Word Alignment by IBM Models

Taro Watanabe

Statistical Machine Translation

$$
\begin{aligned}
\hat{\mathbf{e}} &= \operatorname*{argmax}_{\mathbf{e}} Pr(\mathbf{e}|\mathbf{f}) \\
&= \operatorname*{argmax}_{\mathbf{e}} Pr(\mathbf{f}|\mathbf{e}) Pr(\mathbf{e})
\end{aligned}
$$

- Brown et al. 1993. The mathematics of statistical machine translation: Parameter estimation. Computational Linguistics, 19(2):263-311 (http://www.aclweb.org/anthology/J/J93/J93-2003.pdf)
- Decomposed into translation model of p(f|e) and language model of p(e)

Language Model

$$
\begin{aligned}
Pr(\text{I do not know}) &= ? \\
Pr(\text{I not do know}) &= ?
\end{aligned}
$$

Language Model

$$
\begin{aligned}
Pr(\text{I do not know}) &= ? \\
Pr(\text{I not do know}) &= ?
\end{aligned}
$$

- Likelihood of a string of English words

Language Model

$$
\begin{aligned}
Pr(\text{I do not know}) &= ? \\
Pr(\text{I not do know}) &= ?
\end{aligned}
$$

- Likelihood of a string of English words
- Usually modeled by ngrams

$$W = w_1, w_2, w_3, \cdots w_N$$

Language Model

$$
\begin{aligned}
Pr(\text{I do not know}) &= ? \\
Pr(\text{I not do know}) &= ?
\end{aligned}
$$

- Likelihood of a string of English words
- Usually modeled by ngrams

$$W = w_1, w_2, w_3, \cdots w_N$$

$$
\begin{aligned}
p(W) &= p(w_1, w_2, w_3, \cdots, w_N) \\
&= p(w_1)p(w_2|w_1)p(w_3|w_1, w_2)\cdots \\
&\qquad p(w_N|w_1, w_2, w_3, \cdots, w_{N-1})
\end{aligned}
$$

ngram Language Model

ngram Language Model

- Markov assumption: only n-words are memories in the history
- Bigram:

$$p(\text{I do not know}) \quad = \quad p(\text{I})p(\text{do}|\text{I})p(\text{not}|\text{do})p(\text{know}|\text{not})$$

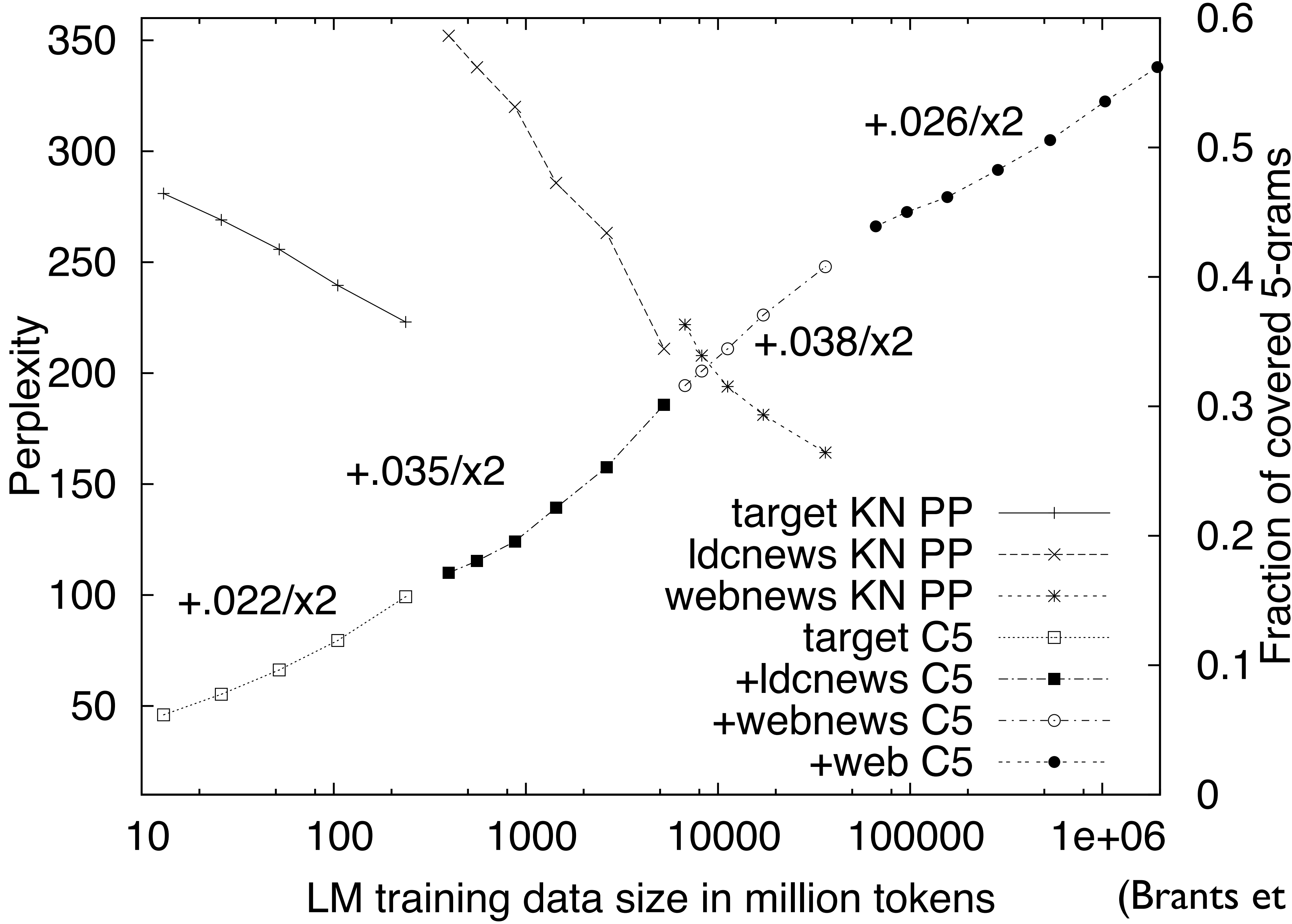
ngram Language Model

- Markov assumption: only n-words are memories in the history
- Bigram:

$$p(\text{I do not know}) = p(\text{I})p(\text{do}|\text{I})p(\text{not}|\text{do})p(\text{know}|\text{not})$$

- Training: Maximum likelihood estimate + smoothing (Good-Turing, Witten-Bell, Kneser-Ney etc.)

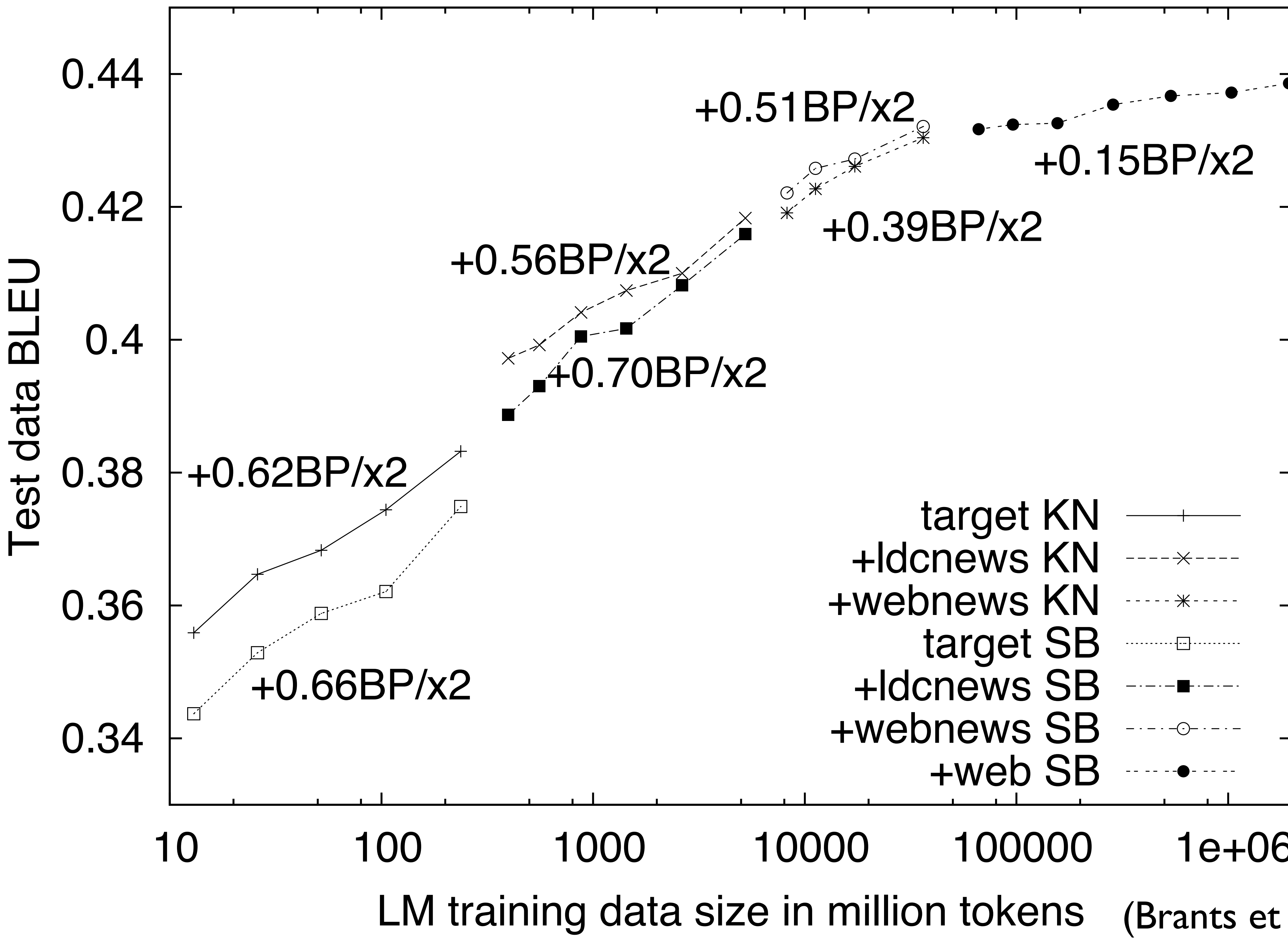
Larger Data, Better LM



(Brants et al., 2007)

- Entropy: $-\frac{1}{N}\log_2 p(W_1^N)$ Perplexity: $2^{-\frac{1}{N}\log_2 p(W_1^N)}$

Better LM, Better MT



(Brants et al., 2007)

Translation Model

$$
\begin{aligned}
\mathbf{f} &= \text{je ne sais pas} \\
\mathbf{e} &= \text{I do not know}
\end{aligned}
$$

$$Pr(\mathbf{f}|\mathbf{e}) = ??$$

Translation Model

$$
\begin{aligned}
\mathbf{f} &= \text{je ne sais pas} \\
\mathbf{e} &= \text{I do not know}
\end{aligned}
$$

$$Pr(\mathbf{f}|\mathbf{e}) = ??$$

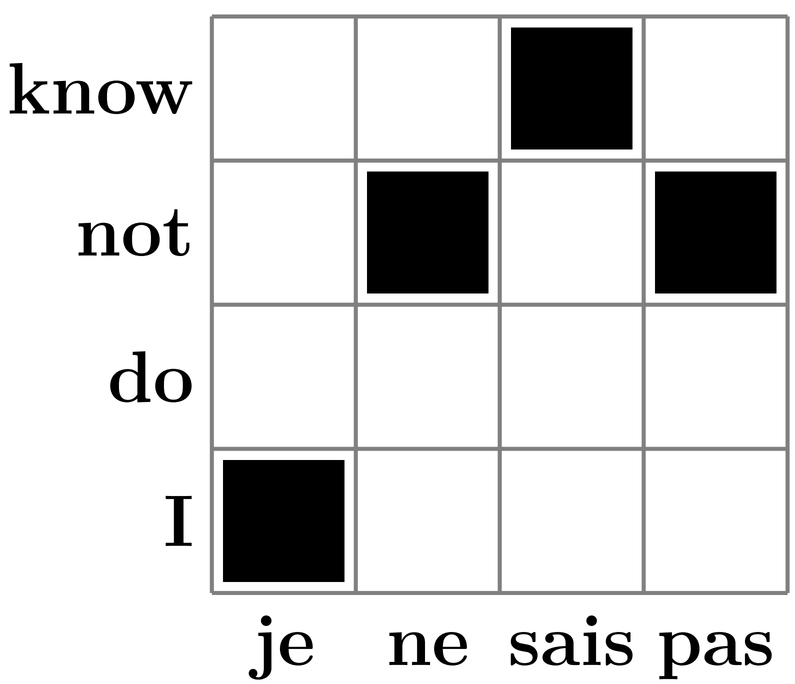
- 5 Models with increasing complexity: Model 1 to Model 5
- We will concentrate on Model 1:
  - How to represent P(f|e)
  - How to estimate P(f|e)

Alignment Representation

$$Pr(\mathbf{f}|\mathbf{e}) = \sum_{\mathbf{a}} Pr(\mathbf{f}, \mathbf{a}|\mathbf{e})$$

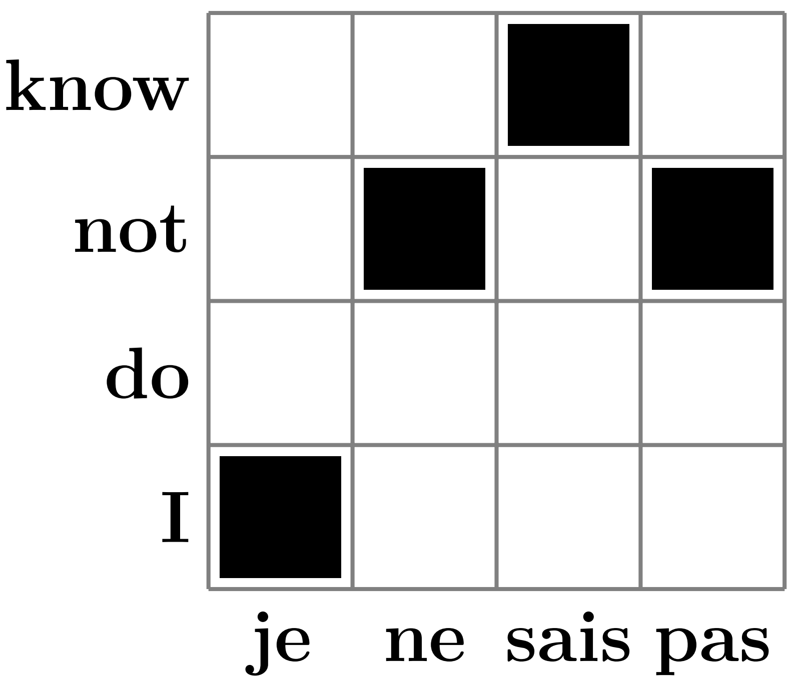
Alignment Representation

$$Pr(\mathbf{f}|\mathbf{e}) = \sum_{\mathbf{a}} Pr(\mathbf{f}, \mathbf{a}|\mathbf{e})$$



