbf{e},\mathbf{f}) = \frac{t(e|f)}{\sum_{i=0}^{l_f} t(e|f_i)} \sum_{j=1}^{l_e} \delta(e,e_j) \sum_{i=0}^{l_f} \delta(f,f_i)$$

IBM Model 1 and EM: Maximization 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



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

E-step

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

M-step **Counts**

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

IBM Model 1 and EM: Maximization Step

After collecting these counts over a corpus, we can estimate the model:

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

IBM Model 1 and EM: Maximization Step

t-table **Probabilities**

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

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**

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

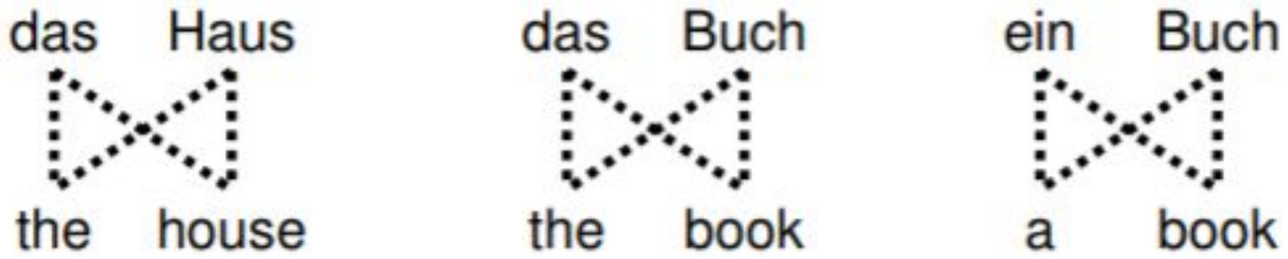
Update t-table:

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

IBM Model 1 and EM: Pseudocode

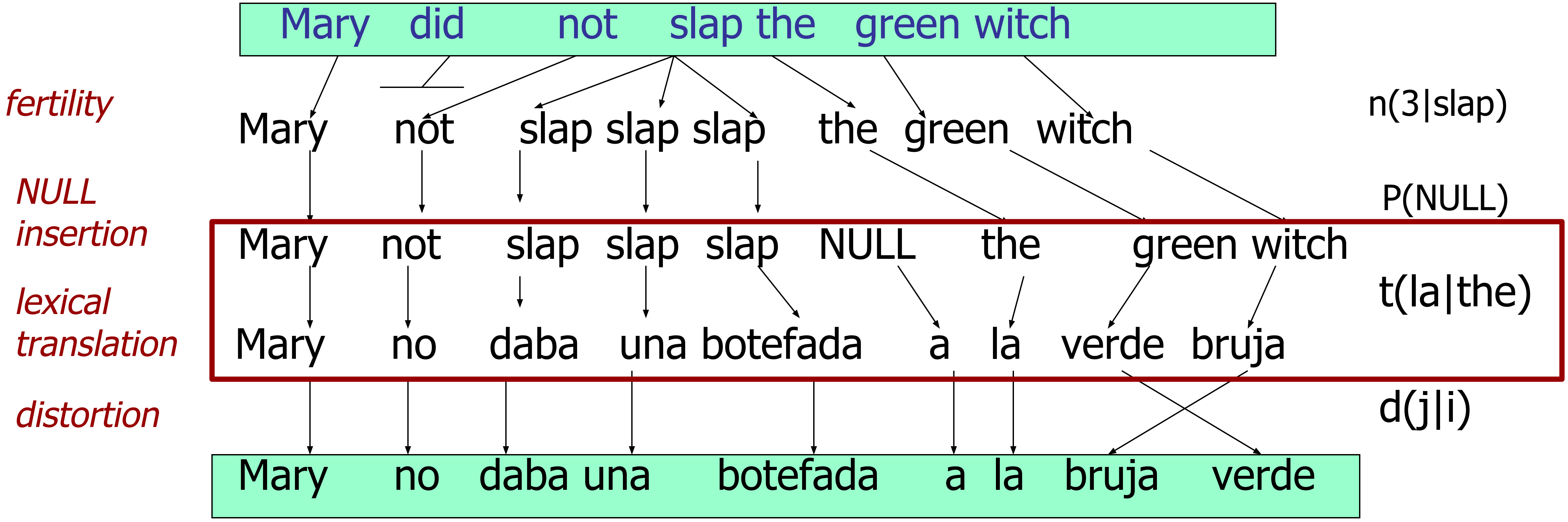
```
Input: set of sentence pairs (e, f)
Output: translation prob. t(e|f)
initialize t(e|f) uniformly
while not converged do
    // initialize
    count(e|f) = 0 for all e, f
    total(f) = 0 for all f
    for all sentence pairs (e,f) do
        // compute normalization
        for all words e in e do
            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
```

