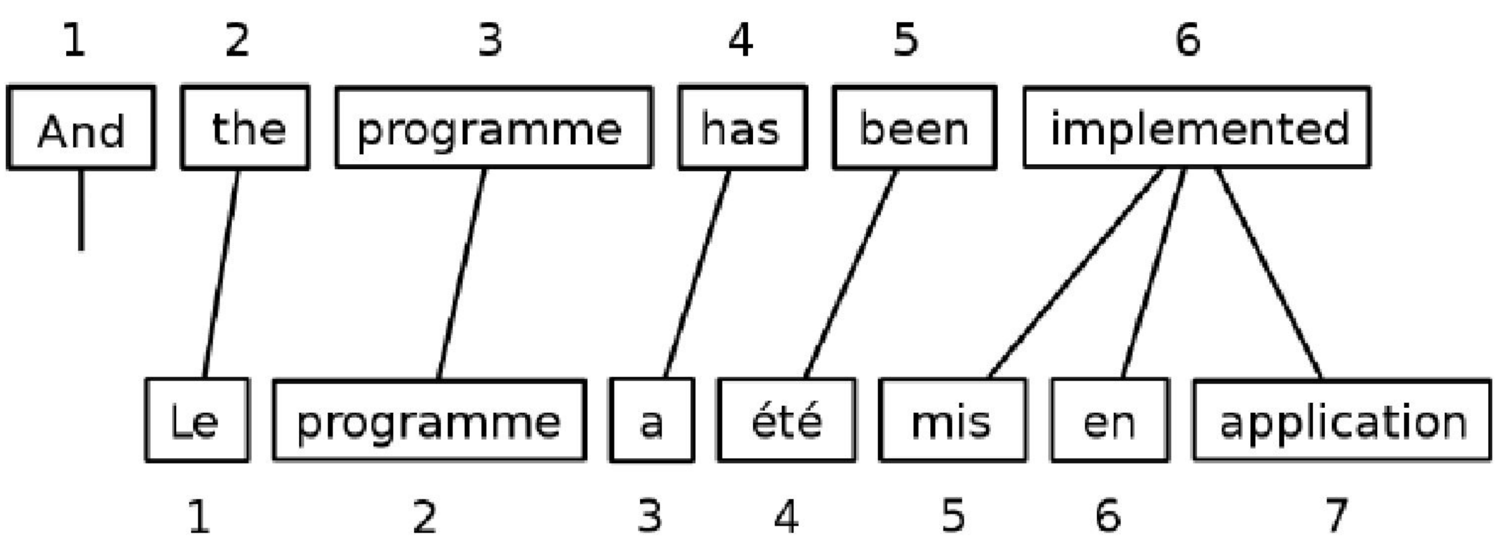
# Homework 4: Word Alignment

11-711 Fall 2018 Recitation

Anjalie Field
(Slides help from Maria Ryskina, Sachin Kumar)

# Task

- Align words between English and French Sentences



# Evaluation

- Alignment Error Rate
- Performance on machine translation (BLEU)

# Requirement: Three Alignment Models

- Heuristic Alignment
  - Not probabilistic, e.g. $c(f, e)/(c(e) \cdot c(f))$
- IBM Model 1
- HMM Model of Vogel et. al (1996)