Alignment Representation

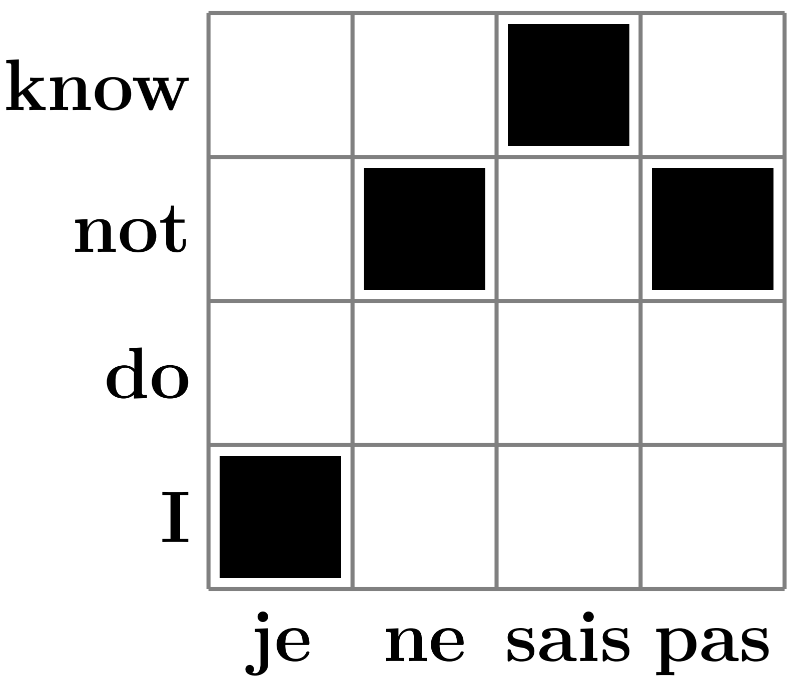
$$Pr(\mathbf{f}|\mathbf{e}) = \sum_{\mathbf{a}} Pr(\mathbf{f}, \mathbf{a}|\mathbf{e})$$



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

Alignment Representation

$$Pr(\mathbf{f}|\mathbf{e}) = \sum_{\mathbf{a}} Pr(\mathbf{f}, \mathbf{a}|\mathbf{e})$$

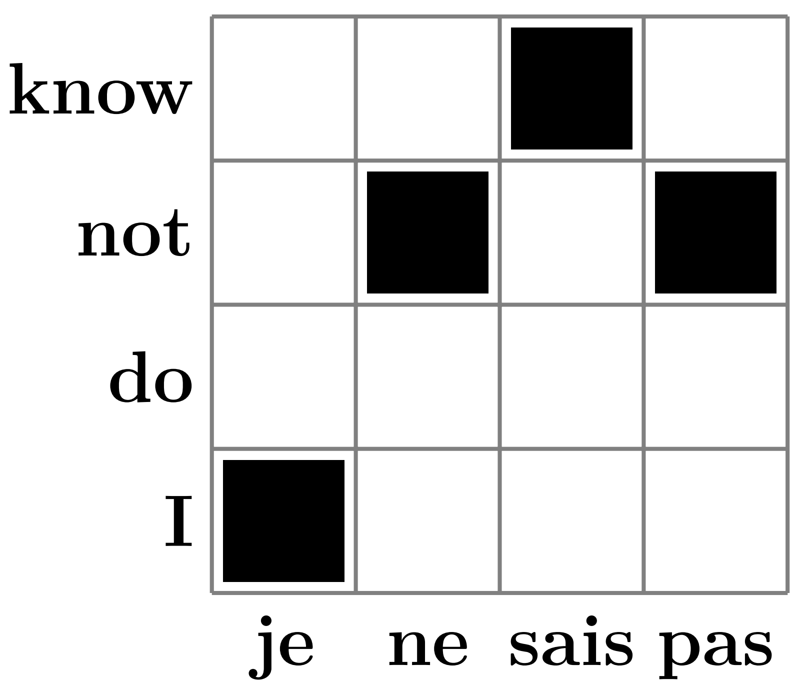


$$\mathbf{a} = \{(1 \to 1), (2 \to 3), (3 \to 4), (4 \to 3)\}$$

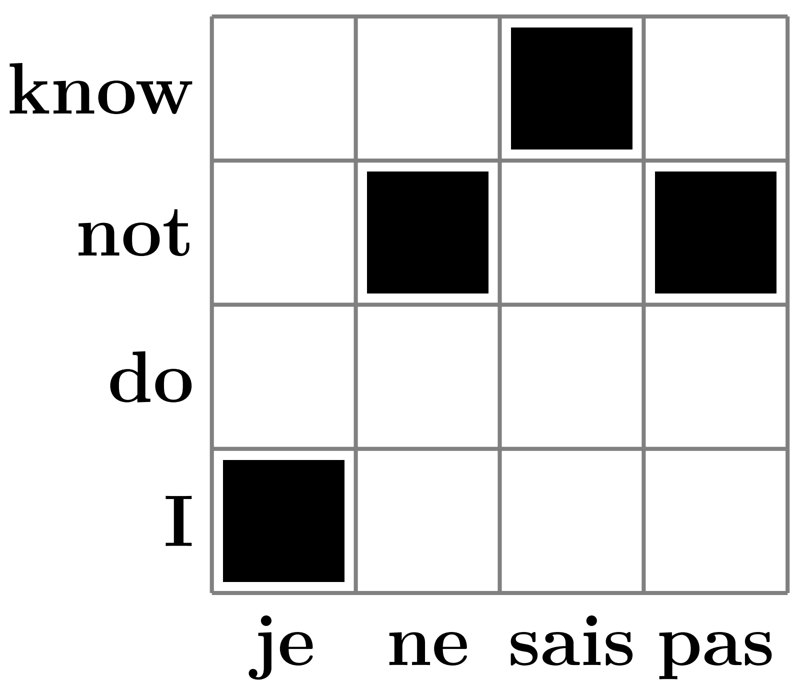
- We decompose P(f|e) into P(f,a|e)
- a: word alignment, mapping from source-to-target
- How many possible “a”? $2^{|\mathbf{e}| \times |\mathbf{f}|}$

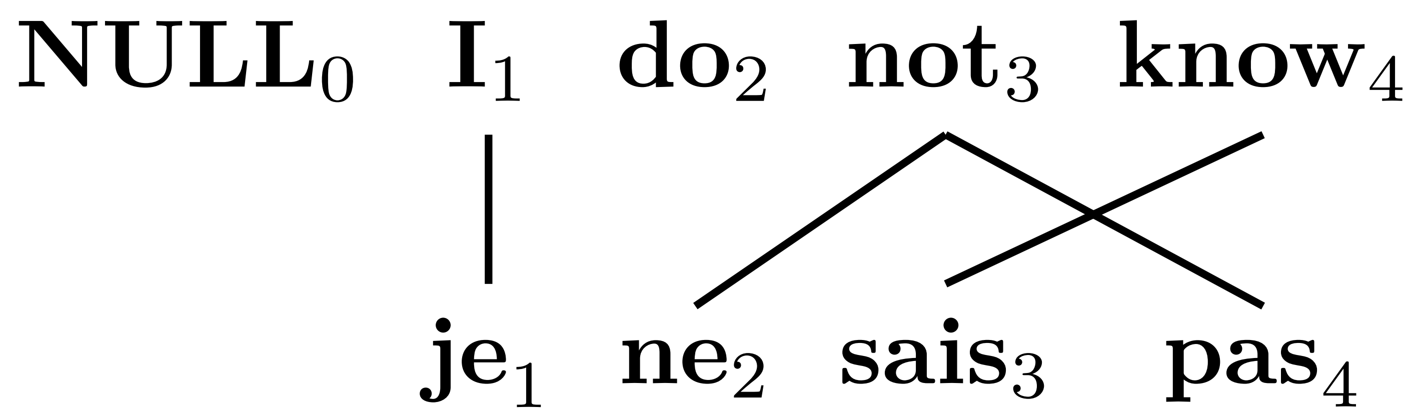
1-to-m Approximation

1-to-m Approximation



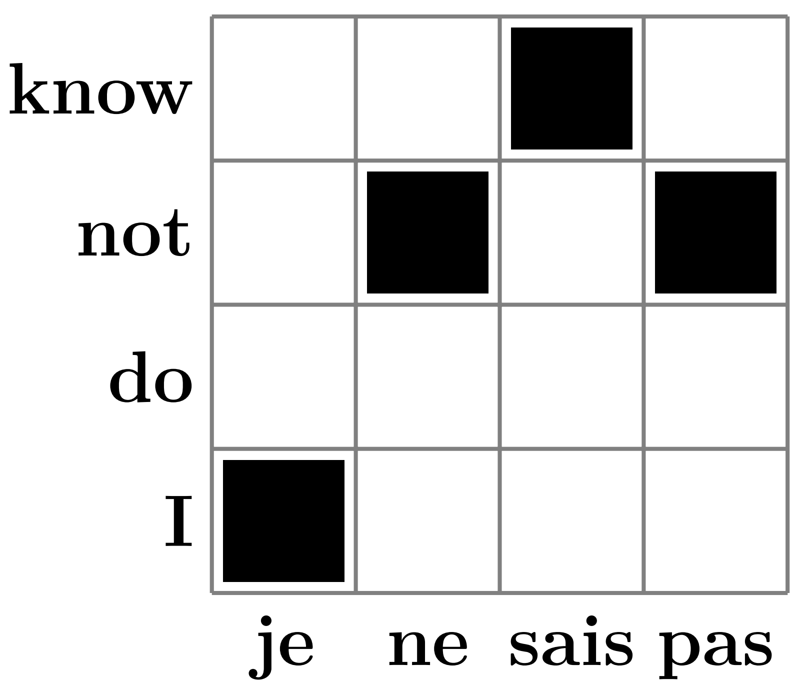
1-to-m Approximation

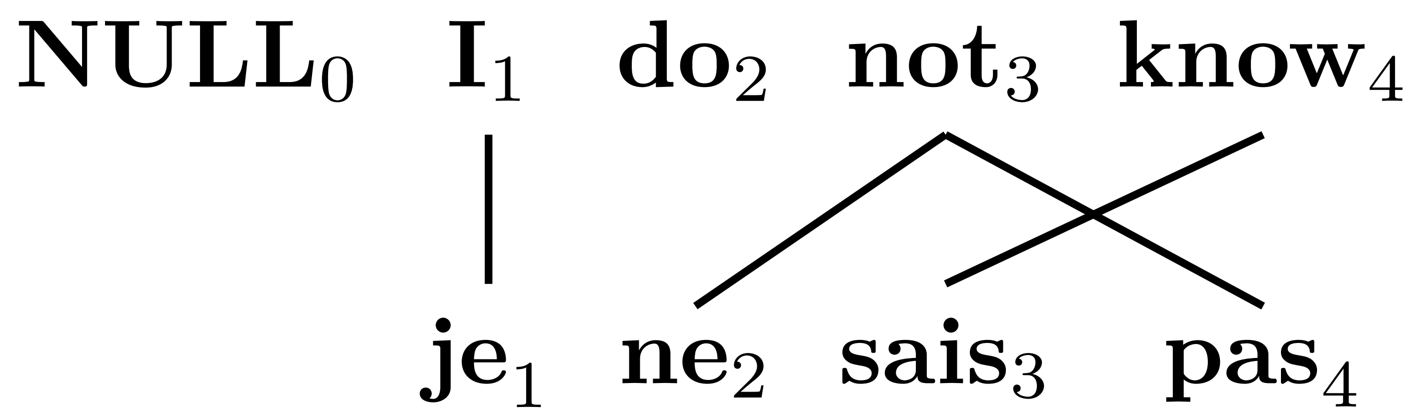




$$\mathbf{a} = \{1, 3, 4, 3\}$$

1-to-m Approximation

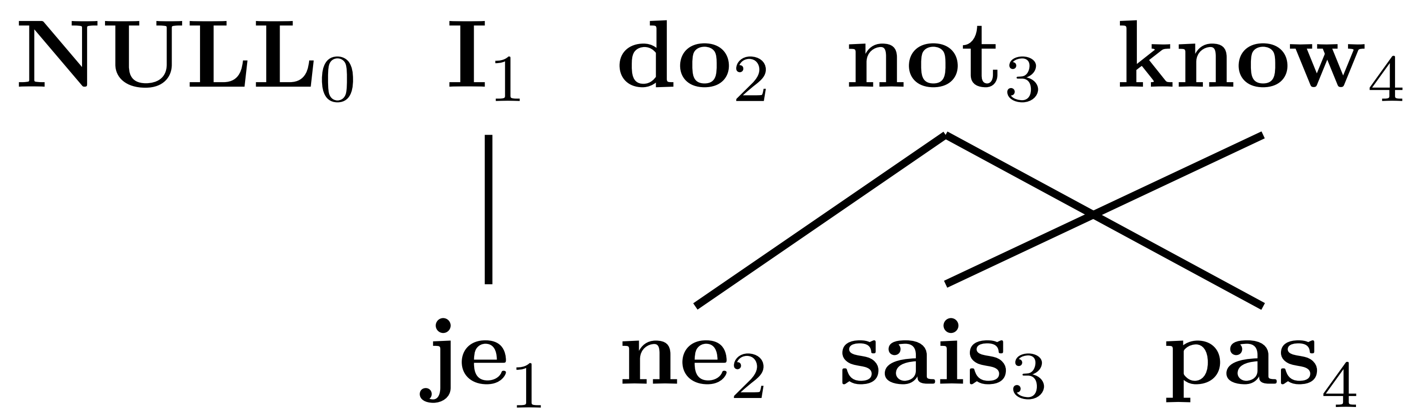




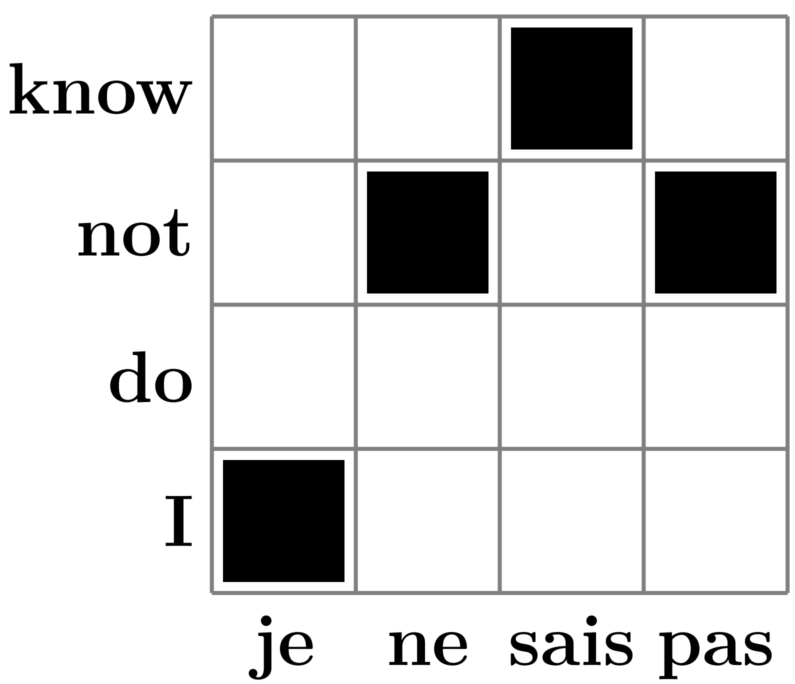
$$\mathbf{a} = \{1, 3, 4, 3\}$$

$$\begin{aligned} \mathbf{f} &= f_1^m &= f_1, f_2, f_3, \cdots \\ \mathbf{e} &= e_0^l &= e_0, e_1, e_2, e_3, \cdots \\ \mathbf{a} &= a_1^m &= a_1, a_2, a_3, \cdots \end{aligned}$$