Convergence

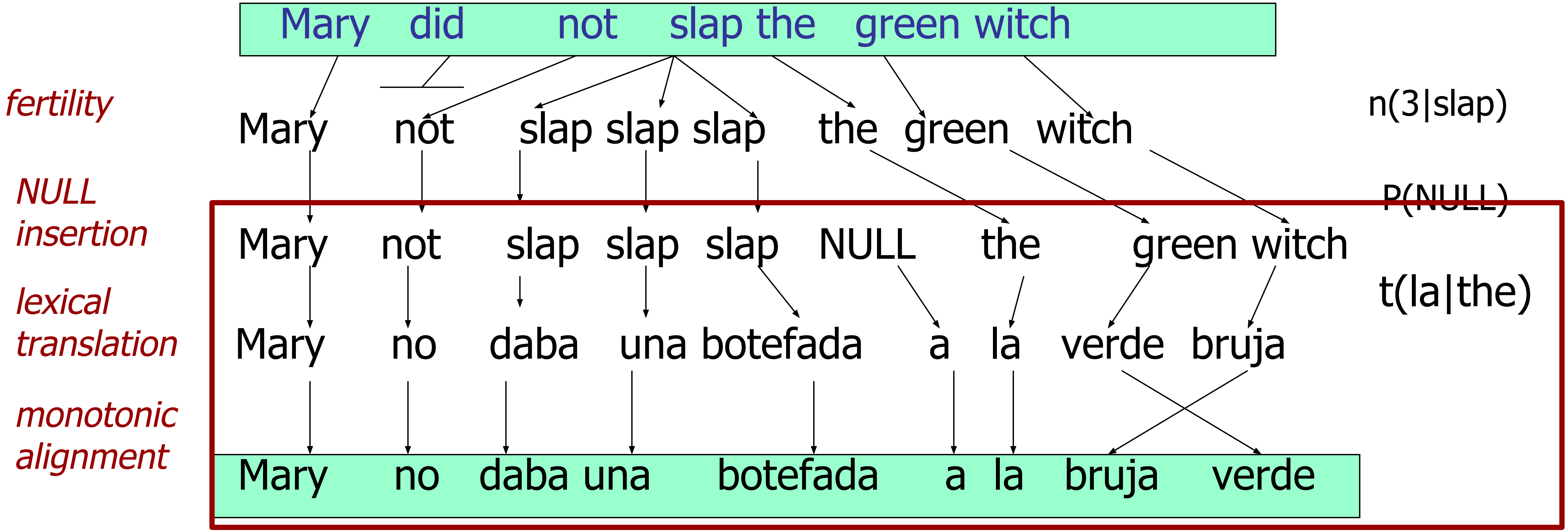


e	*f*	initial	1st it.	2nd it.	3rd it.	...	final
the	das	0.25	0.5	0.6364	0.7479	...	1
book	das	0.25	0.25	0.1818	0.1208	...	0
house	das	0.25	0.25	0.1818	0.1313	...	0
the	buch	0.25	0.25	0.1818	0.1208	...	0
book	buch	0.25	0.5	0.6364	0.7479	...	1
a	buch	0.25	0.25	0.1818	0.1313	...	0
book	ein	0.25	0.5	0.4286	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

Problems with IBM Model 1



IBM Model 2


Mary did not slap the green witch
fertility
Mary not slap slap slap the green witch
n(3|slap)
NULL insertion
Mary not slap slap slap NULL the green witch
P(NULL)
lexical translation
Mary no daba una botefada a la verde bruja
t(la|the)
monotonic alignment
Mary no daba una botefada a la bruja verde

IBM Model 2

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

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

- compare with Model 1:

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

Higher IBM Models

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

Only IBM Model 1 has global maximum

– training of a higher IBM model builds on previous model

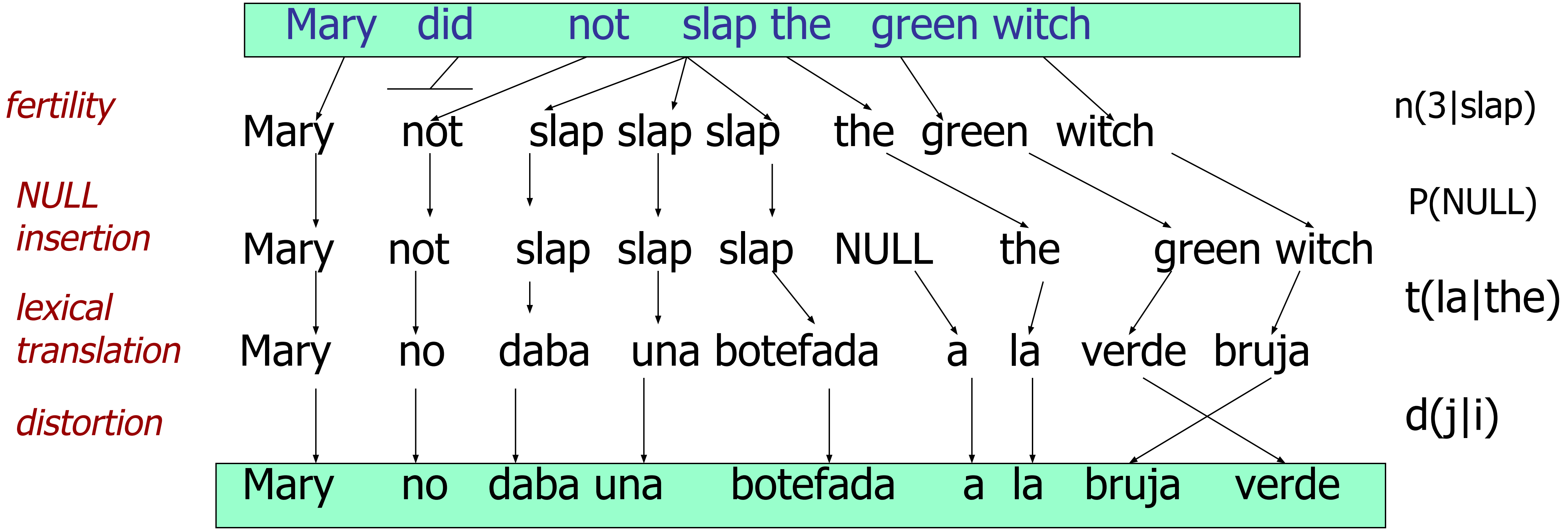
Computationally biggest change in Model 3