# Code structure

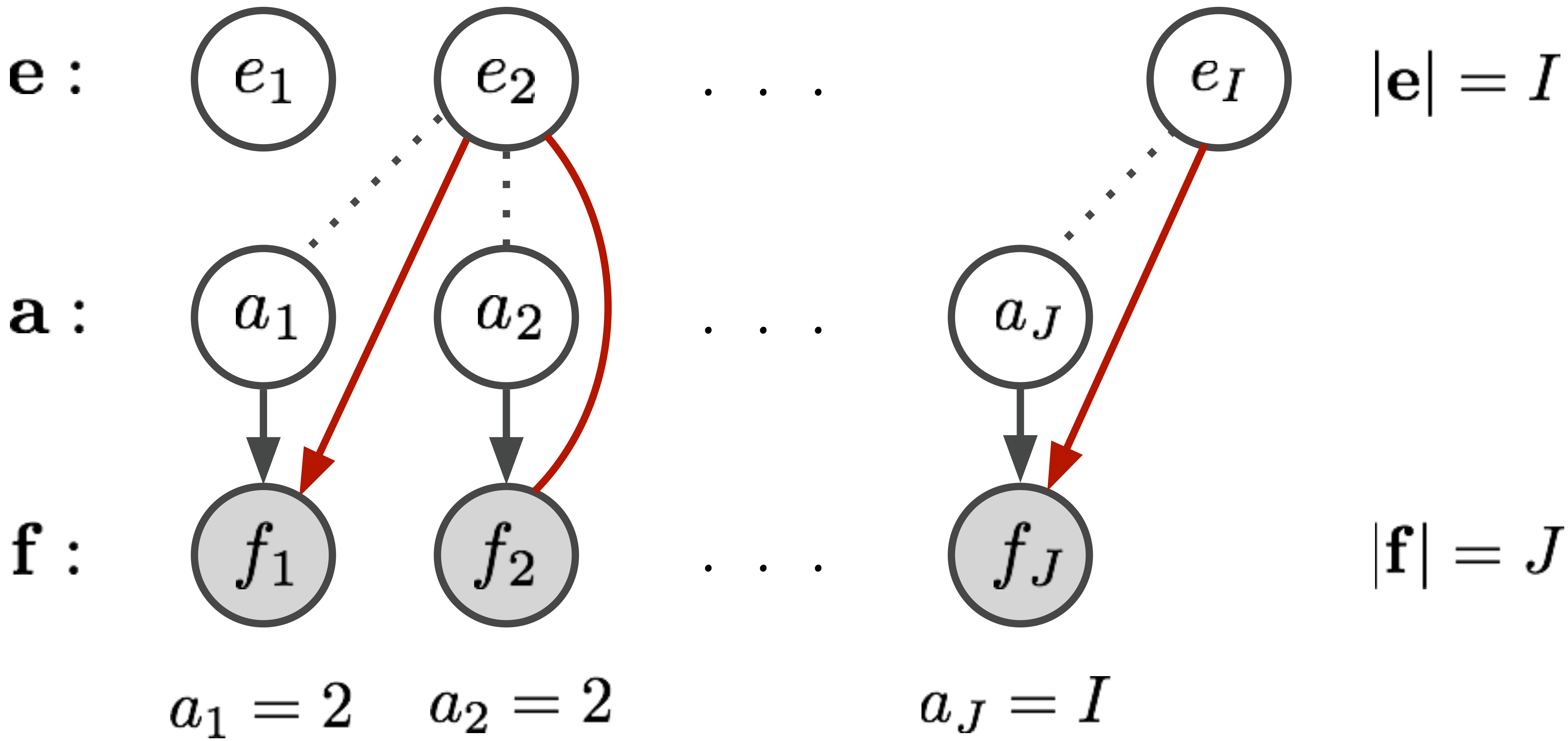
```
package edu.berkeley.nlp.mt;

public interface WordAligner
{
Alignment alignSentencePair(SentencePair sentencePair);
}
```

You must return an object of type "WordAligner" (your class should inherit this)

You must implement this: returns best alignment

# IBM Model 1




For some fixed alignment a

$$p(\mathbf{f}, \mathbf{a} | \mathbf{e}) = \prod_{j=1}^{J} p(f_j | e_{a_j}) p(a_j)$$

# Model 1: parameters

Emission (translation) probabilities: $\theta_{f,e}$

word types: $p(\text{chat}|\text{cat})$

Training objective: $\max_{\theta} p(\mathbf{f}|\mathbf{e},\theta) = \max_{\theta} \sum_{\mathbf{a}} p(\mathbf{f},\mathbf{a}|\mathbf{e},\theta)$

$$p(\mathbf{f},\mathbf{a}|\mathbf{e},\theta) = \prod_{j=1}^{J} p(a_j = i) p(f_j|e_i,\theta)$$

$$p(a_j = i) = \frac{1}{I+1}$$

uniform prior

$$p(f_j = f|e_i = e) = \theta_{f,e}$$

translation probability

# Model 1: EM algorithm

- E-step: impute missing data $\leftarrow \mathbf{a}$
- M-step: estimate parameters based on imputed complete data $\leftarrow \theta$