1-to-m Approximation



$$\mathbf{a} = \{1, 3, 4, 3\}$$



$$\begin{aligned} \mathbf{f} &= f_1^m &= f_1, f_2, f_3, \cdots \\ \mathbf{e} &= e_0^l &= e_0, e_1, e_2, e_3, \cdots \\ \mathbf{a} &= a_1^m &= a_1, a_2, a_3, \cdots \end{aligned}$$

- Each word in f is aligned to one of e
- Assume NULL word in e
- How many possible “a”? $(|\mathbf{e}| + 1)^{|\mathbf{f}|}$

Decomposition: Model 1

$$
\begin{aligned}
Pr(\mathbf{f}|\mathbf{e}) &= \sum_{\mathbf{a}} Pr(\mathbf{f}, \mathbf{a}|\mathbf{e}) \\
&= \sum_{\mathbf{a}} Pr(\mathbf{f}|\mathbf{a}, \mathbf{e}) Pr(\mathbf{a}|\mathbf{e}) \\
&= Pr(m|\mathbf{e}) \sum_{\mathbf{a}} Pr(\mathbf{f}|\mathbf{a}, m, \mathbf{e}) Pr(\mathbf{a}|m, \mathbf{e}) \\
&\approx \epsilon \sum_{\mathbf{a}} \prod_{j=1}^{m} t(f_j|e_{a_j}) \frac{1}{(l+1)^m}
\end{aligned}
$$

$$
\text{s.t.} \forall e : \sum_{f} t(f|e) = 1
$$

Decomposition: Model 1

$$
\begin{aligned}
Pr(\mathbf{f}|\mathbf{e}) &= \sum_{\mathbf{a}} Pr(\mathbf{f},\mathbf{a}|\mathbf{e}) \\
&= \sum_{\mathbf{a}} Pr(\mathbf{f}|\mathbf{a},\mathbf{e}) Pr(\mathbf{a}|\mathbf{e}) \\
&= Pr(m|\mathbf{e}) \sum_{\mathbf{a}} Pr(\mathbf{f}|\mathbf{a},m,\mathbf{e}) Pr(\mathbf{a}|m,\mathbf{e}) \\
&\approx \epsilon \sum_{\mathbf{a}} \prod_{j=1}^{m} t(f_j|e_{a_j}) \frac{1}{(l+1)^m}
\end{aligned}
$$

$$\text{s.t.} \forall e : \sum_{f} t(f|e) = 1$$

- An example for a fixed "a":

$\mathbf{NULL}_0$ $\mathbf{I}_1$ $\mathbf{do}_2$ $\mathbf{not}_3$ $\mathbf{know}_4$

$\mathbf{je}_1$ $\mathbf{ne}_2$ $\mathbf{sais}_3$ $\mathbf{pas}_4$

$$
\begin{aligned}
&\epsilon \times t(\mathbf{je}_1|\mathbf{I}_1) \times t(\mathbf{ne}_2|\mathbf{not}_3) \\
&\times t(\mathbf{sais}_3|\mathbf{know}_4) \times t(\mathbf{pas}_4|\mathbf{not}_3) \\
&\times \frac{1}{5^4}
\end{aligned}
$$

Efficient Computation

$$\epsilon \sum_{\mathbf{a}} \prod_{j=1}^{m} t(f_j|e_{a_j}) \frac{1}{(l+1)^m}$$

$$= \epsilon \sum_{a_1=0}^{l} \sum_{a_2=0}^{l} \cdots \sum_{a_m=0}^{l} \prod_{j=1}^{m} t(f_j|e_{a_j}) \frac{1}{(l+1)^m}$$

$$= \epsilon \prod_{j=1}^{m} \sum_{i=0}^{l} t(f_j|e_i) \frac{1}{(l+1)^m}$$

Efficient Computation

$$\epsilon \sum_{\mathbf{a}} \prod_{j=1}^{m} t(f_j|e_{a_j}) \frac{1}{(l+1)^m}$$

$$= \epsilon \sum_{a_1=0}^{l} \sum_{a_2=0}^{l} \cdots \sum_{a_m=0}^{l} \prod_{j=1}^{m} t(f_j|e_{a_j}) \frac{1}{(l+1)^m}$$

$$= \epsilon \prod_{j=1}^{m} \sum_{i=0}^{l} t(f_j|e_i) \frac{1}{(l+1)^m}$$

$$\begin{aligned}
&\epsilon \times \{\cdots \\
&\quad + t(\textbf{je}_1|\textbf{NULL}_0) \times t(\textbf{ne}_2|\textbf{not}_3) \times \cdots \\
&\quad + t(\textbf{je}_1|\textbf{I}_1) \times t(\textbf{ne}_2|\textbf{not}_3) \times \cdots \\
&\quad + t(\textbf{je}_1|\textbf{do}_2) \times t(\textbf{ne}_2|\textbf{not}_3) \times \cdots \\
&\quad + t(\textbf{je}_1|\textbf{not}_3) \times t(\textbf{ne}_2|\textbf{not}_3) \times \cdots \\
&\quad + t(\textbf{je}_1|\textbf{know}_4) \times t(\textbf{ne}_2|\textbf{not}_3) \times \cdots \\
&\quad + \cdots \} \times \tfrac{1}{5^4}
\end{aligned}$$

Efficient Computation

$$\epsilon \sum_{\mathbf{a}} \prod_{j=1}^{m} t(f_j|e_{a_j}) \frac{1}{(l+1)^m}$$
$$= \epsilon \sum_{a_1=0}^{l} \sum_{a_2=0}^{l} \cdots \sum_{a_m=0}^{l} \prod_{j=1}^{m} t(f_j|e_{a_j}) \frac{1}{(l+1)^m}$$
$$= \epsilon \prod_{j=1}^{m} \sum_{i=0}^{l} t(f_j|e_i) \frac{1}{(l+1)^m}$$

$$\begin{aligned} \epsilon \times \{ & \cdots \\ & + t(\mathbf{je}_1|\mathbf{NULL}_0) \times t(\mathbf{ne}_2|\mathbf{not}_3) \times \cdots \\ & + t(\mathbf{je}_1|\mathbf{I}_1) \times t(\mathbf{ne}_2|\mathbf{not}_3) \times \cdots \\ & + t(\mathbf{je}_1|\mathbf{do}_2) \times t(\mathbf{ne}_2|\mathbf{not}_3) \times \cdots \\ & + t(\mathbf{je}_1|\mathbf{not}_3) \times t(\mathbf{ne}_2|\mathbf{not}_3) \times \cdots \\ & + t(\mathbf{je}_1|\mathbf{know}_4) \times t(\mathbf{ne}_2|\mathbf{not}_3) \times \cdots \\ & + \cdots \} \times \tfrac{1}{5^4} \end{aligned}$$

$$\epsilon \times \{ \left\{ \begin{array}{l} t(\mathbf{je}_1|\mathbf{NULL}_0) \\ +t(\mathbf{je}_1|\mathbf{I}_1) \\ +t(\mathbf{je}_1|\mathbf{do}_2) \\ +t(\mathbf{je}_1|\mathbf{not}_3) \\ +t(\mathbf{je}_1|\mathbf{know}_4) \end{array} \right\} \times \left\{ \begin{array}{l} t(\mathbf{ne}_2|\mathbf{NULL}_0) \\ +t(\mathbf{ne}_2|\mathbf{I}_1) \\ +t(\mathbf{ne}_2|\mathbf{do}_2) \\ +t(\mathbf{ne}_2|\mathbf{not}_3) \\ +t(\mathbf{ne}_2|\mathbf{know}_4) \end{array} \right\} \times \cdots \times \} \tfrac{1}{5^4}$$

Estimation: Model 1

- Given bilingual data, a set of f and e: $\mathcal{D} = \langle \mathcal{F}, \mathcal{E} \rangle$
- Likelihood of data: $\prod_{\langle \mathbf{f}, \mathbf{e} \rangle \in \mathcal{D}} Pr(\mathbf{f}|\mathbf{e})$
- Learn parameters Θ that maximize the log-likelihood of data: $\hat{\theta} = \operatorname*{argmax}_{\theta} \sum_{\langle \mathbf{f}, \mathbf{e} \rangle \in \mathcal{D}} \log P_{\theta}(\mathbf{f}|\mathbf{e})$

  - For Model 1, Θ corresponds to t(f | e)

Objectives: Model 1

$$\begin{aligned}
\sum_{\langle \mathbf{f},\mathbf{e}\rangle} \log P_\theta(\mathbf{f}|\mathbf{e}) &= \sum_{\langle \mathbf{f},\mathbf{e}\rangle} \log \epsilon \prod_{j=1}^{m} \sum_{i=0}^{l} t(f_j|e_i) \frac{1}{(l+1)^m} \\
&= \text{constant} + \sum_{\langle \mathbf{f},\mathbf{e}\rangle} \log \prod_{j=1}^{m} \sum_{i=0}^{l} t(f_j|e_i) \\
&= \text{constant} + \sum_{\langle \mathbf{f},\mathbf{e}\rangle} \sum_{j=1}^{m} \log \sum_{i=0}^{l} t(f_j|e_i)
\end{aligned}$$

Objectives: Model 1

$$\begin{aligned}
\sum_{\langle \mathbf{f},\mathbf{e}\rangle} \log P_\theta(\mathbf{f}|\mathbf{e}) &= \sum_{\langle \mathbf{f},\mathbf{e}\rangle} \log \epsilon \prod_{j=1}^{m} \sum_{i=0}^{l} t(f_j|e_i) \frac{1}{(l+1)^m} \\
&= \text{constant} + \sum_{\langle \mathbf{f},\mathbf{e}\rangle} \log \prod_{j=1}^{m} \sum_{i=0}^{l} t(f_j|e_i) \\
&= \text{constant} + \sum_{\langle \mathbf{f},\mathbf{e}\rangle} \sum_{j=1}^{m} \log \sum_{i=0}^{l} t(f_j|e_i)
\end{aligned}$$

- Maximize: $$L(\theta) = \sum_{\langle \mathbf{f},\mathbf{e}\rangle} \sum_{j=1}^{m} \log \sum_{i=0}^{l} t(f_j|e_i)$$
- Constraints: $$\forall e : \sum_{f} t(f|e) = 1$$

Iterative Learning: Model 1

Iterative Learning: Model 1

- We will build an iterative procedure to maximize $L(\Theta)$: choose Θ' which is better than Θ

Iterative Learning: Model 1

- We will build an iterative procedure to maximize L(Θ): choose Θ' which is better than Θ
- Introduce an auxiliary variable: probability of aligning f_j and e_i given f,e

$$q_{i,j}(\theta; \mathbf{f}, \mathbf{e}) = \frac{t_\theta(f_j|e_i)}{\sum_{i'=0}^{l} t_\theta(f_j|e_{i'})}$$

Iterative Learning: Model 1

- We will build an iterative procedure to maximize L(Θ): choose Θ' which is better than Θ
- Introduce an auxiliary variable: probability of aligning f_j and e_i given f,e

$$q_{i,j}(\theta; \mathbf{f}, \mathbf{e}) = \frac{t_\theta(f_j|e_i)}{\sum_{i'=0}^{l} t_\theta(f_j|e_{i'})}$$

- Remark:

$$P_\theta(\mathbf{a}|\mathbf{f}, \mathbf{e}) = \frac{P_\theta(\mathbf{f}, \mathbf{a}|\mathbf{e})}{P_\theta(\mathbf{f}|\mathbf{e})} = \prod_{j=1}^{m} q_{i,j}(\theta; \mathbf{f}, \mathbf{e})$$

Iterative Learning: Model 1

- We will build an iterative procedure to maximize L(Θ): choose Θ' which is better than Θ
- Introduce an auxiliary variable: probability of aligning f_j and e_i given f,e

$$q_{i,j}(\theta; \mathbf{f}, \mathbf{e}) = \frac{t_\theta(f_j|e_i)}{\sum_{i'=0}^{l} t_\theta(f_j|e_{i'})}$$

- Remark:

$$P_\theta(\mathbf{a}|\mathbf{f}, \mathbf{e}) = \frac{P_\theta(\mathbf{f}, \mathbf{a}|\mathbf{e})}{P_\theta(\mathbf{f}|\mathbf{e})} = \prod_{j=1}^{m} q_{i,j}(\theta; \mathbf{f}, \mathbf{e})$$

- Use Jensen's inequality:

$$\log \sum_z q(z) \frac{p(x,z)}{q(z)} \geq \sum_z q(z) \log \frac{p(x,z)}{q(z)}$$

Lower Bound: Model 1

$$
\begin{aligned}
L(\theta^T) &= \sum_{\langle \mathbf{f},\mathbf{e} \rangle} \sum_{j=1}^{m} \log \sum_{i=0}^{l} t_{\theta^T}(f_j|e_i) \\
&= \sum_{\langle \mathbf{f},\mathbf{e} \rangle} \sum_{j=1}^{m} \log \sum_{i=0}^{l} q_{i,j}(\theta^{T-1}) \frac{t_{\theta^T}(f_j|e_i)}{q_{i,j}(\theta^{T-1})} \\
&\geq \sum_{\langle \mathbf{f},\mathbf{e} \rangle} \sum_{j=1}^{m} q_{i,j}(\theta^{T-1}) \log \sum_{i=0}^{l} \frac{t_{\theta^T}(f_j|e_i)}{q_{i,j}(\theta^{T-1})} \\
&= \sum_{\langle \mathbf{f},\mathbf{e} \rangle} \sum_{j=1}^{m} q_{i,j}(\theta^{T-1}) \log \sum_{i=0}^{l} t_{\theta^T}(f_j|e_i) + \text{constant}
\end{aligned}
$$