– trick to simplify estimation does not work anymore

→ exhaustive count collection becomes computationally too expensive

– sampling over high probability alignments is used instead

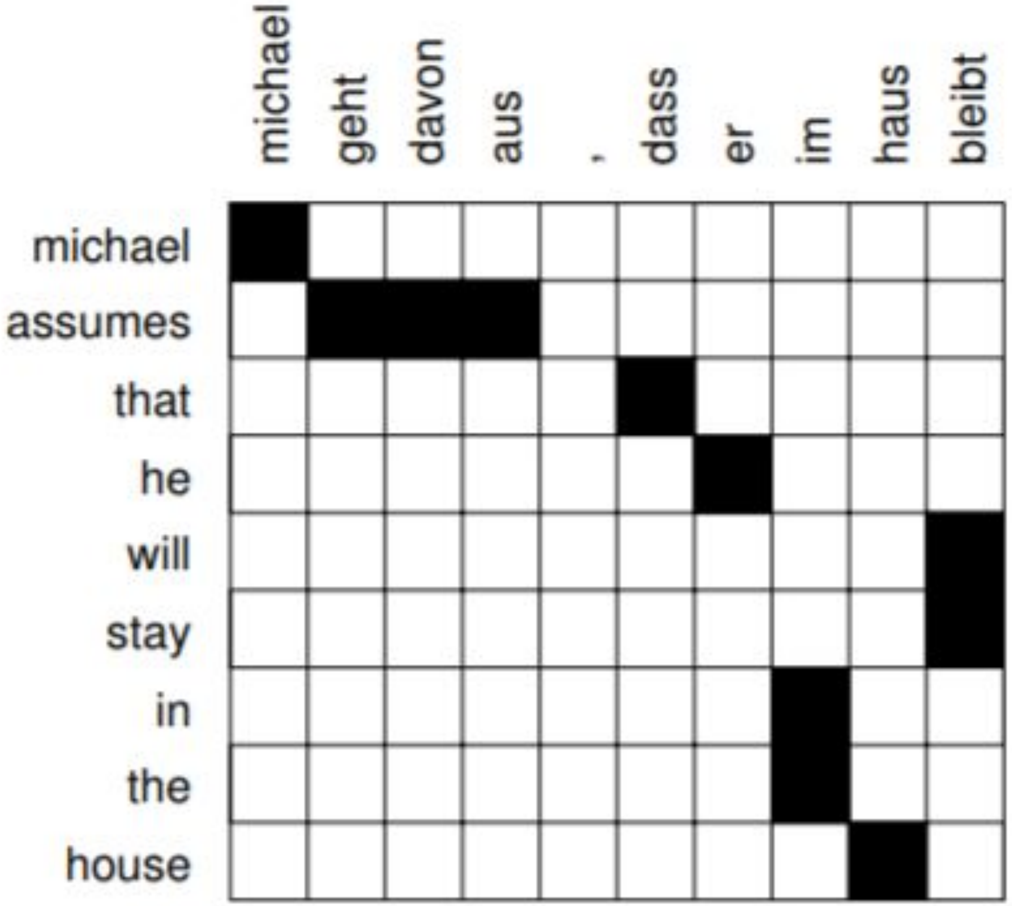
The IBM Models 1--5 (Brown et al. 93)



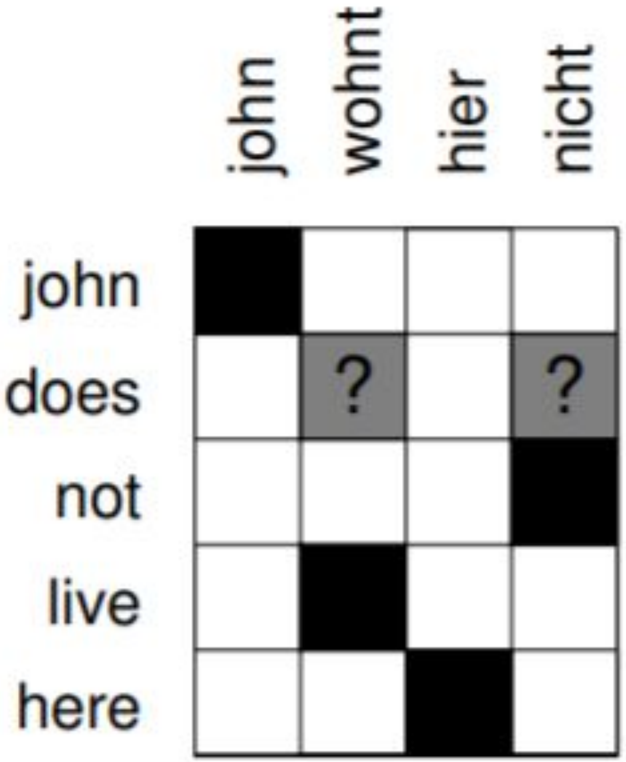
[from Al-Onaizan and Knight, 1998]