E-step: computing expected counts:

$$d_{f,e}(\theta) = \mathbb{E}_{p(\mathbf{a}|\mathbf{f},\mathbf{e},\theta)}[c_{f,e}]$$

M-step: reestimating parameters:

$$\theta_{f,e} \propto d_{f,e}(\theta)$$

# Model 1: E-step

Expected counts:

$$d_{f,e}(\theta) = \mathbb{E}_{p(\mathbf{a}|\mathbf{f},\mathbf{e},\theta)}[c_{f,e}]$$

At iteration $t$ :

$$d^{(t)}_{f,e}(\theta) = \sum_{i=1}^{I}\sum_{j=1}^{J} \underbrace{\mathbb{1}[f_j = f]}_{\text{french token of type } f} \underbrace{\mathbb{1}[a_j = i]}_{\text{aligned to}} \underbrace{\mathbb{1}[e_i = e]}_{\text{English token of type } e} \cdot p(a_j = i|\mathbf{f},\mathbf{e},\theta^{(t)})$$

# Model 1: E-step

Computing posteriors:

$$p(\mathbf{a}|\mathbf{f},\mathbf{e},\theta) = \frac{p(\mathbf{f},\mathbf{a}|\mathbf{e},\theta)}{p(\mathbf{f}|\mathbf{e},\theta)} = \prod_{j=1}^{J} \frac{p(f_j, a_j|\mathbf{e},\theta)}{p(f_j|\mathbf{e},\theta)}$$

$$p(a_j|\mathbf{f},\mathbf{e},\theta) = \frac{p(f_j, a_j|\mathbf{e},\theta)}{p(f_j|\mathbf{e},\theta)}$$

$$p(a_j = i|\mathbf{f},\mathbf{e},\theta^{(t)}) = \frac{\theta^{(t)}_{f_j,e_i} \cdot p(a_j = i)}{\sum_{k=1}^{I} \theta^{(t)}_{f_j,e_k} \cdot p(a_j = k)}$$

# Model 1: M-step

Reestimating parameters:

$$\theta_{f,e}^{(t+1)} \propto d_{f,e}^{(t)}(\theta^{(t)})$$

$$\theta_{f,e}^{(t+1)} = \frac{d_{f,e}^{(t)}(\theta^{(t)})}{\sum_{\tilde{f}} d_{\tilde{f},e}^{(t)}(\theta^{(t)})}$$