Lower Bound: Model 1

$$
\begin{aligned}
L(\theta^T) &= \sum_{\langle \mathbf{f}, \mathbf{e} \rangle} \sum_{j=1}^{m} \log \sum_{i=0}^{l} t_{\theta^T}(f_j | e_i) \\
&= \sum_{\langle \mathbf{f}, \mathbf{e} \rangle} \sum_{j=1}^{m} \log \sum_{i=0}^{l} q_{i,j}(\theta^{T-1}) \frac{t_{\theta^T}(f_j | e_i)}{q_{i,j}(\theta^{T-1})} \\
&\geq \sum_{\langle \mathbf{f}, \mathbf{e} \rangle} \sum_{j=1}^{m} q_{i,j}(\theta^{T-1}) \log \sum_{i=0}^{l} \frac{t_{\theta^T}(f_j | e_i)}{q_{i,j}(\theta^{T-1})} \\
&= \sum_{\langle \mathbf{f}, \mathbf{e} \rangle} \sum_{j=1}^{m} q_{i,j}(\theta^{T-1}) \log \sum_{i=0}^{l} t_{\theta^T}(f_j | e_i) + \text{constant}
\end{aligned}
$$

Jensen's inequality

Lower Bound: Model 1

$$
\begin{aligned}
L(\theta^T) &= \sum_{\langle \mathbf{f}, \mathbf{e} \rangle} \sum_{j=1}^{m} \log \sum_{i=0}^{l} t_{\theta^T}(f_j | e_i) \\
&= \sum_{\langle \mathbf{f}, \mathbf{e} \rangle} \sum_{j=1}^{m} \log \sum_{i=0}^{l} q_{i,j}(\theta^{T-1}) \frac{t_{\theta^T}(f_j | e_i)}{q_{i,j}(\theta^{T-1})} \\
&\geq \sum_{\langle \mathbf{f}, \mathbf{e} \rangle} \sum_{j=1}^{m} q_{i,j}(\theta^{T-1}) \log \sum_{i=0}^{l} \frac{t_{\theta^T}(f_j | e_i)}{q_{i,j}(\theta^{T-1})} \\
&= \boxed{\sum_{\langle \mathbf{f}, \mathbf{e} \rangle} \sum_{j=1}^{m} q_{i,j}(\theta^{T-1}) \log \sum_{i=0}^{l} t_{\theta^T}(f_j | e_i)} + \text{constant}
\end{aligned}
$$

Jensen's inequality

lower bound

Maximize: Model 1

$$\hat{\theta^T} = \operatorname*{argmax}_{\theta^T} \sum_{\langle \mathbf{f}, \mathbf{e} \rangle} \sum_{j=1}^{m} q_{i,j}(\theta^{T-1}) \log \sum_{i=0}^{l} t_{\theta^T}(f_j | e_i)$$

$$\text{s.t.} \forall e : \sum_{f} t_{\theta}(f|e) = 1$$

- Objective is concave: we can compute global maximum
  - But, potentially many global maximum (Why?)
  - Brown et al. (1993) says "strictly concave" (Toutanova and Galley, 2011)
- Standard maximization technique: Introduce Lagrangian + take its partial differentiation + maximize

Maximize: Model 1

Maximize: Model 1

- Lagrangian

$$
\begin{aligned}
h(\theta^T) &= \sum_{\langle \mathbf{f},\mathbf{e}\rangle} \sum_{j=1}^{m} q_{i,j}(\theta^{T-1}) \log \sum_{i=0}^{l} t_{\theta^T}(f_j|e_i) \\
&\quad - \sum_{e} \alpha_e \left(\sum_{f} t_{\theta^T}(f|e) - 1 \right)
\end{aligned}
$$

Maximize: Model 1

- Lagrangian

$$h(\theta^T) = \sum_{\langle \mathbf{f},\mathbf{e}\rangle} \sum_{j=1}^{m} q_{i,j}(\theta^{T-1}) \log \sum_{i=0}^{l} t_{\theta^T}(f_j|e_i)$$

$$-\sum_{e} \alpha_e \left(\sum_{f} t_{\theta^T}(f|e) - 1 \right)$$

- Partial derivation

$$\frac{\partial h(\theta^T)}{\partial t_{\theta^T(f|e)}} = \sum_{\langle \mathbf{f},\mathbf{e}\rangle} \sum_{j=1}^{m} \sum_{i=0}^{l} q_{i,j}(\theta^{T-1}) t_{\theta^T}(f_j|e_i)^{-1} \delta(f, f_j) \delta(e, e_i) - \alpha_e$$

Maximize: Model 1

- Lagrangian

$$h(\theta^T) = \sum_{\langle \mathbf{f}, \mathbf{e} \rangle} \sum_{j=1}^{m} q_{i,j}(\theta^{T-1}) \log \sum_{i=0}^{l} t_{\theta^T}(f_j | e_i)$$

$$- \sum_{e} \alpha_e \left(\sum_{f} t_{\theta^T}(f|e) - 1 \right)$$

- Partial derivation

$$\frac{\partial h(\theta^T)}{\partial t_{\theta^T(f|e)}} = \sum_{\langle \mathbf{f}, \mathbf{e} \rangle} \sum_{j=1}^{m} \sum_{i=0}^{l} q_{i,j}(\theta^{T-1}) t_{\theta^T}(f_j | e_i)^{-1} \delta(f, f_j) \delta(e, e_i) - \alpha_e$$

- Maximize

$$t_{\theta^T}(f|e) = \alpha_e^{-1} \sum_{\langle \mathbf{f}, \mathbf{e} \rangle} \sum_{j=1}^{m} \sum_{i=0}^{l} q_{i,j}(\theta^{T-1}) \delta(f, f_j) \delta(e, e_i)$$

$$\delta(x, y) = \begin{cases} 1 & \text{if } x = y \\ 0 & \text{otherwise} \end{cases}$$

EM-Algorithm: Model 1

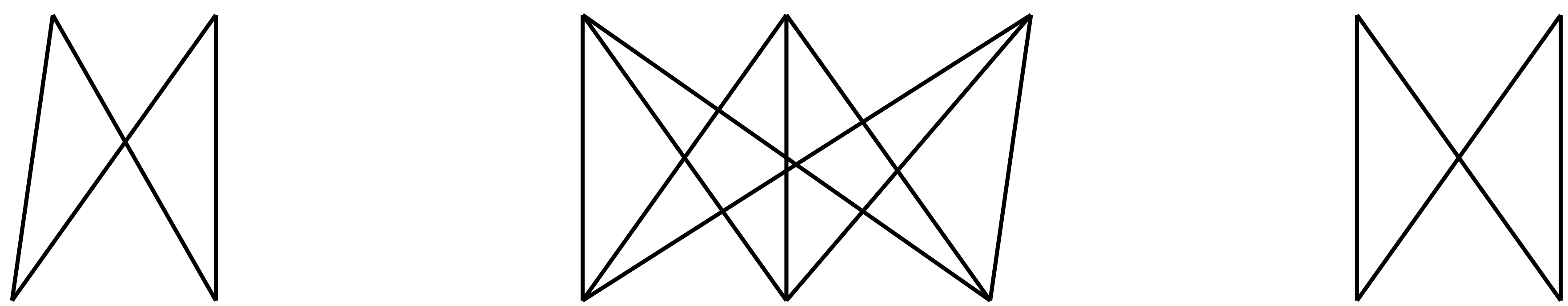
$$t_{\theta^T}(f|e) = \alpha_e^{-1} \sum_{\langle \mathbf{f}, \mathbf{e} \rangle} \sum_{j=1}^{m} \sum_{i=0}^{l} q_{i,j}(\theta^{T-1}) \delta(f, f_j) \delta(e, e_i)$$

$$\forall e : \sum_f t(f|e) = 1 \qquad q_{i,j}(\theta; \mathbf{f}, \mathbf{e}) = \frac{t_\theta(f_j|e_i)}{\sum_{i'=0}^{l} t_\theta(f_j|e_{i'})}$$

- New parameter t(f|e) in LHS is estimated from the expected counts using the old parameters
- alpha serves as a normalizer
- Starting from Θ^0, compute Θ^T from Θ^{T-1}
  - Compute expected counts (E-step)
  - Perform maximization (M-step)

An Example

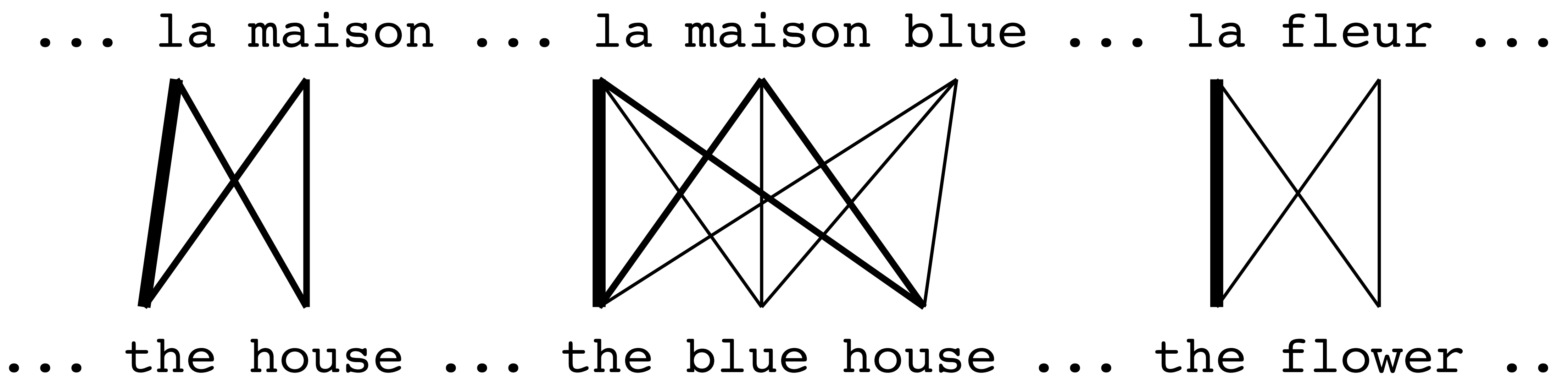
... la maison ... la maison blue ... la fleur ...



... the house ... the blue house ... the flower ...

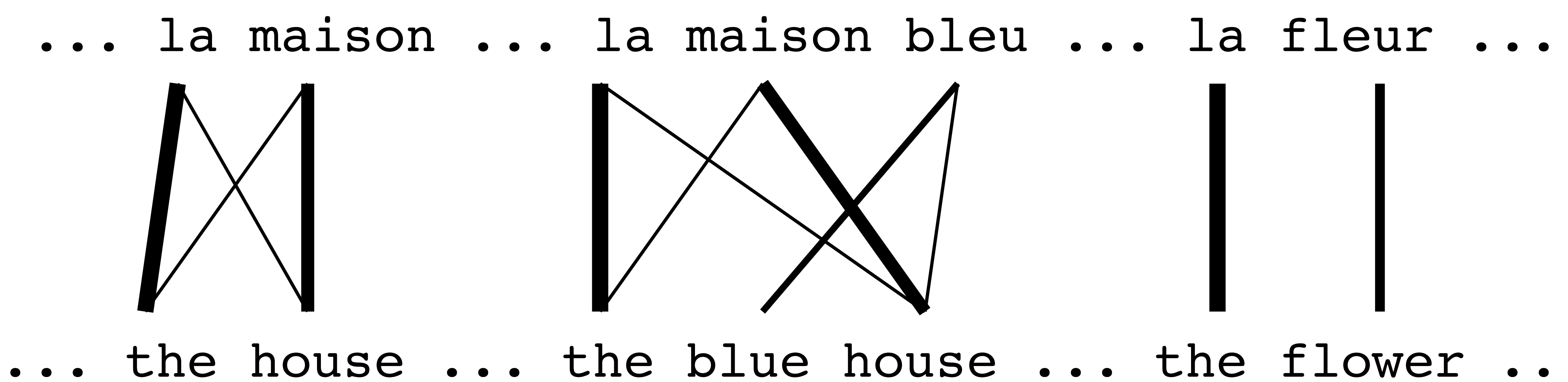
- Initial steps: all alignments equal likely
- An example from Chapter 4 of (Koehn, 2009)

An Example



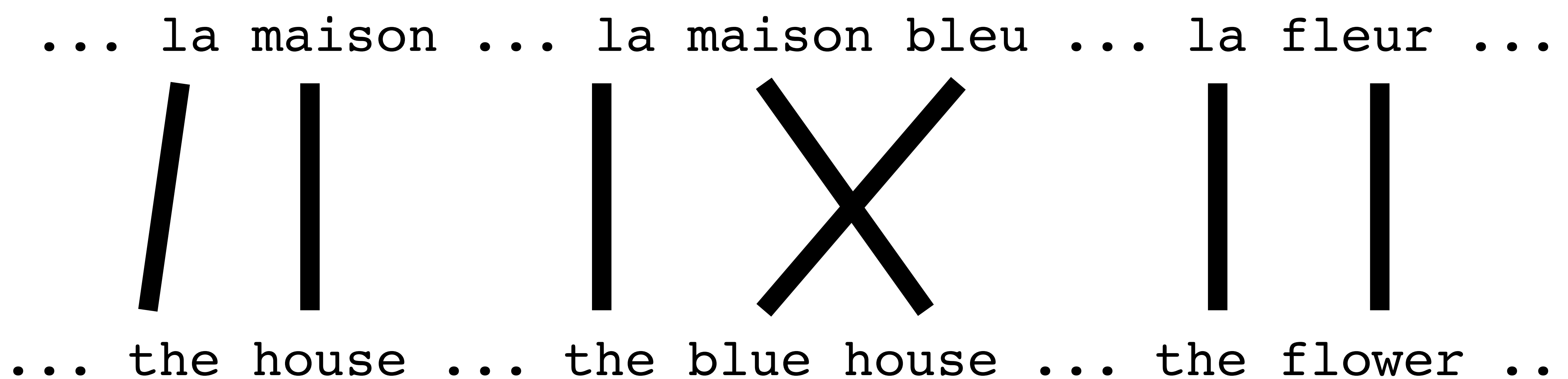
- After one iteration, alignments between "le" and "the" are more likely