Word Alignment

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

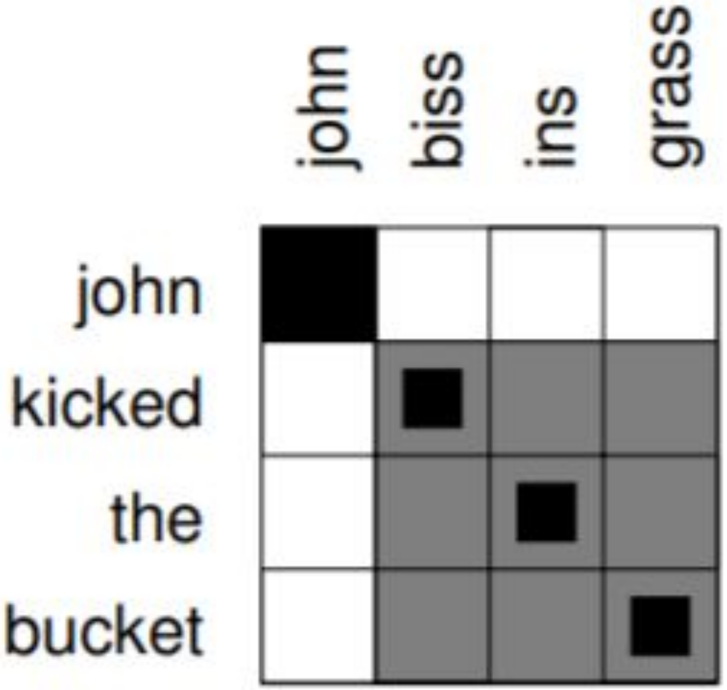


Word Alignment?



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

Word Alignment?

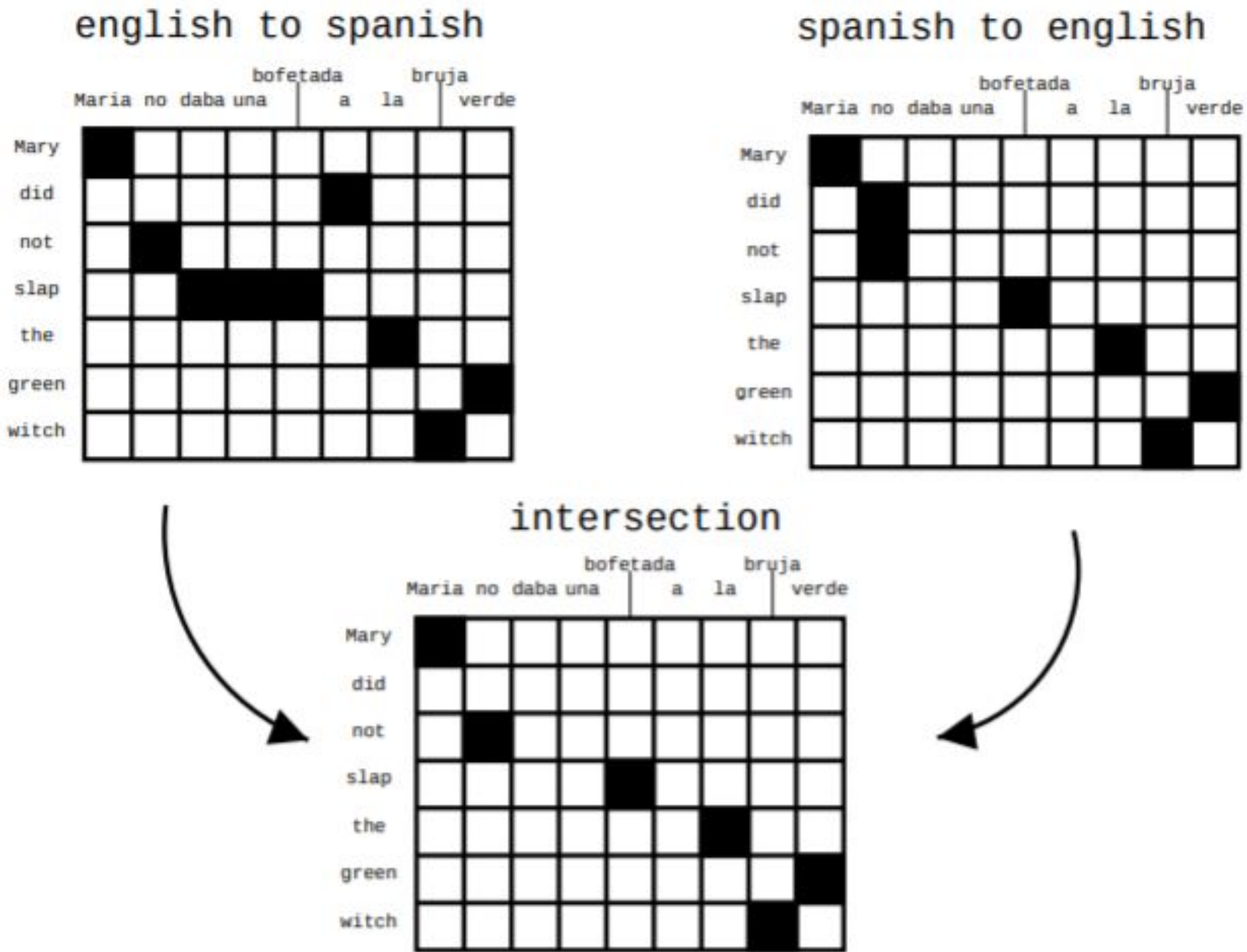


How do the idioms kicked the bucket and biss ins grass match up?
Outside this exceptional context, bucket is never a good translation for grass