# HMM Model Intuition

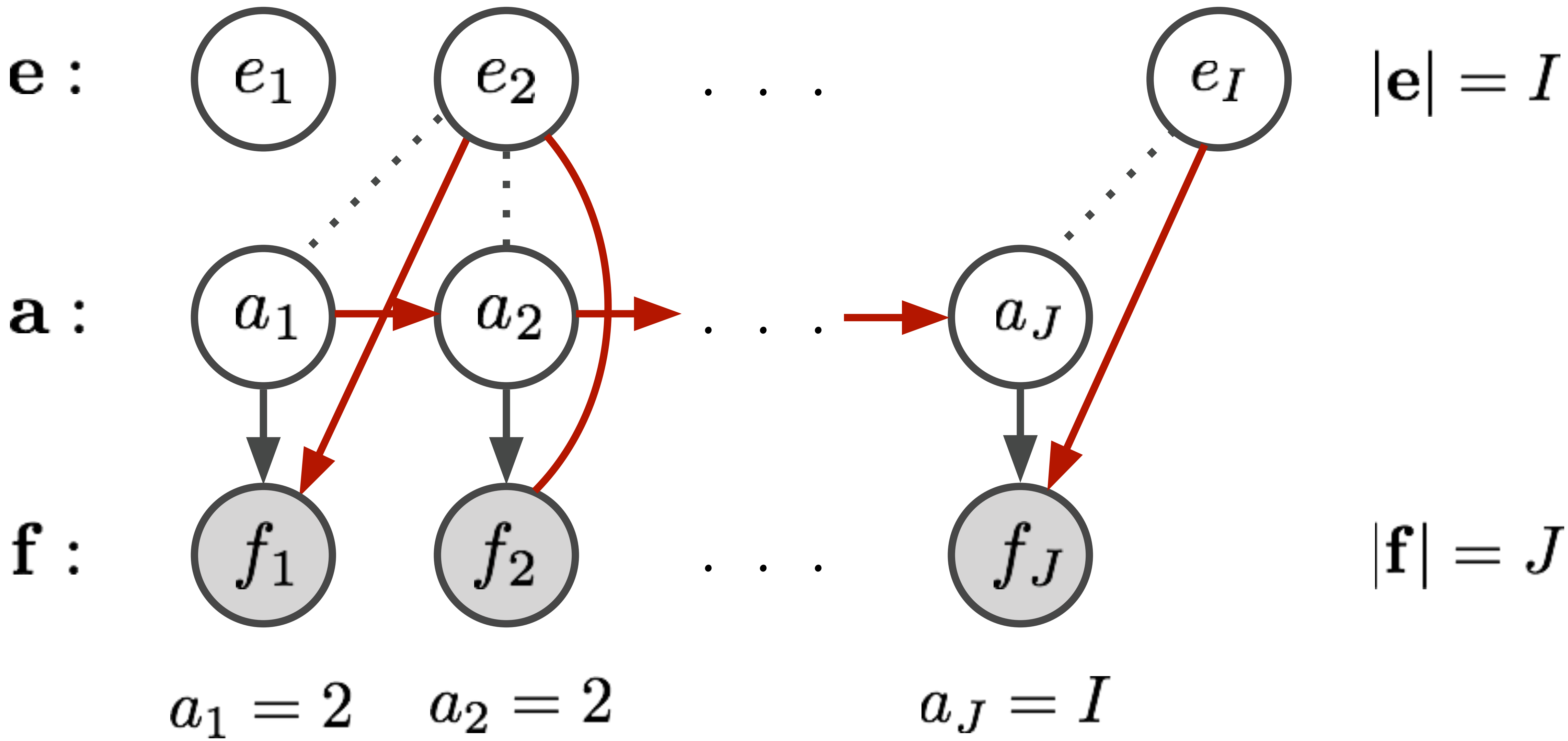
**Ella lleva el libro azul**

**The girl carried the blue book**

- "azul" is more likely to align to "blue" than to "girl" because "libro" aligned to "book"

- Local alignment patterns are likely
  [side note: IBM2 assumes global alignment patterns are likely]

# HMM Model




$$p(\mathbf{f}, \mathbf{a} | \mathbf{e}) = \prod_{j=1}^{J} p(f_j | e_{a_j}) p(a_j | a_{j-1})$$

# HMM Model Intuition

**Ella lleva el libro azul**

**1 2 3 5 4**

**The girl carried the blue book**

5 – 4 = 1 (small jump)

If we align “azul” to “girl”: 5 – 1 = 4 (bigger jump)

# HMM: parameters

Emission (translation) probabilities: $\theta_{f,e}$

Transition ("jump") probabilities: $\psi_k$

$$p(a_j = i | a_{j-1} = i', \psi) = \psi_{|i-i'|}$$

Try other bucketing strategies!