An Example



- After another iteration, “fleur” and “flower” are more likely aligned

An Example

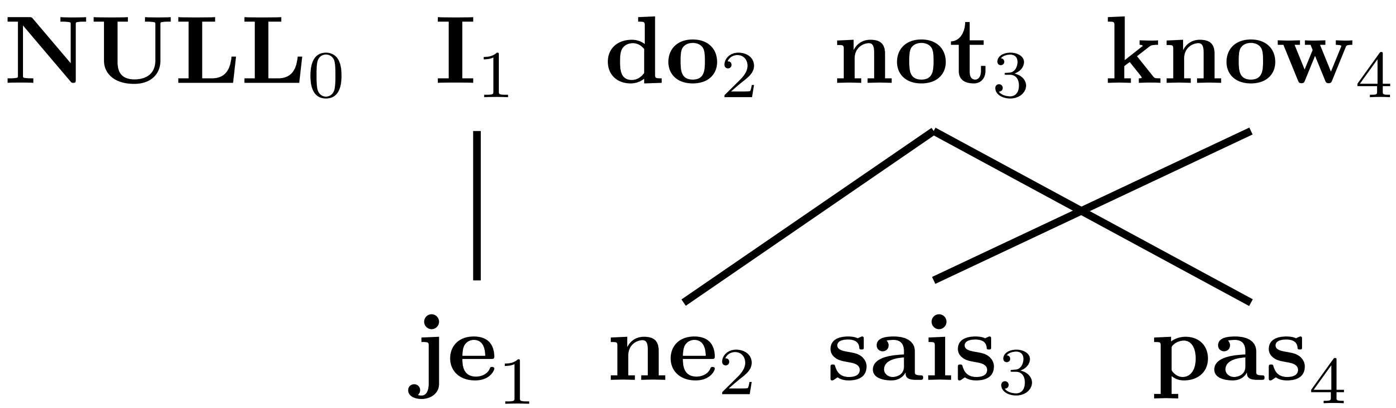


- Convergence

Interpretation: Model 1

Interpretation: Model 1

- If "a" is given, we collect counts from alignment



$$t(\mathbf{je}|\mathbf{I}) = \frac{\mathrm{count}(\mathbf{je}, \mathbf{I})}{\sum_f \mathrm{count}(f, \mathbf{I})}$$

Interpretation: Model 1

- If "a" is given, we collect counts from alignment

$$\mathbf{NULL}_0 \quad \mathbf{I}_1 \quad \mathbf{do}_2 \quad \mathbf{not}_3 \quad \mathbf{know}_4$$

$$\mathbf{je}_1 \quad \mathbf{ne}_2 \quad \mathbf{sais}_3 \quad \mathbf{pas}_4$$

$$t(\mathbf{je}|\mathbf{I}) = \frac{\mathrm{count}(\mathbf{je}, \mathbf{I})}{\sum_f \mathrm{count}(f, \mathbf{I})}$$

- EM-Algorithm: collect "fractional counts" from t(f|e)

$$\mathbf{NULL}_0 \quad \mathbf{I}_1 \quad \mathbf{do}_2 \quad \mathbf{not}_3 \quad \mathbf{know}_4$$

$$\mathbf{je}_1 \quad \mathbf{ne}_2 \quad \mathbf{sais}_3 \quad \mathbf{pas}_4$$

$$t(\mathbf{je}|\mathbf{I}) = \frac{\mathrm{count}(\mathbf{je}, \mathbf{I}; \theta)}{\sum_f \mathrm{count}(f, \mathbf{I}; \theta)}$$

