CMP462: Natural Language Processing

Lecture 14: IBM Translation Models

Mohamed Alaa El-Dien Aly
Computer Engineering Department
Cairo University

Spring 2013

Agenda

- IBM Model 1
- IBM Model 2
- Training of Models 1 and 2

Acknowledgment:
Most slides adapted from Michael Collins NLP class on Coursera.

IBM Model 1: Alignments

- How do we model $p(f|e)$?

  English: The dog eats
  French: Le chien mange

- English sentence e has l words $e_1, \ldots, e_l$

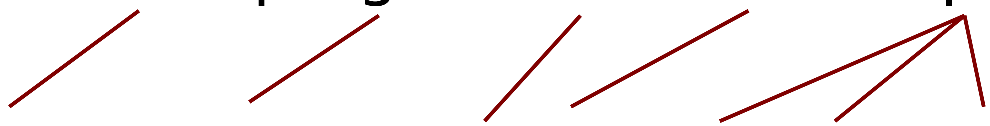
  French sentence f has m words $f_1, \ldots, f_m$

- An alignment a identifies the source of each *french* word
  - Above: {1, 2, 3}
- Formally, an alignment a is $\{a_1, \ldots, a_m\}$ where each $a_i \in \{0, \ldots, l\}$. Why 0?
  - French words with no English equivalent (*NULL* word)
- How many possible alignments?
  - $(l+1)^m$

IBM Model 1: Alignments

e.g., l = 6, m = 7

e = And the program has been implemented



f = Le programme a ete mis en application

One alignment is

{2, 3, 4, 5, 6, 6, 6}

Another (bad!) alignment is

{1, 1, 1, 1, 1, 1, 1}

Alignments in IBM Models

- We'll define models for $p(a|e, m)$ and $p(f \mid a, e, m)$, giving

$$p(f, a|e, m) = p(a|e, m)\, p(f|a, e, m)$$

Example: e = the dog eats m = 3

f = f_1 f_2 f_3

We can estimate $p(\text{le chien mange}, \{1, 2, 3\}|\text{the dog eats}, 3)$

- Also,

$$p(f \mid e, m) = \sum_{a \in A} \overbrace{p(a \mid e, m)\, p(f \mid a, e, m)}^{p(f,\, a \,\mid\, e,\, m)}$$

where A is the set of all possible alignments

By-product: Most Likely Alignments

- Once we have a model for $p(f, a \mid e, m) = p(a \mid e, m)\, p(f \mid a, e, m)$, we can calculate

$$p(a \mid f, e, m) = \frac{p(f, a \mid e, m)}{\underbrace{\sum_{\alpha \in A} p(f, \alpha \mid e, m)}_{p(f \mid e, m)}}$$

  for any alignment a.

- For a given f, e pair, we can also compute the most likely alignment

$$a^* = \operatorname{argmax}_a p(a \mid f, e, m)$$

- Nowadays, these IBM models are rarely used for translation, but are used for recovering alignments

Example Alignment

French:

le conseil a rendu son avis , et nous devons à présent adopter un nouvel avis sur la base de la première position .

English:

the council has stated its position , and now , on the basis of the first position , we again have to give our opinion .

Alignment:

the/le council/conseil has/à stated/rendu its/son position/avis ,/, and/et now/présent ,/NULL on/sur the/le basis/base of/de the/la first/première position/position ,/NULL we/nous again/NULL have/devons to/a give/adopter our/nouvel opinion/avis ./.

Alignment from *English* to *French*

IBM Model 1: Alignments

- Recall: $p(f, a|e, m) = p(a|e, m)\, p(f|a, e, m)$
- In IBM Model 1, all alignments are equally likely:

$$p(a|e, m) = \frac{1}{(l+1)^m}$$

- This is a major simplifying assumption …

IBM Model 1: Translation Probabilities

- Recall: $p(f, a | e, m) = p(a | e, m)\, p(f | a, e, m)$

- In IBM Model 1, this is:

$$p(f | a, e, m) = \prod_{i=1}^{m} t(f_i | e_{a_i})$$

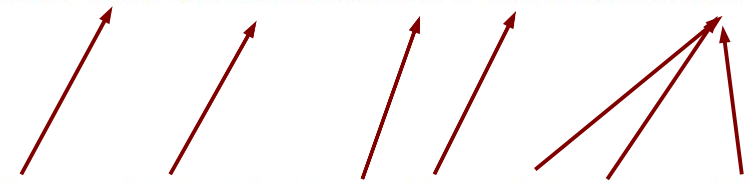
Example: e = the dog eats
f = le chien mange
m = 3, a = {1, 2, 3}

$$p(f | a, e, m) = t(\text{le} | \text{the}) \times t(\text{chien} | \text{dog}) \times t(\text{mange} | \text{eats})$$