Transition probabilities can be:

- fixed: e.g. $\psi_k \propto \exp(-\lambda(k-1))$
- learned with EM algorithm

# HMM: EM algorithm

E-step: computing expected counts:

$$d_{f,e}(\theta) = \mathbb{E}_{p(\mathbf{a}|\mathbf{f},\mathbf{e},\psi,\theta)}[c_{f,e}]$$

$$d_k(\psi) = \mathbb{E}_{p(\mathbf{a}|\mathbf{f},\mathbf{e},\psi,\theta)}[c_{|a_j - a_{j-1}|=k}]$$

M-step: reestimating parameters:

$$\theta_{f,e} \propto d_{f,e}(\theta)$$

$$\psi_k \propto d_k(\psi)$$

# HMM: E-step

Expected counts:

$$d_{f,e}^{(t)}(\theta) = \sum_{i=1}^{I} \sum_{j=1}^{J} \mathbb{1}[f_j = f]\mathbb{1}[a_j = i]\mathbb{1}[e_i = e] \times$$
$$\times\, p(a_j = i | \mathbf{f}, \mathbf{e}, \theta^{(t)}, \psi^{(t)})$$

$$d_{k}^{(t)}(\psi) = \sum_{i=1}^{I} \sum_{i'=1}^{I} \sum_{j=1}^{J} \mathbb{1}[a_j = i]\mathbb{1}[|i - i'| = k]\mathbb{1}[a_{j-1} = i'] \times$$
$$\times\, p(a_j = i, a_{j-1} = i' | \mathbf{f}, \mathbf{e}, \theta^{(t)}, \psi^{(t)})$$

# HMM: E-step

Computing posteriors: forward-backward

$\alpha_j^i$ — sum of all paths up to $a_j = i$

$\beta_j^i$ — sum of all paths starting from $a_j = i$

$$p(a_j = i | \mathbf{f}, \mathbf{e}, \theta^{(t)}, \psi^{(t)}) = \frac{\alpha_j^i \beta_j^i}{Z}$$

$$p(a_j = i, a_{j-1} = i' | \mathbf{f}, \mathbf{e}, \theta^{(t)}, \psi^{(t)}) = \frac{\alpha_{j-1}^{i'} \cdot \beta_j^i \cdot \psi_{|i-i'|}^{(t)} \cdot \theta_{f_j, e_i}}{Z}$$

# Recursive Computation

$$\alpha_j^i = \sum_k \alpha_{j-1}^k \psi_{|k-j|} \theta_{e_i, f_j}$$

$$\alpha_0^i = \psi_{|0-i|} \theta_{e_i, f_0}$$

$$\beta_j^i = \sum_k \beta_{j+1}^k \psi_{|k-j|} \theta_{e_i, f_j}$$

$$\beta_{J-1}^i = 1$$

# HMM: M-step

Reestimating parameters:

$$\theta_{f,e}^{(t+1)} = \frac{d_{f,e}^{(t)}(\theta^{(t)})}{\sum_{\tilde{f}} d_{\tilde{f},e}^{(t)}(\theta^{(t)})}$$

$$\psi_k^{(t+1)} = \frac{d_k(\psi^{(t)})}{\sum_l d_l(\psi^{(t)})}$$

# Computing Best Alignment: Viterbi

- Essentially forward pass of forward-backwards algorithm
- For each word in the sentence, compute probability for each possible alignment and store backpointer to best preceding alignment
- After the last word, trace backpointers to get overall best alignment
- [For IBM1, just need to compute best English word for each French word]

# Possible solutions for null

- Fixed transition to null:

$$p(a_j = \text{null}|a_{j-1} = i') = \epsilon$$

$$p(a_j = i|a_{j-1} = i') = (1-\epsilon)\psi_{|i-i'|}$$

- Uniform transition from null:

$$p(a_j = i|a_{j-1} = \text{null}) = \frac{1-\epsilon}{I}$$

- Smarter: insert a null for every target word (Och & Ney '03)

- Special prior on null:

$$p(a_j = \text{null}) = \epsilon$$

$$p(a_j = i) = \frac{1-\epsilon}{I}$$

# Incorporation of NULL

**A la mujer le encanta aprender**

**The woman loves to learn NULL**

# Other Tricks

- "Intersected"
  - Separately train f → e and e → f alignment models
  - When computing the best alignment for a sentence pair, only align $f_j$ to $e_i$ if both models find that they should align
- *SloppyMath.logadd*
  - Use this when you want to sum log probabilities:
  - P(A)P(B) → logP(A) + logP(B)
  - P(A) + P(B) → SloppyMath.logadd(logP(A), logP(B))

# Questions?