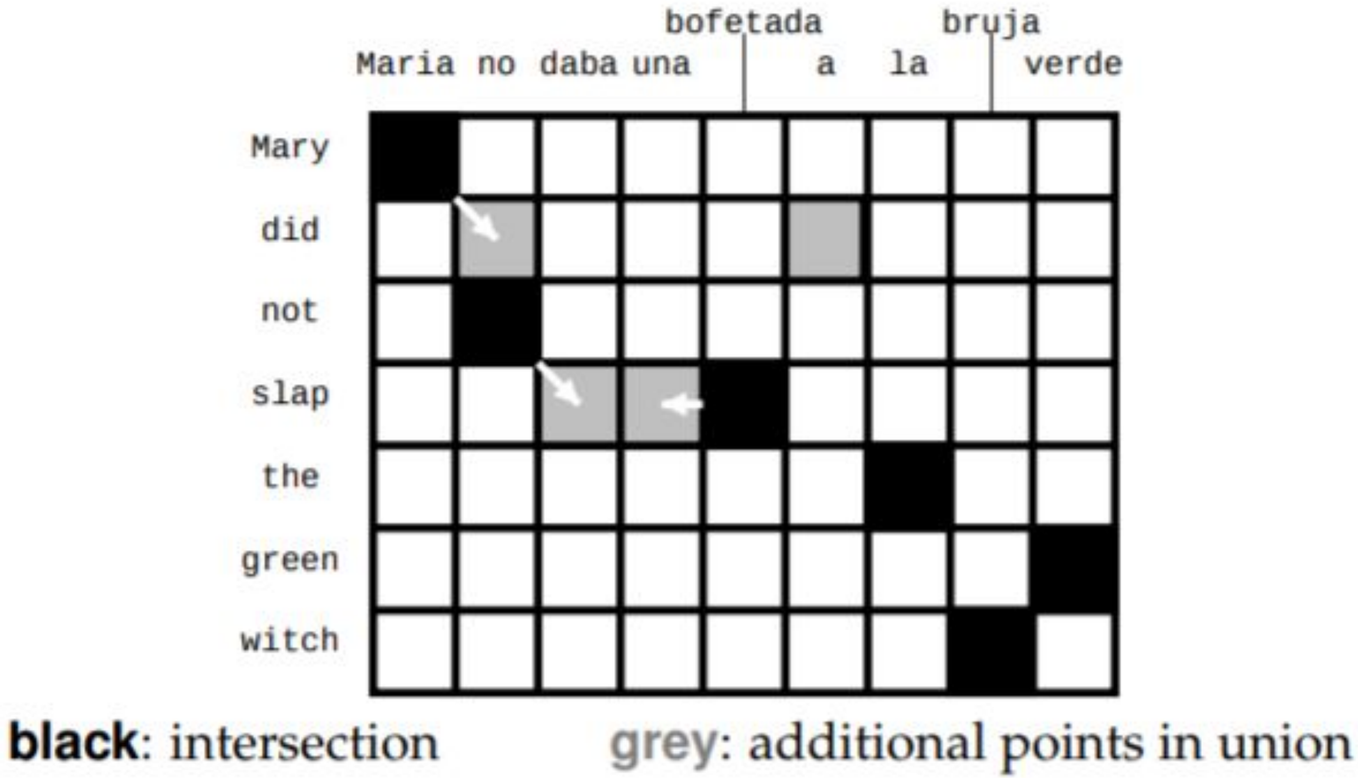
Word Alignment and IBM Models

- 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

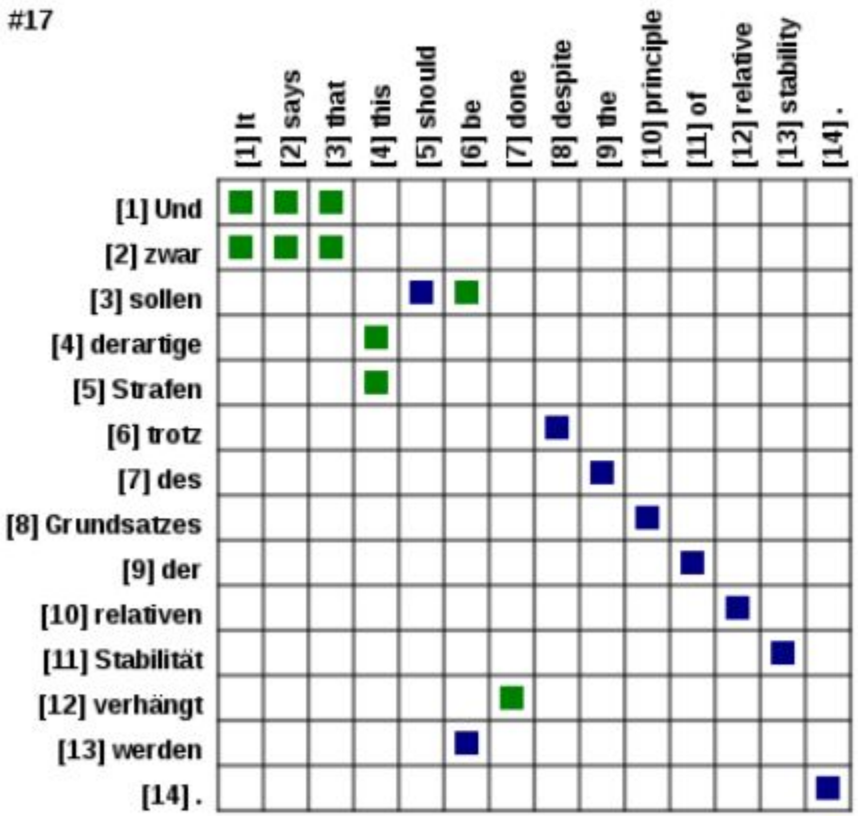


- Add alignment points from union based on heuristics
- Popular method: grow-diag-final-and