Another Example

e.g., $l = 6$, $m = 7$

$e =$ And the program has been implemented



$f =$ Le programme a ete mis en application

$a = \{2, 3, 4, 5, 6, 6, 6\}$

$$
\begin{aligned}
p(f \mid a, e, m) = \; & t(Le \mid the) \times \\
& t(programme \mid program) \times \\
& t(a \mid has) \times \\
& t(ete \mid been) \times \\
& t(mis \mid implemented) \times \\
& t(en \mid implemented) \times \\
& t(application \mid implemented)
\end{aligned}
$$

IBM Model 1: The Generative Process

To generate a French string *f* from an English string *e*

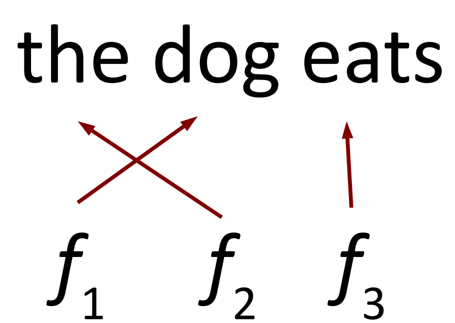
- *Step 1*: Pick an alignment with probability

$$p(a|e,m)=\frac{1}{(l+1)^m}$$

- *Step 2*: Given the alignment, pick the french words with probability

$$p(f|a,e,m)=\prod_{i=1}^{m} t(f_i|e_{a_i})$$

Example:



generate f_1 from $t(\text{-}\mid\text{dog})$, f_2 from $t(\text{-}\mid\text{the})$ … etc

IBM Model 1: The Generative Process

To generate a French string *f* from an English string *e*

- *Step 1*: Pick an alignment with probability

$$p(a|e,m)=\frac{1}{(l+1)^m}$$

- *Step 2*: Given the alignment, pick the french words with probability

$$p(f|a,e,m)=\prod_{i=1}^{m} t(f_i|e_{a_i})$$

The final result for IBM Model 1:

$$p(f,a|e,m)=p(a|e,m)\,p(f|a,e,m)=\frac{1}{(l+1)^m}\prod_{i=1}^{m} t(f_i|e_{a_i})$$

An Example Lexical Entry

English	French	Probability
position	position	0.756715
position	situation	0.0547918
position	mesure	0.0281663
position	vue	0.0169303
position	point	0.0124795
position	attitude	0.0108907

$t(- \mid \text{position})$

. . . de la situation au niveau des négociations de l ' ompi . . .
. . . of the current position in the wipo negotiations . . .

nous ne sommes pas en mesure de décider , . . .
we are not in a position to decide , . . .

. . . le point de vue de la commission face à ce problème complexe .
. . . the commission 's position on this complex problem .

IBM Model 2

- Only difference: *alignment* or *distortion* parameters

$$q(j|i,l,m)$$

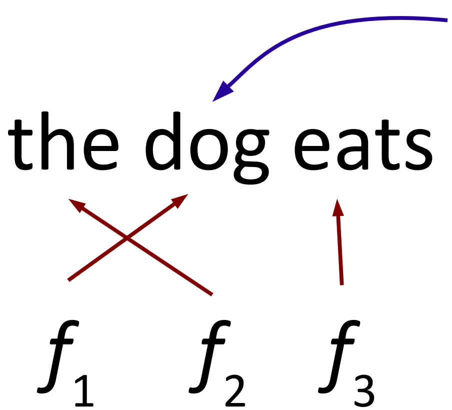
Probability that i^{th} French word is connected to j^{th} English word, given sentence lengths of e and f are l and m

- Define $p(a|e,m)=\prod_{i=1}^{m} q(a_i|i,l,m)$

where $a = \{a_1, \ldots, a_m\}$

- Example:

$$q(a_1=2|i=1,l=3,m=3)$$



$$a=\{2,1,3\}$$

$$p(a|e,m)=q(2|1,3,3)\times q(1|2,3,3)\times q(3|3,3,3)$$

IBM Model 2

- Only difference: *alignment* or *distortion* parameters

$$q(j|i,l,m)$$

Probability that j^{th} French word is connected to i^{th} English word, given sentence lengths of e and f are l and m