Pseudo code: Model 1

Input: set of sentence pairs $(\mathbf{f}, \mathbf{e})$

Output: translation prob. $t(f|e)$

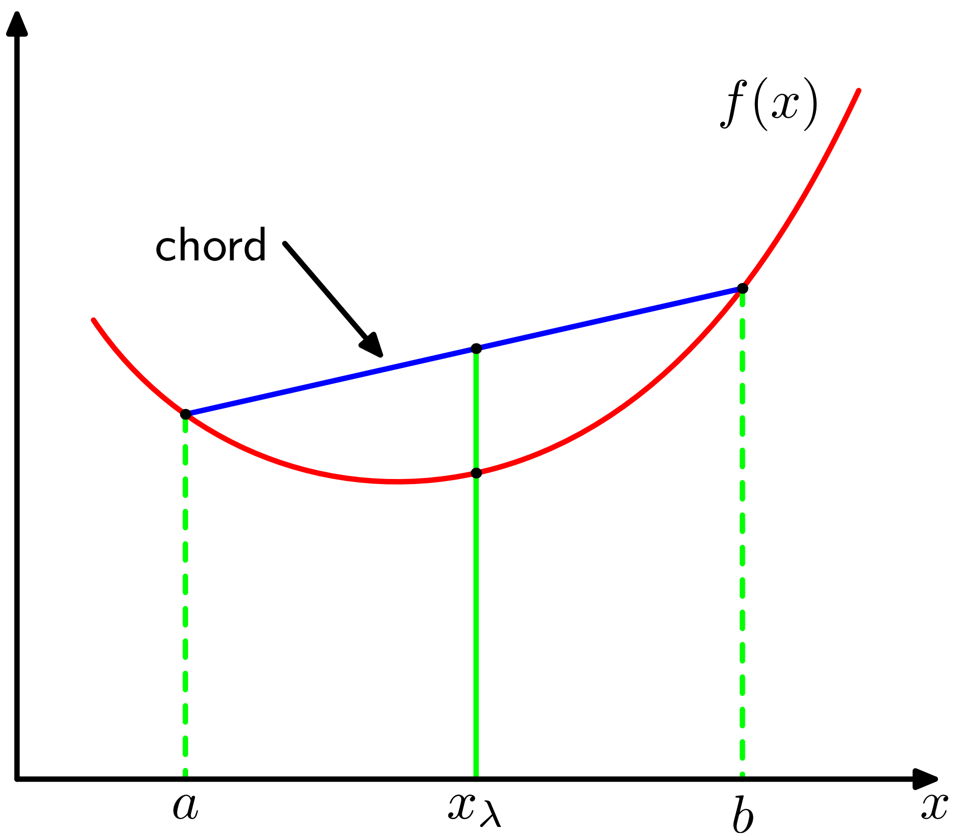
```
initialize t(f|e) uniformly
while not converged do
    // initialize
    count(f|e) = 0 for all f, e
    total(e) = 0 for all e
    for all sentence pairs (f,e) do
        // compute normalization
        for all words f in f do
            s-total(f) = 0
            for all words e in e do
    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

- Adapted from Chapter 4 of (Koehn, 2009)

Summary: Model 1

- Modeling: Model 1 parameter Θ consists of lexical translation parameters of $t(f|e)$
- Learning: EM-algorithm to learn Θ given f, e
- Remaining questions:
  - Given Θ, f, e, what is the most likely "a"
    - Viterbi alignment: replace summation with "max"
  - Given Θ, f, what is the most likely "e, a"
    - decoding problem: we will cover this later

Some notes on Model 1



$$L(\theta) = \sum_{\langle \mathbf{f}, \mathbf{e} \rangle} \sum_{j=1}^{m} \log \sum_{i=0}^{l} t(f_j | e_i)$$

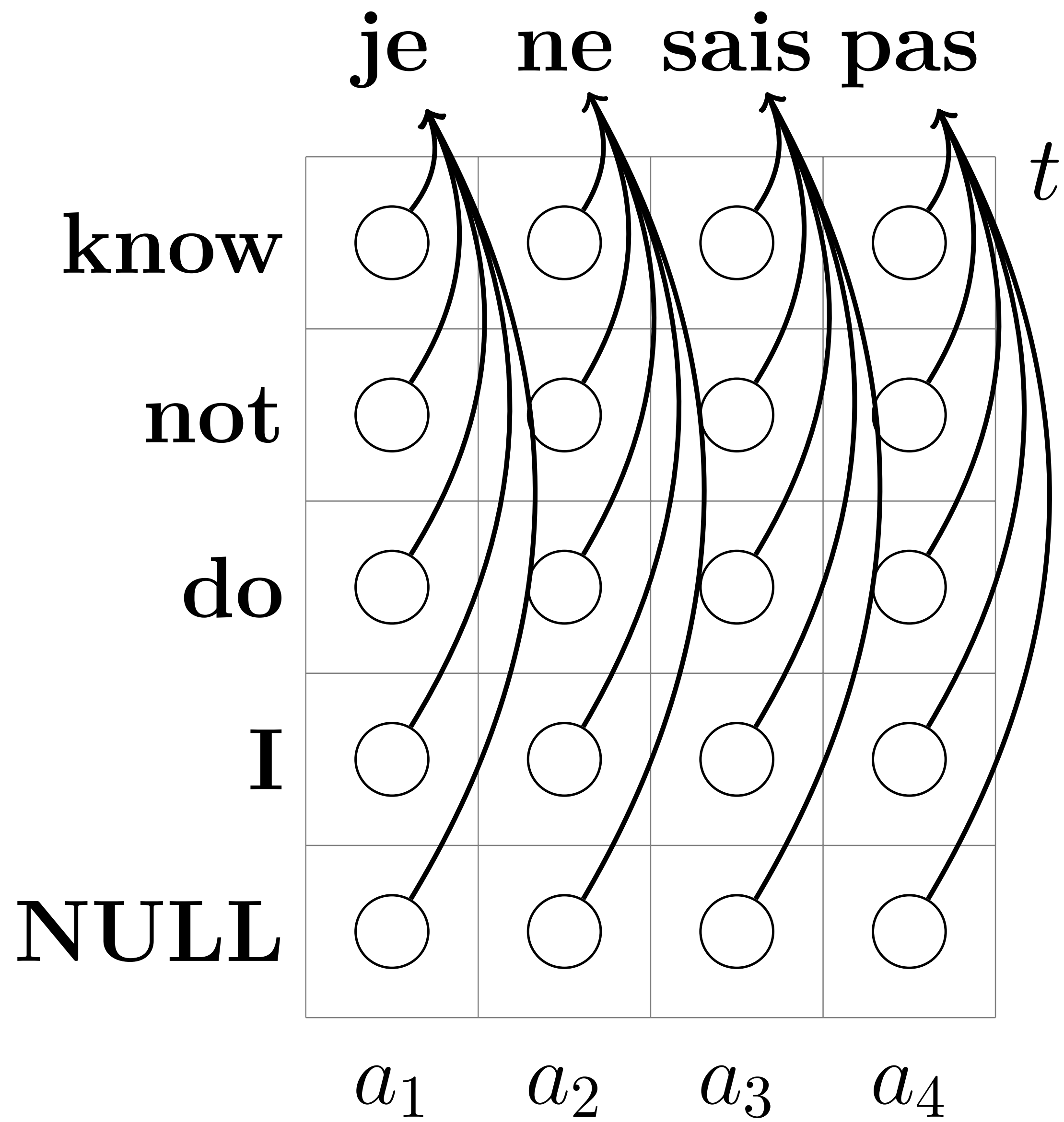
$$\forall e : \sum_{f} t(f|e) = 1$$

$$f(\lambda x + (1-\lambda)y) \leq \lambda f(x) + (1-\lambda) f(y)$$

$$f(\lambda x + (1-\lambda)y) < \lambda f(x) + (1-\lambda) f(y)$$

- -log(x) is strictly convex, but -log(∑x) is convex
- Many global optimum (Toutanova and Galley, 2011)
- We can easily re-distribute ∑x among others
  - If e and e' always co-occur in a data, we cannot distinguish them

Other Models

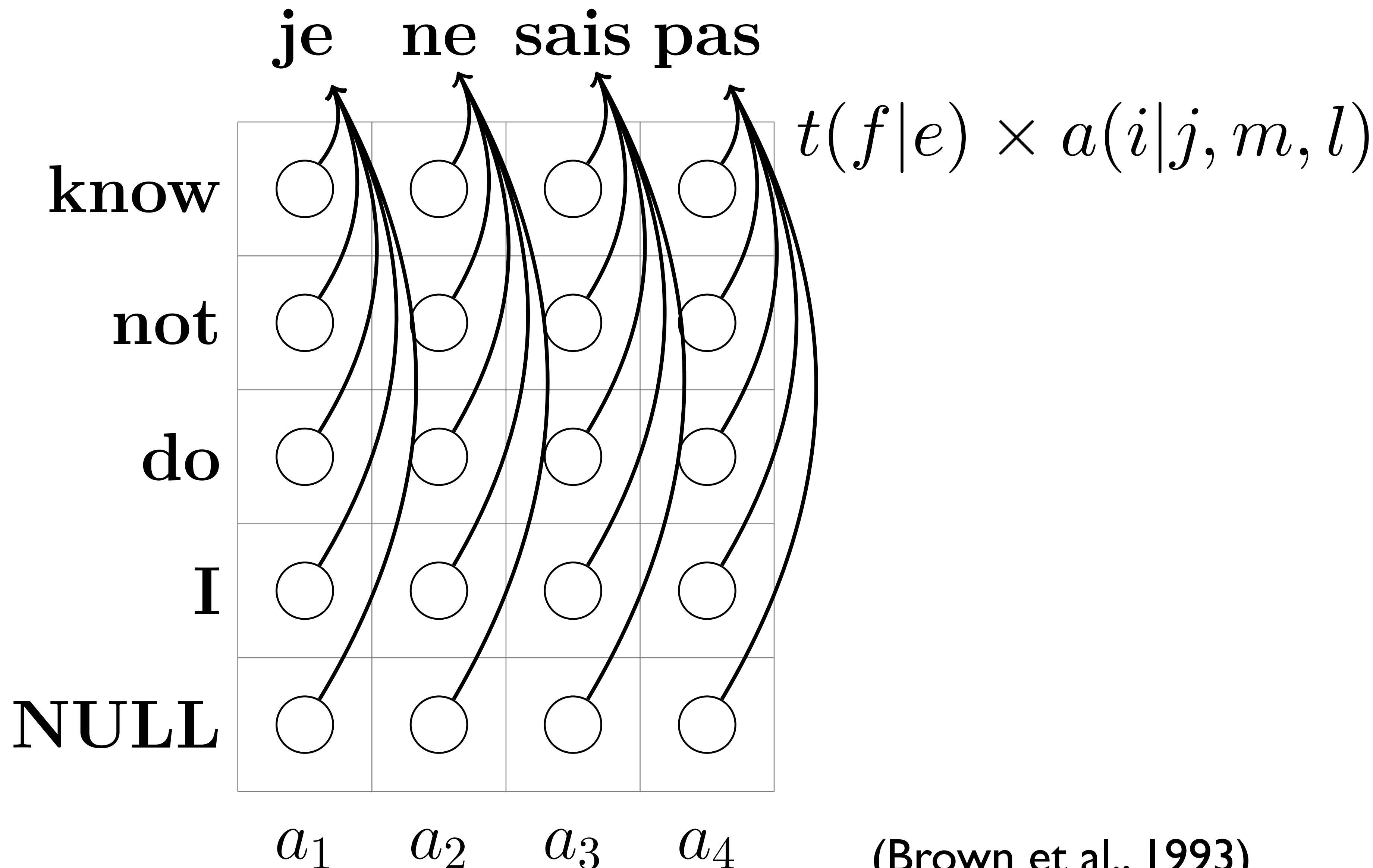


$$t(f|e) \times \frac{1}{l+1}$$

(Brown et al., 1993)

- Reminder: Generative story of Model 1
  - Each word f is generated from one of e

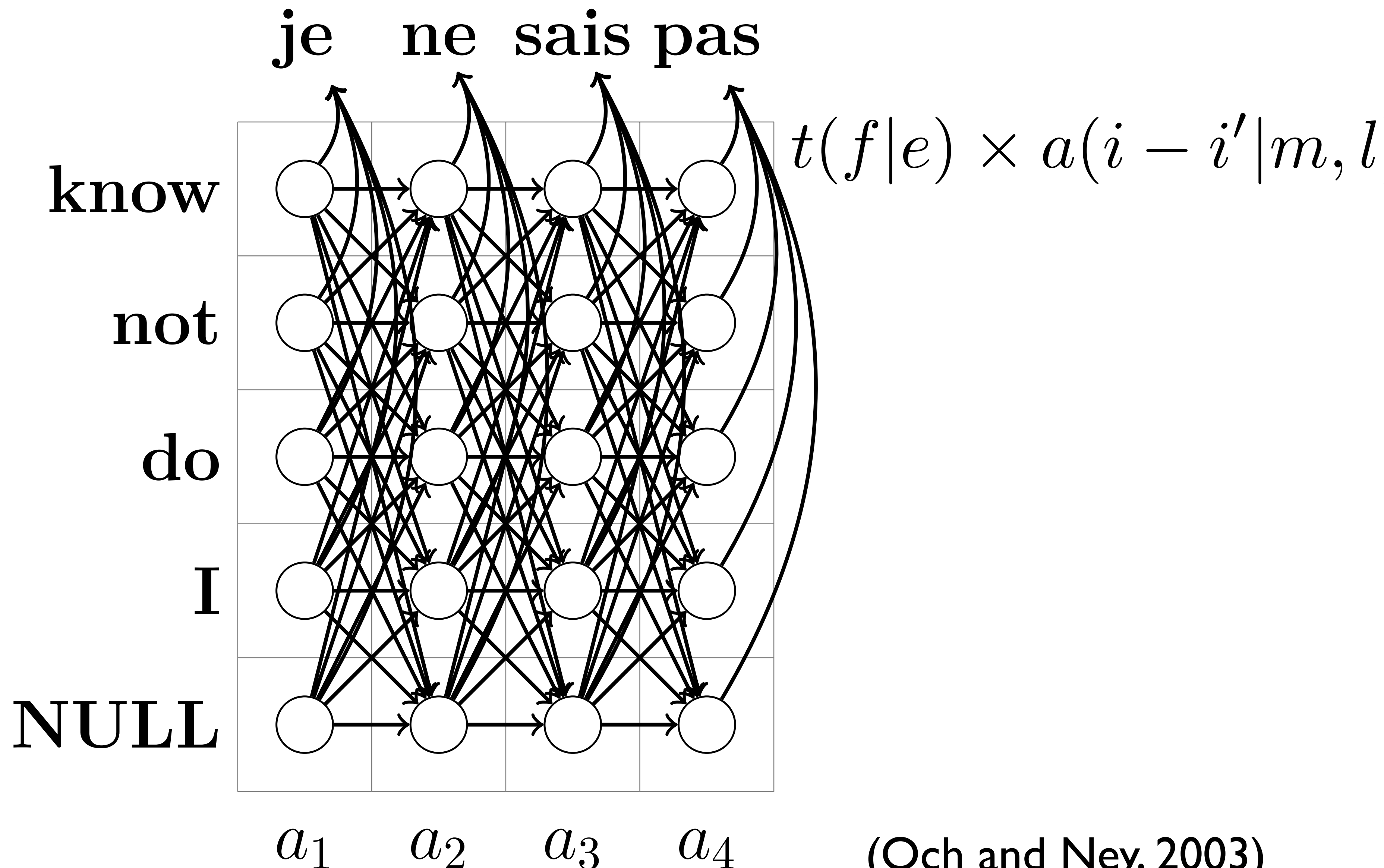
Model 2



(Brown et al., 1993)

- Like Model 1, each f is generated independently, but with alignment distribution

HMM Model



(Och and Ney, 2003)

- Each f is emitted from one of e, and alignment is conditioned on previous alignment

Model 3-5

(Brown et al., 1993)

- Completely different story from Model 1,2 or HMM
- Explicitly model one-to-many alignment via fertility
- Unlike Model 1,2, HMM, no Dynamic Programming

Model 3-5

I do not know

(Brown et al., 1993)

- Completely different story from Model 1,2 or HMM
- Explicitly model one-to-many alignment via fertility
- Unlike Model 1,2, HMM, no Dynamic Programming

Model 3-5

I do not know

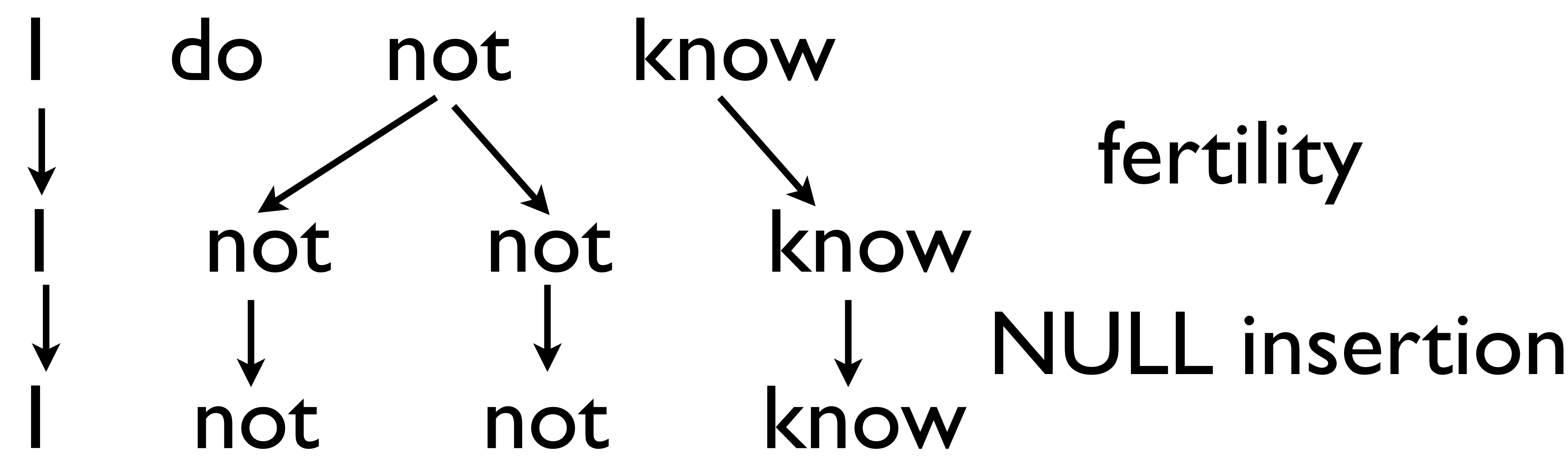
I not not know

fertility

(Brown et al., 1993)

- Completely different story from Model 1,2 or HMM
- Explicitly model one-to-many alignment via fertility
- Unlike Model 1,2, HMM, no Dynamic Programming

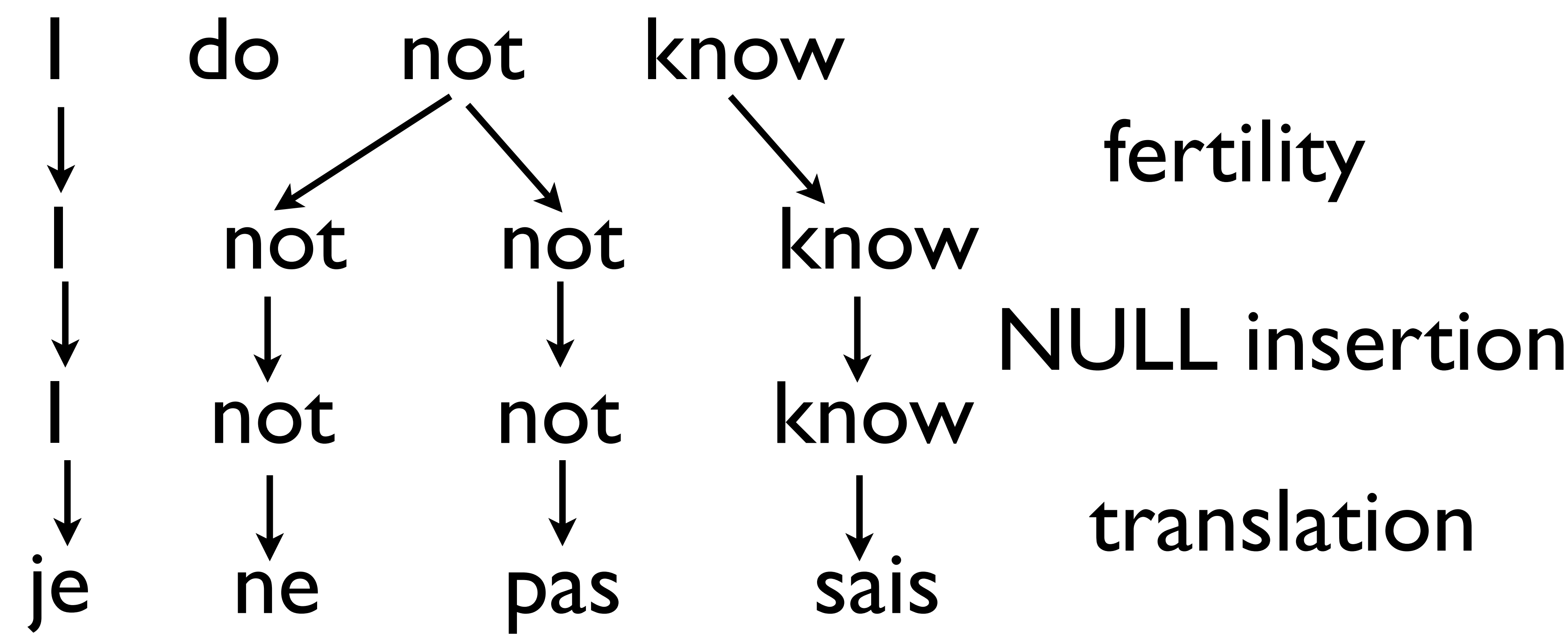
Model 3-5



(Brown et al., 1993)

- Completely different story from Model 1,2 or HMM
- Explicitly model one-to-many alignment via fertility
- Unlike Model 1,2, HMM, no Dynamic Programming

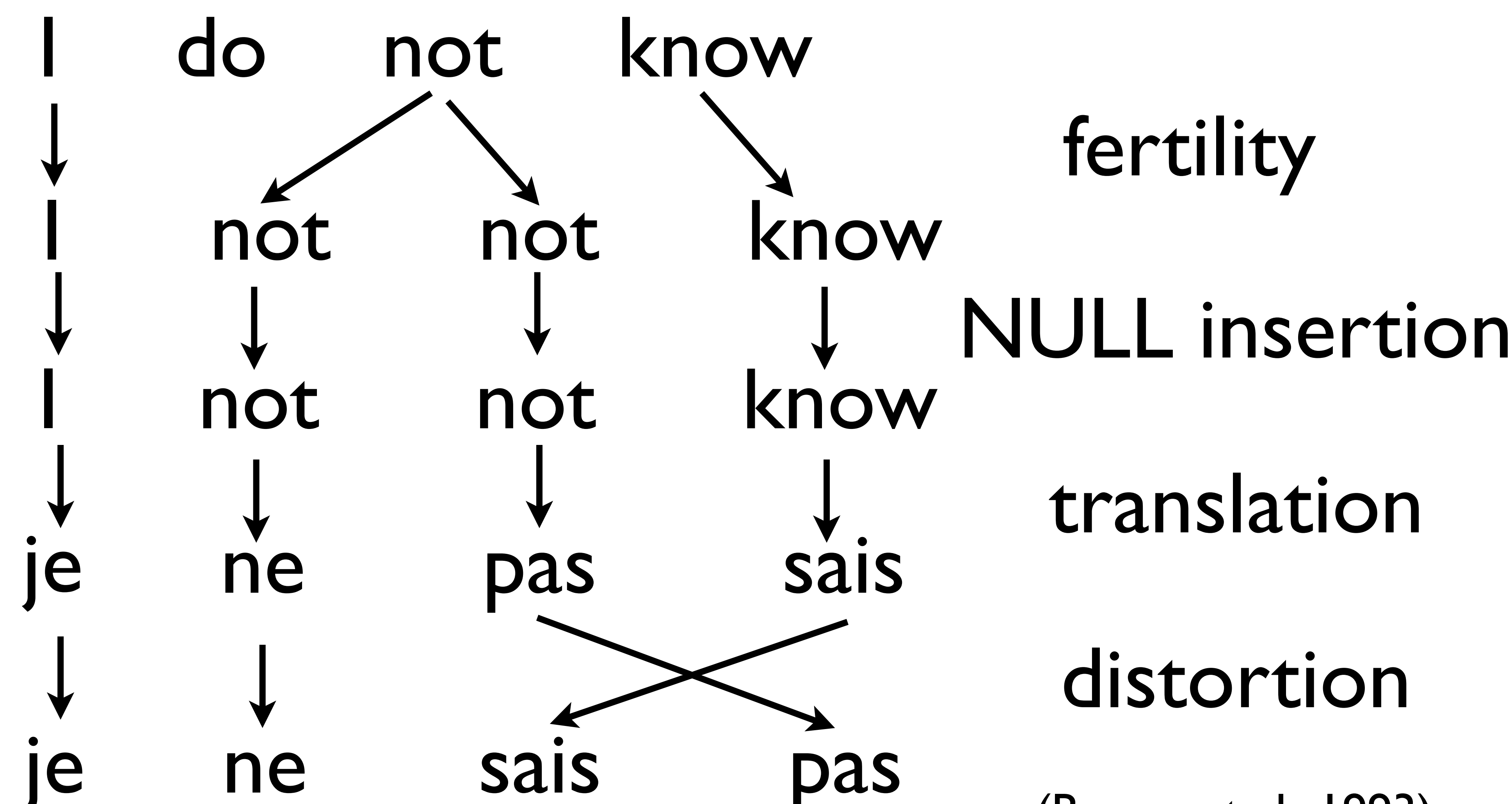
Model 3-5



(Brown et al., 1993)

- Completely different story from Model 1,2 or HMM
- Explicitly model one-to-many alignment via fertility
- Unlike Model 1,2, HMM, no Dynamic Programming

Model 3-5



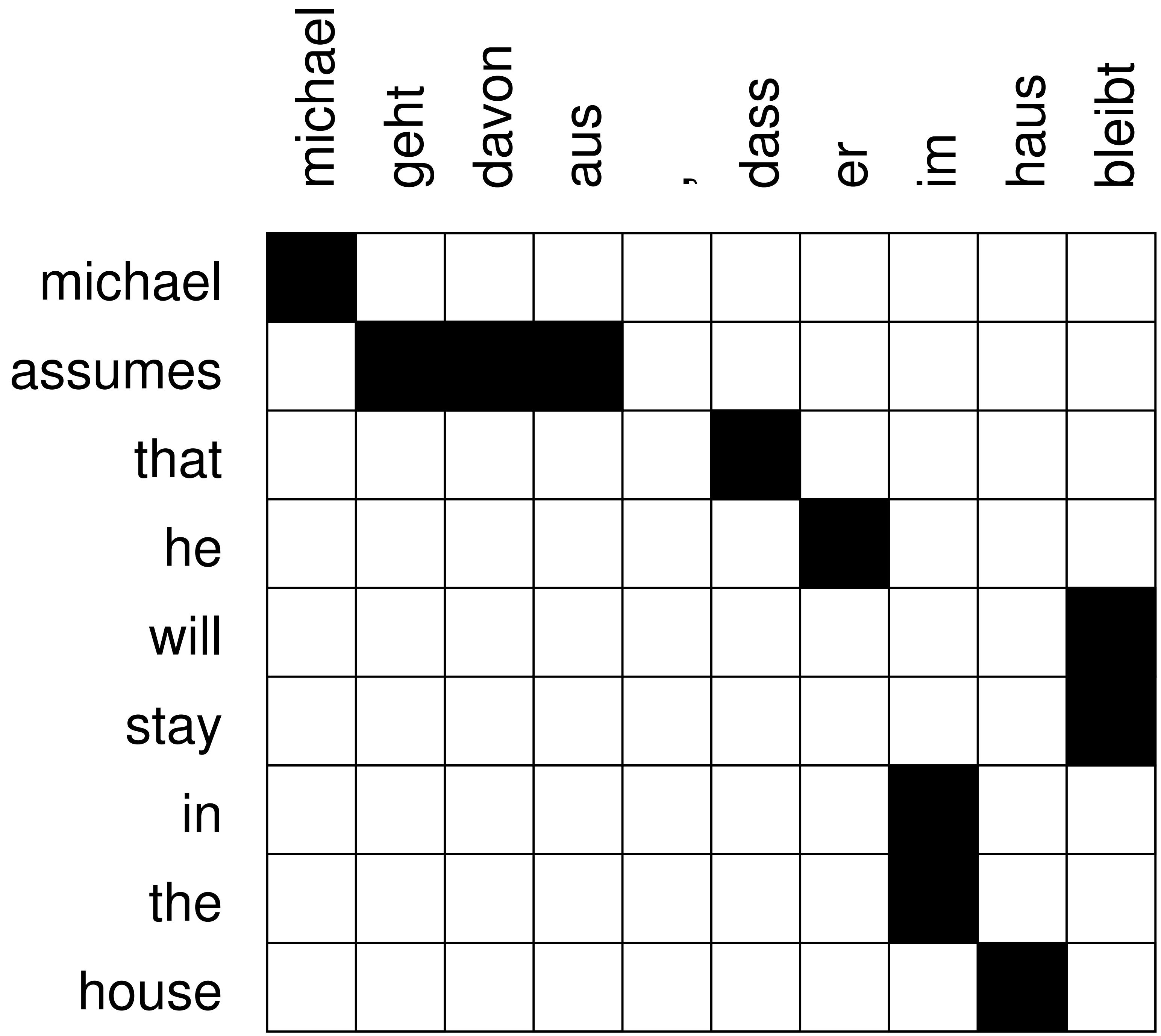
(Brown et al., 1993)

- Completely different story from Model 1,2 or HMM
- Explicitly model one-to-many alignment via fertility
- Unlike Model 1,2, HMM, no Dynamic Programming

Conclusion

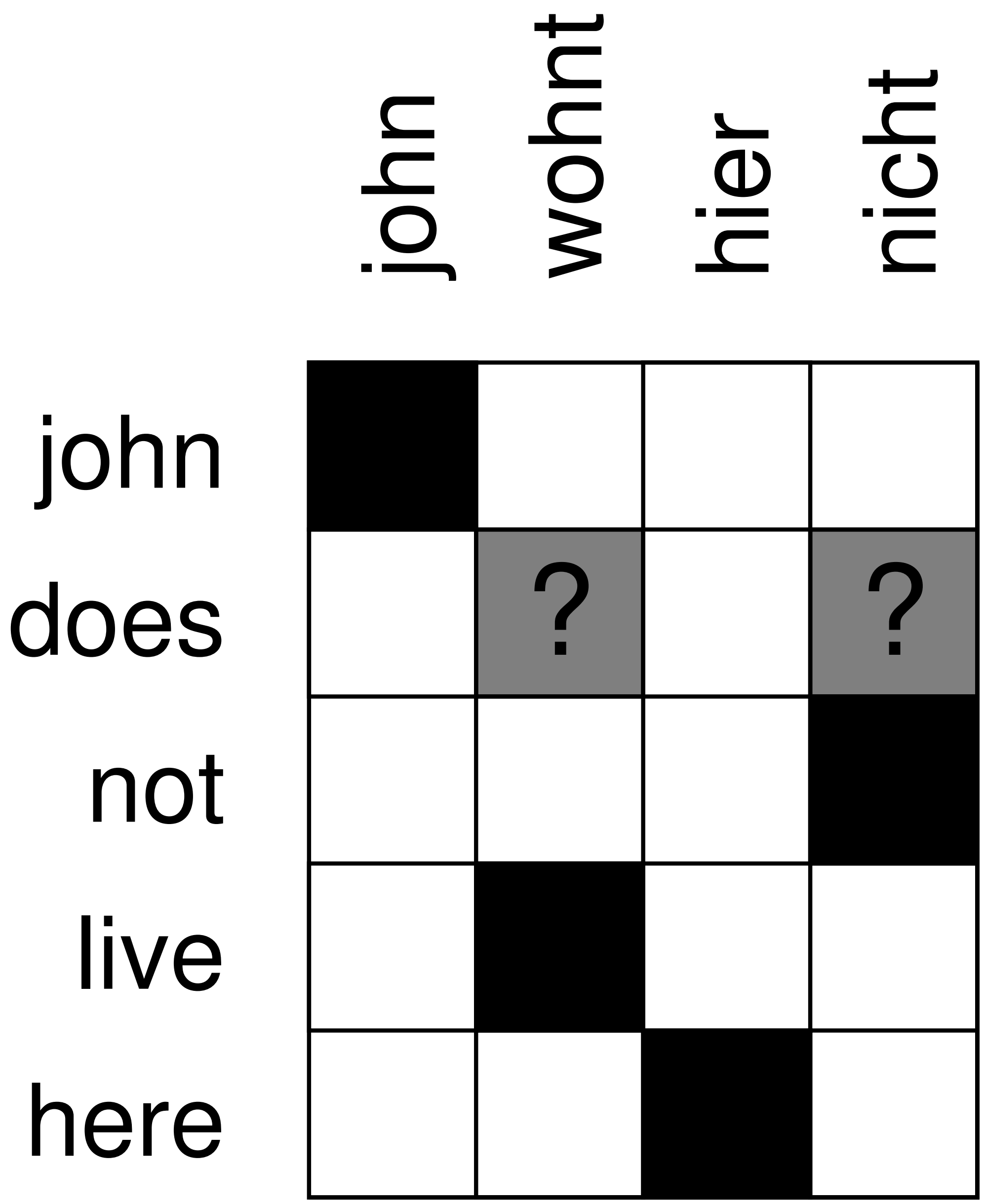
- Introduced IBM Models, a basis of SMT
- Derived iterative procedure for estimation
  - Generative model, EM-algorithm
  - Higher models (Model 1-5, HMM)
- We can answer a question: P(f | e) = ?
  - By-product, we can also answer two questions: P(f, a | e) = ? and P(a | f, e) = ?

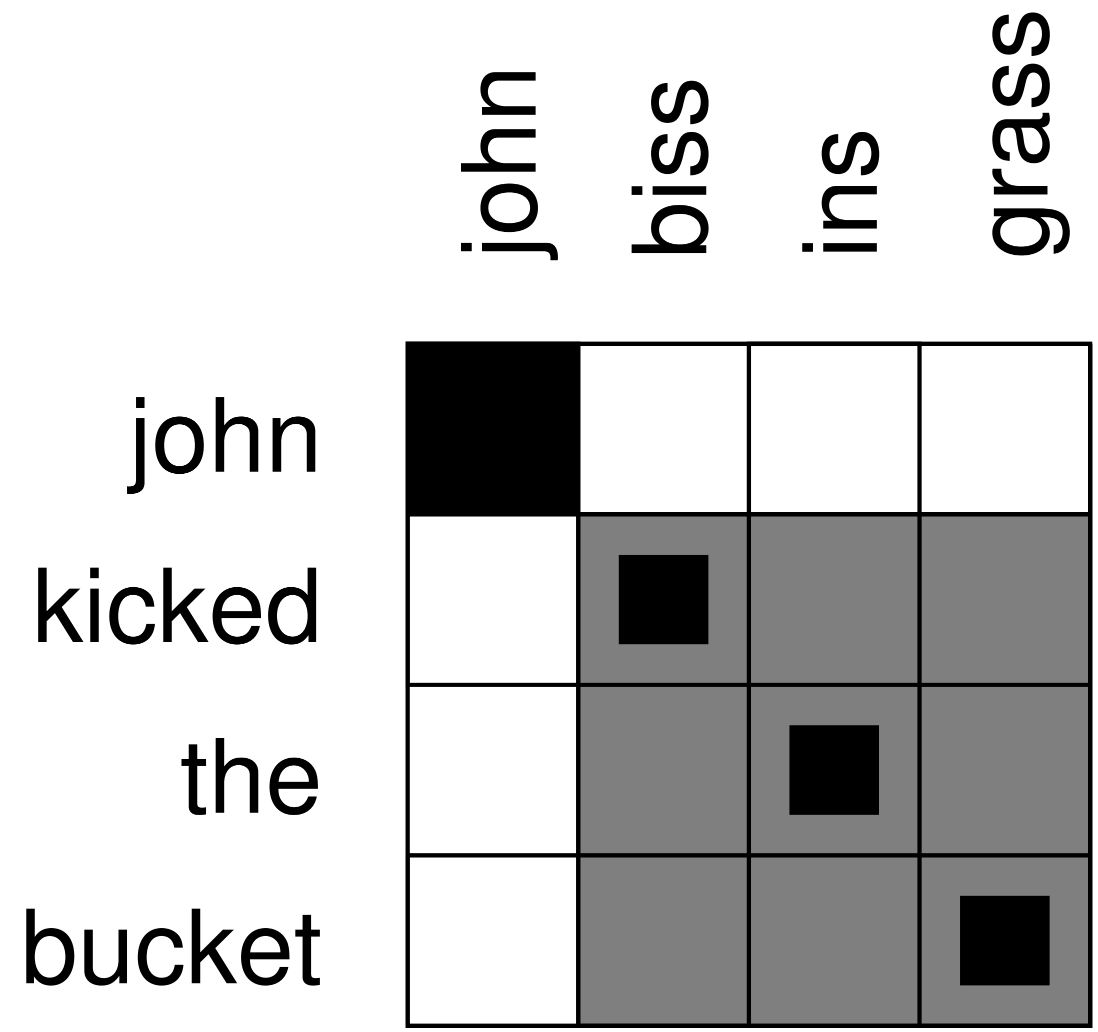
Word Alignment



- Given a sentence pair, can we compute word correspondence? (An example from Chapter 4 of Koehn, 2009)

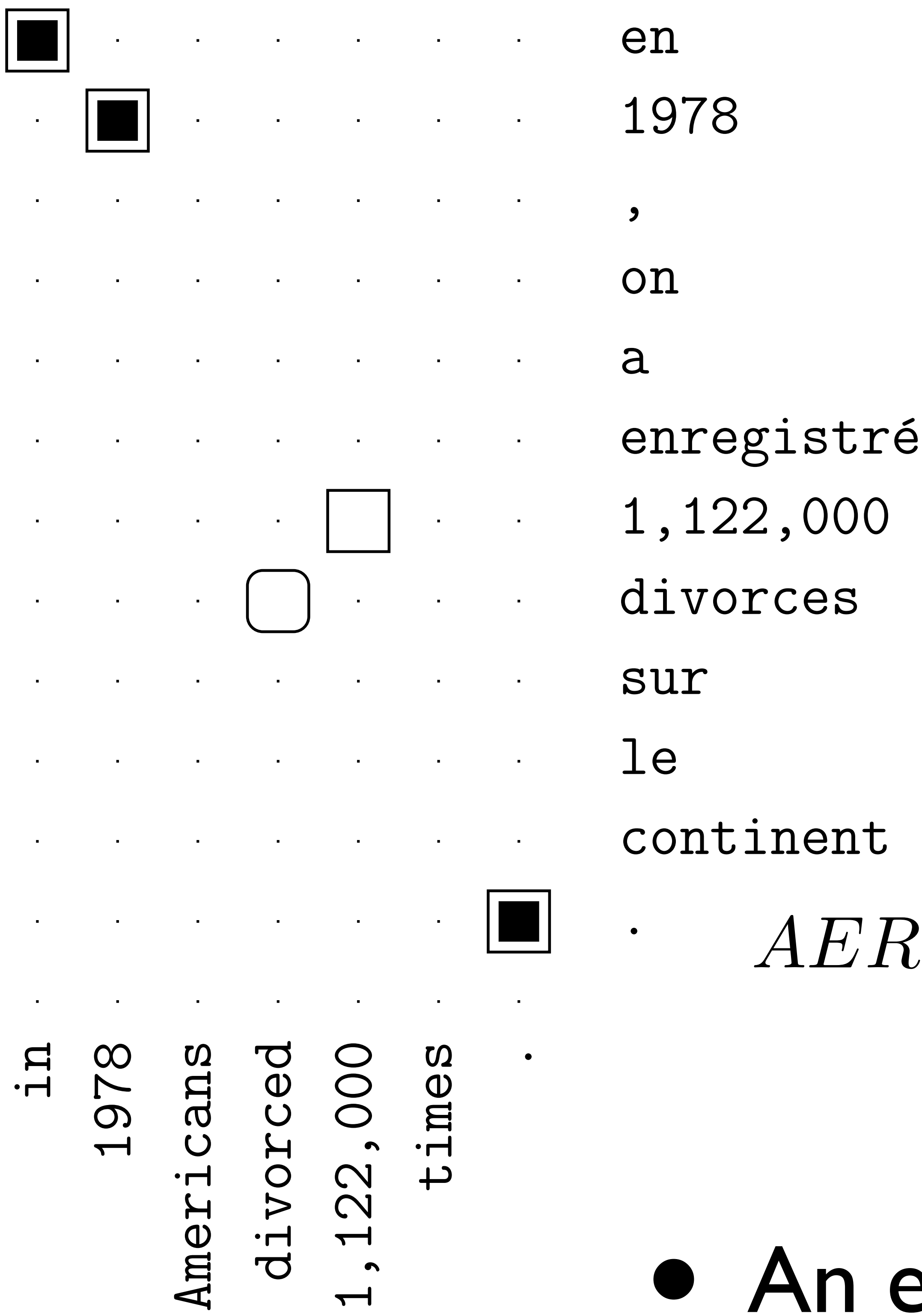