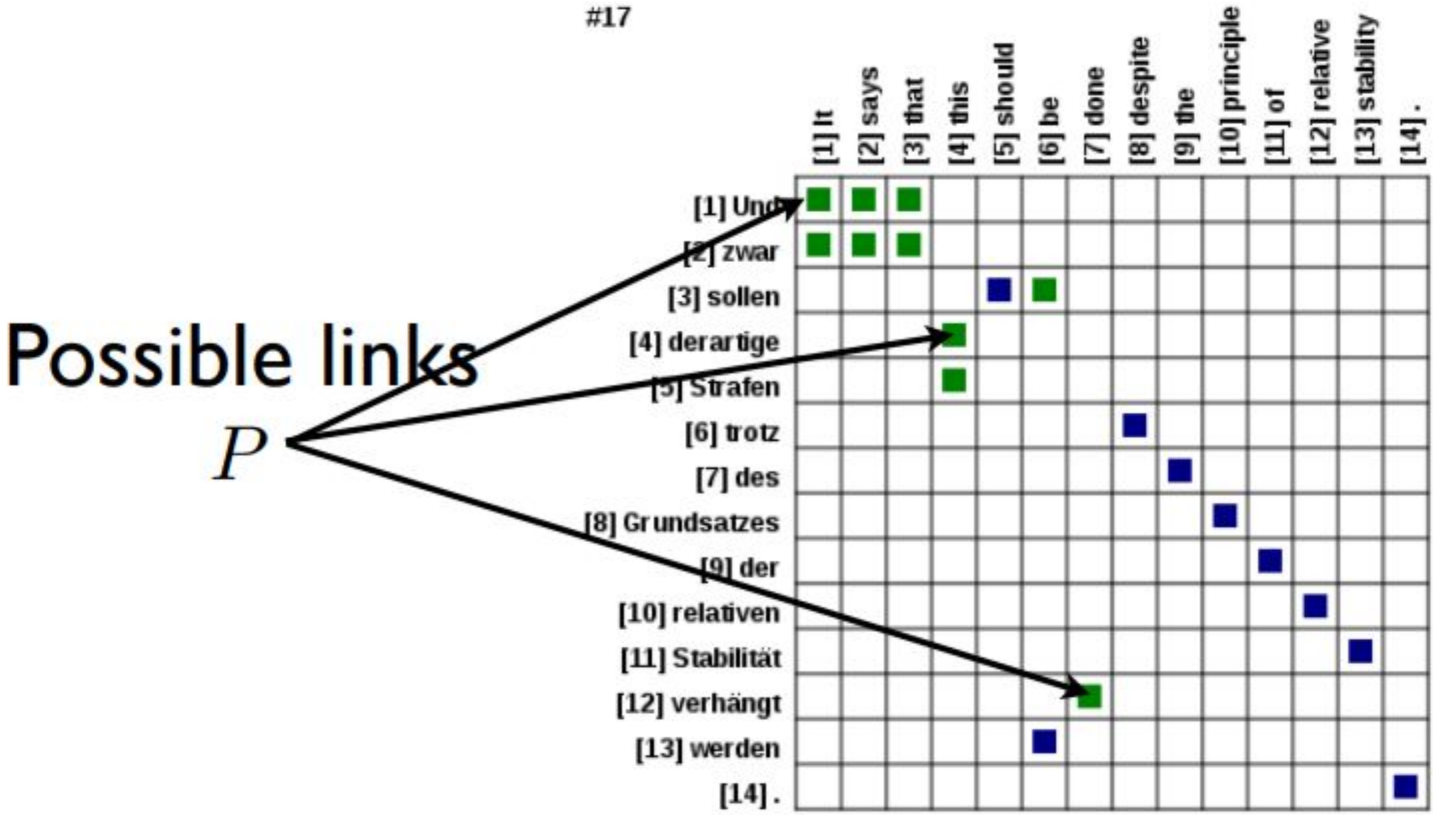
Evaluating Alignment Models

- How do we measure quality of a word-to-word model?
  - Method 1: use in an end-to-end translation system
    - Hard to measure translation quality
    - Option: human judges
    - Option: reference translations (NIST, BLEU)
    - Option: combinations (HTER)
    - Actually, no one uses word-to-word models alone as TMs
  - Method 2: measure quality of the alignments produced
    - Easy to measure
    - Hard to know what the gold alignments should be
    - Often does not correlate well with translation quality (like perplexity in LMs)