Word Alignment?

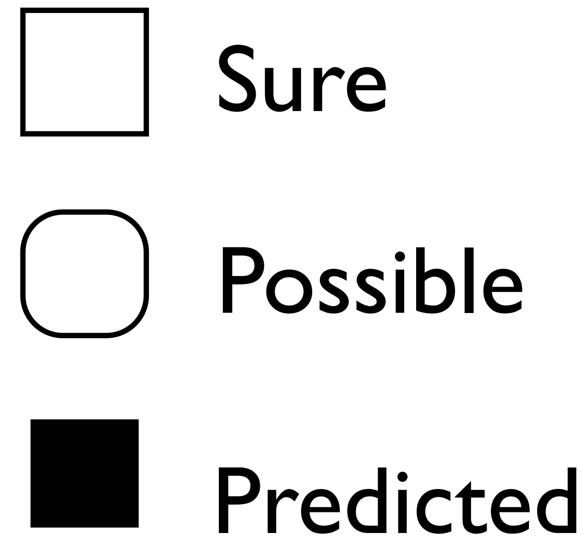




- one-to-many for does-to-{wohnt, nicht}
- phrasal correspondence in “kicked the bucket”

Alignment Error Rate





$$
\begin{aligned}
AER(A,S,P) &= \left(1 - \frac{|A \cap S| + |A \cap P|}{|A| + |S|}\right) \\
&= \left(1 - \frac{3+3}{3+4}\right) = \frac{1}{7}
\end{aligned}
$$

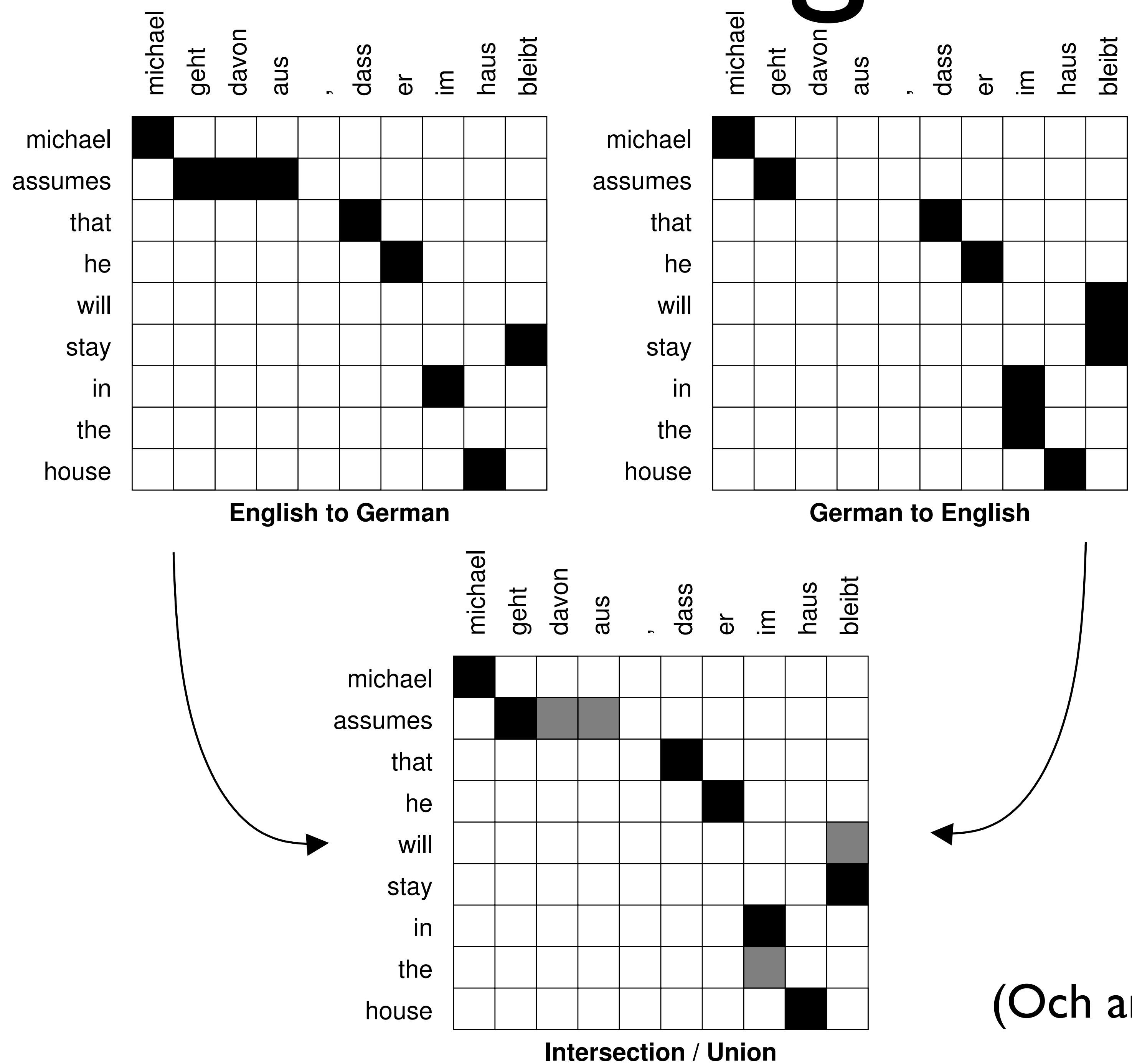
- An example from (Taskar et al., 2005)

AER Results

Model	Training scheme	0.5K	8K	128K	1.47M
Dice		50.9	43.4	39.6	38.9
Dice+C		46.3	37.6	35.0	34.0
Model 1	1^5	40.6	33.6	28.6	25.9
Model 2	$1^5 2^5$	46.7	29.3	22.0	19.5
HMM	$1^5 H^5$	26.3	23.3	15.0	10.8
Model 3	$1^5 2^5 3^3$	43.6	27.5	20.5	18.0
	$1^5 H^5 3^3$	27.5	22.5	16.6	13.2
Model 4	$1^5 2^5 3^3 4^3$	41.7	25.1	17.3	14.1
	$1^5 H^5 3^3 4^3$	26.1	20.2	13.1	9.4
	$1^5 H^5 4^3$	26.3	21.8	13.3	9.3
Model 5	$1^5 H^5 4^3 5^3$	26.5	21.5	13.7	9.6
	$1^5 H^5 3^3 4^3 5^3$	26.5	20.4	13.4	9.4
Model 6	$1^5 H^5 4^3 6^3$	26.0	21.6	12.8	8.8
	$1^5 H^5 3^3 4^3 6^3$	25.9	20.3	12.5	8.7

- Fr-En Hansard Task (Och and Ney, 2003)

Symmetric Alignment



(Och and Ney, 2003)

- Take intersection of two directions
- Heuristic to add union alignment points

Agreement Training

$$\text{E-step:} \quad q(\mathbf{a};\mathbf{f},\mathbf{e}) = \frac{1}{Z_{\mathbf{f},\mathbf{e}}} p_1(\mathbf{a}|\mathbf{f},\mathbf{e};\theta_1) \cdot p_2(\mathbf{a}|\mathbf{e},\mathbf{f};\theta_2)$$

$$\text{M-step:} \quad \theta' = \underset{\theta}{\operatorname{argmax}} \sum_{\mathbf{f},\mathbf{e},\mathbf{a}} q(\mathbf{a};\mathbf{f},\mathbf{e}) \log p_1(\mathbf{f},\mathbf{e},\mathbf{a};\theta_1) + \sum_{\mathbf{f},\mathbf{e},\mathbf{a}} q(\mathbf{a};\mathbf{f},\mathbf{e}) \log p_2(\mathbf{f},\mathbf{e},\mathbf{a};\theta_2)$$

- As an alternative to the heuristic approach, we can enforce agreement of two models during EM-algorithm (Liang et al., 2006)
  - Summation is intractable: Approximate q by multiple of $q_{i,j}(\Theta;f,e)$ from two models
  - M-step is performed for each individual model

Posterior Constraints

$$
\begin{aligned}
q_{i,j}(\theta, \lambda; \mathbf{f}, \mathbf{e}) &\leftarrow \frac{t_\theta(f_j|e_i)e^{\lambda_{i,j}}}{\sum_{i'=0}^{l} t_\theta(f_j|e_{i'})e^{\lambda_{i',j}}} \\
q_{j,i}(\theta, \lambda; \mathbf{e}, \mathbf{f}) &\leftarrow \frac{t_\theta(e_i|f_j)e^{-\lambda_{i,j}}}{\sum_{j'=0}^{m} t_\theta(e_i|f_{j'})e^{-\lambda_{i,j'}}} \\
\lambda_{i,j} &\leftarrow \lambda_{i,j} - q_{i,j}(\theta, \lambda; \mathbf{f}, \mathbf{e}) + q_{j,i}(\theta, \lambda; \mathbf{e}, \mathbf{f})
\end{aligned}
$$

- Another objective to make an agreement (Ganchev et al., 2008)
- Additional projection step to adjust λ so that two posterior probabilities $q_{i,j}()$ and $q_{j,i}()$ agree

Other Topics for Alignment

- Supervised training (Taskar et al., 2005; Haghighi et al., 2009)
- Unsupervised training with many features (Berg-Kirkpatrick et al., 2010; Dyer et al., 2011)
- Syntactically constrained alignment (DeNero and Klein, 2007; Burkett et al. 2010; Riesa and Marcu, 2010; Pauls et al., 2010)
- Phrasal alignment (Marcu and Wong, 2002; Blunsom et al., 2009; Neubig et al., 2011)

Implementations

- Language Model
  - SRILM (http://www-speech.sri.com/projects/srilm/)
  - BerkeleyLM (http://code.google.com/p/berkeleylm/)
  - kenlm (http://kheafield.com/code/kenlm/)
- IBM Models
  - GIZA++ (http://code.google.com/p/giza-pp/)
  - MGIZA (http://geek.kyloo.net/software/doku.php/mgiza:overview)
- Agreement/Posterior constrained training
  - BerkeleyAligner (http://code.google.com/p/berkeleyaligner/)
  - PostCat (http://www.seas.upenn.edu/~strctlrn/CAT/CAT.html)

References

- P. F. Brown, S. A. D. Pietra, V. J. D. Pietra, and R. L. Mercer, ``The mathematics of statistical machine translation: Parameter estimation," *Computational Linguistics*, vol. 19, no. 2, pp. 263--311, 1993.