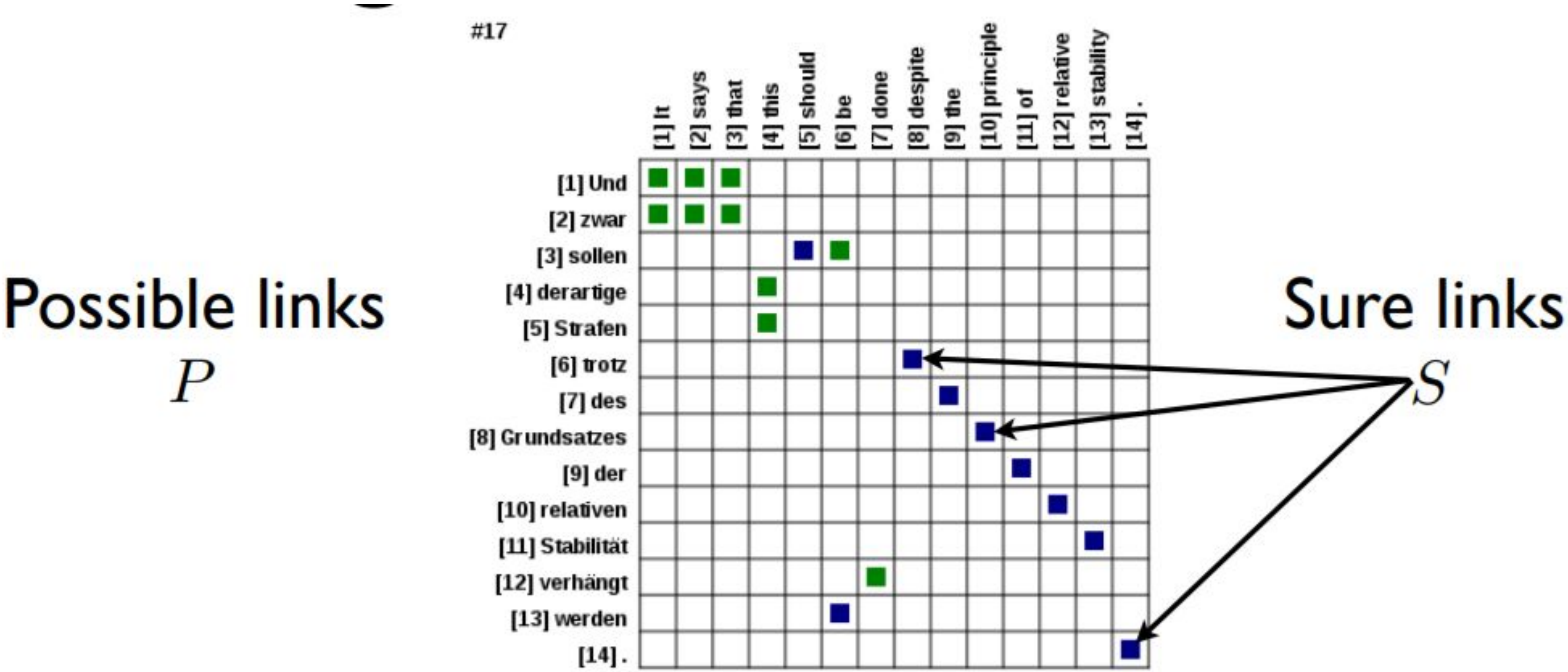
Alignment Error Rate



Alignment Error Rate

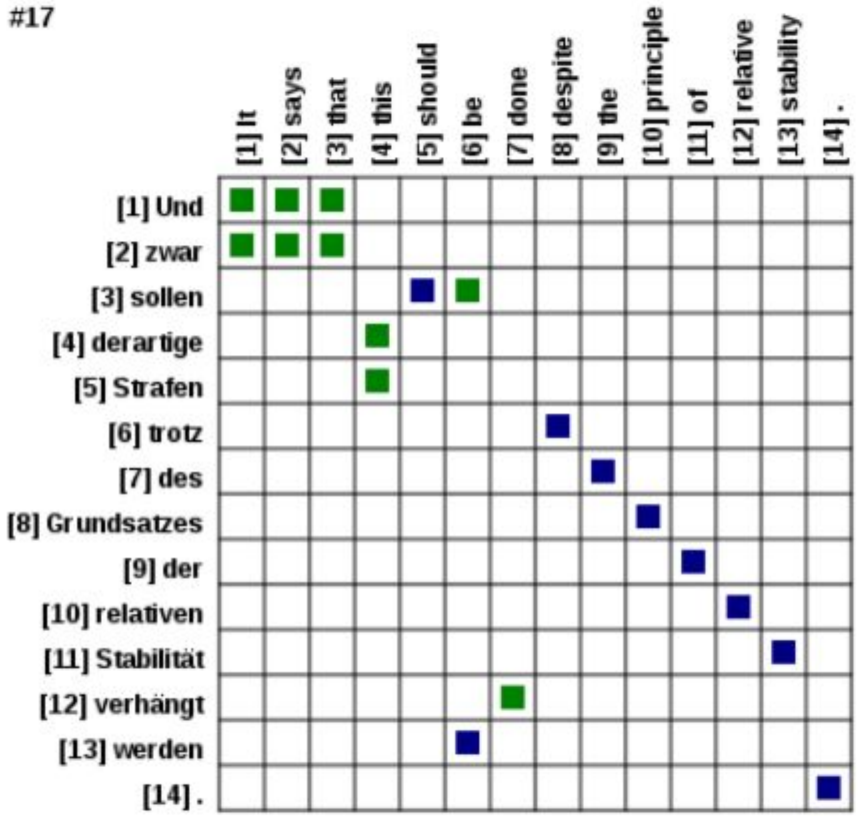


Alignment Error Rate



Alignment Error Rate

Possible links
P



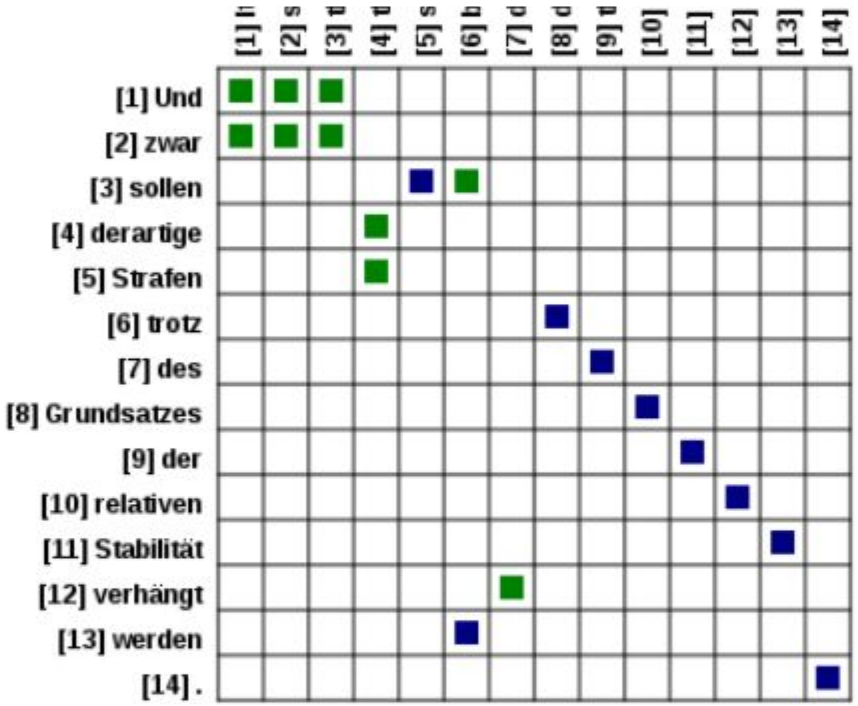
Sure links
S

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

Alignment Error Rate

Possible links

P



Sure links

S

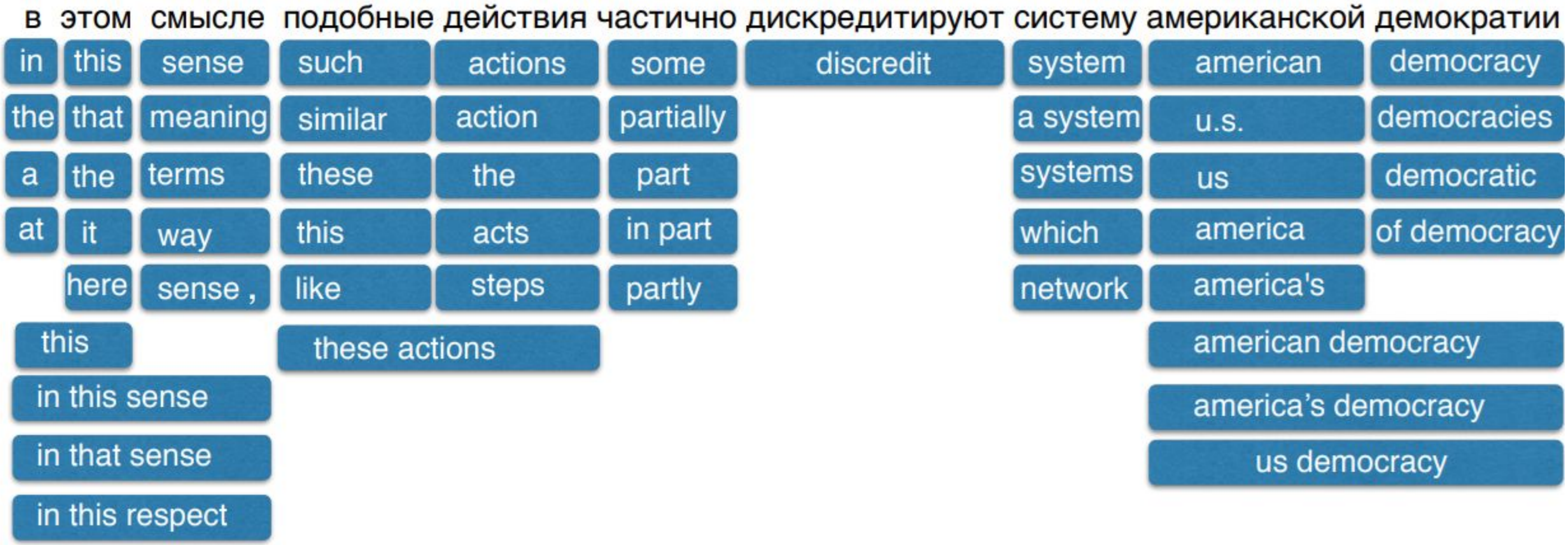
$$\text{Precision}(A, P) = \frac{|P \cap A|}{|A|} \qquad \text{Recall}(A, S) = \frac{|S \cap A|}{|S|}$$

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

Problems with Lexical Translation

- Complexity -- exponential in sentence length
- Weak reordering -- the output is not fluent
- Many local decisions -- error propagation

Phrase-Based Translation



$$P(e, \text{alignment}|f) = p_{segmentation}\, p_{\text{translation}}\, p_{reorderings}$$

Phrase-Based MT