- K. Toutanova and M. Galley, ``Why initialization matters for ibm model 1: Multiple optima and non-strict convexity," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, (Portland, Oregon, USA), pp. 461--466, Association for Computational Linguistics, June 2011.

- P. Koehn, *Statistical Machine Translation*. Cambridge University Press, 2009.

- F. J. Och and H. Ney, ``A systematic comparison of various statistical alignment models," *Computational Linguistics*, vol. 29, pp. 19--51, March 2003.

- B. Taskar, S. Lacoste-Julien, and D. Klein, ``A discriminative matching approach to word alignment," in *HLT '05:* Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing, (Morristown, NJ, USA), pp. 73--80, Association for Computational Linguistics, 2005.

- P. Liang, B. Taskar, and D. Klein, ``Alignment by agreement," in *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*, (New York City, USA), pp. 104--111, Association for Computational Linguistics, June 2006.

References

- K. Ganchev, J. a.V. Grac ¸a, and B. Taskar, ``Better alignments = better translations?," in *Proceedings of ACL-08: HLT*, (Columbus, Ohio), pp. 986--993, Association for Computational Linguistics, June 2008.
- A. Haghighi, J. Blitzer, J. DeNero, and D. Klein, ``Better word alignments with supervised itg models," in *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, (Suntec, Singapore), pp. 923--931, Association for Computational Linguistics, August 2009.
- T. Berg-Kirkpatrick, A. Bouchard-Coˆte ´, J. DeNero, and D. Klein, ``Painless unsupervised learning with features," in *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, (Los Angeles, California), pp. 582--590, Association for Computational Linguistics, June 2010.
- C. Dyer, J. H. Clark, A. Lavie, and N. A. Smith, ``Unsupervised word alignment with arbitrary features," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, (Portland, Oregon, USA), pp. 409--419, Association for Computational Linguistics, June 2011.
- J. DeNero and D. Klein, ``Tailoring word alignments to syntactic machine translation," in *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, (Prague, Czech Republic), pp. 17--24, Association for Computational Linguistics, June 2007.
- D. Burkett, J. Blitzer, and D. Klein, ``Joint parsing and alignment with weakly synchronized grammars," in *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, (Los Angeles, California), pp. 127--135, Association for Computational Linguistics, June 2010.

References

- J. Riesa and D. Marcu, ``Hierarchical search for word alignment," in *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, (Uppsala, Swe-den), pp. 157--166, Association for Computational Linguistics, July 2010.
- A. Pauls, D. Klein, D. Chiang, and K. Knight, ``Unsupervised syntactic alignment with inversion transduction grammars," in *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, (Los Angeles, California), pp. 118--126, Association for Computational Linguistics, June 2010.
- D. Marcu and W. Wong, ``A phrase-based, joint probability model for statistical machine translation," in *Proc. of EMNLP-2002*, (Philadelphia, PA), July 2002.
- P. Blunsom, T. Cohn, C. Dyer, and M. Osborne, ``A gibbs sampler for phrasal synchronous grammar induction," in *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, (Suntec, Singapore), pp. 782--790, Association for Computational Linguistics, August 2009.
- G. Neubig, T. Watanabe, E. Sumita, S. Mori, and T. Kawahara, ``An unsupervised model for joint phrase alignment and extraction," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, (Portland, Oregon, USA), pp. 632-641, Association for Computational Linguistics, June 2011.
- T. Brants, A. C. Popat, P. Xu, F. J. Och, and J. Dean, ``Large language models in machine translation," in *Proceedings of the 2007 Joint Conference on Empirical Methods in Natu- ral Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pp. 858--867, 2007.