- Define $p(a|e,m)=\prod_{i=1}^{m} q(a_i|i,l,m)$

where $a = \{a_1, \ldots, a_m\}$

- The final result for IBM Model 2:

$$p(f,a|e,m)=\prod_{i=1}^{m} q(a_i|i,l,m)\,t(f_i|e_{a_i})$$

Another Example

$$
\begin{aligned}
l &= 6 \\
m &= 7 \\
e &= \text{And the program has been implemented} \\
f &= \text{Le programme a ete mis en application} \\
a &= \{2, 3, 4, 5, 6, 6, 6\}
\end{aligned}
$$

$$
\begin{aligned}
p(a \mid e, 7) = \; & \mathbf{q}(2 \mid 1, 6, 7) \times \\
& \mathbf{q}(3 \mid 2, 6, 7) \times \\
& \mathbf{q}(4 \mid 3, 6, 7) \times \\
& \mathbf{q}(5 \mid 4, 6, 7) \times \\
& \mathbf{q}(6 \mid 5, 6, 7) \times \\
& \mathbf{q}(6 \mid 6, 6, 7) \times \\
& \mathbf{q}(6 \mid 7, 6, 7)
\end{aligned}
$$

Another Example

$$\begin{aligned} l &= 6 \\ m &= 7 \\ e &= \text{And the program has been implemented} \\ f &= \text{Le programme a ete mis en application} \\ a &= \{2, 3, 4, 5, 6, 6, 6\} \end{aligned}$$

$$\begin{aligned} p(f \mid a, e, 7) = \; & \mathbf{t}(Le \mid the) \times \\ & \mathbf{t}(programme \mid program) \times \\ & \mathbf{t}(a \mid has) \times \\ & \mathbf{t}(ete \mid been) \times \\ & \mathbf{t}(mis \mid implemented) \times \\ & \mathbf{t}(en \mid implemented) \times \\ & \mathbf{t}(application \mid implemented) \end{aligned}$$

IBM Model 2: The Generative Process

To generate a French string *f* from an English string *e*

- *Step 1*: Pick an alignment with probability

$$p(a|e,m)=\prod_{i=1}^{m} q(a_i|i,l,m)$$

- *Step 2*: Given the alignment, pick the french words with probability

$$p(f|a,e,m)=\prod_{i=1}^{m} t(f_i|e_{a_i})$$

The final result:

$$p(f,a|e,m)=p(a|e,m)\,p(f|a,e,m)=\prod_{i=1}^{m} q(a_i|i,l,m)\,t(f_i|e_{a_i})$$

Recovering Alignments

- If we have estimates for the parameters *q* and *t*, we can easily recover the most likely alignment for any sentence pair

- Given a sentence pair $e_1, e_2, \ldots, e_l$ and $f_1, \ldots, f_m$, define

$$a_i = \text{argmax}_{a \in \{0, \ldots, l\}} q(a \mid i, l, m) t(f_i \mid e_a)$$

for $i = 1, \ldots, m$

Recovering Alignments

$$a_i = \text{argmax}_{a \in \{0,\ldots,l\}} q(a|i,l,m)\, t(f_i|e_a)$$

e = And the program has been implemented

f = Le programme a ete mis en application

Focus on computing a_3

NULL:	$q(0\|3, 6, 7)\ t(\text{a}\ \|\ \text{NULL})$
And:	$q(1\|3, 6, 7)\ t(\text{a}\ \|\ \text{And})$
the:	$q(2\|3, 6, 7)\ t(\text{a}\ \|\ \text{the})$
program:	$q(3\|3, 6, 7)\ t(\text{a}\ \|\ \text{program})$
has:	$q(4\|3, 6, 7)\ t(\text{a}\ \|\ \text{has})$
been:	$q(5\|3, 6, 7)\ t(\text{a}\ \|\ \text{been})$
implemented:	$q(6\|3, 6, 7)\ t(\text{a}\ \|\ \text{implemented})$

Choose as a_3 the best value

EM Training

- Till now we saw IBM Models 1 & 2
- The models need the parameters t and q
- Using the parameters, we can find the “best” alignment
- Now, how do we get these parameters?

The Parameter Estimation Problem

- Input: to the parameter estimation algorithm $(e^{(k)}, f^{(k)})$ for $k = 1 \ldots n$. Each $e^{(k)}$ is an English sentence, each $f^{(k)}$ is a French sentence
- Output: parameters $t(f|e)$ and $q(j \mid i, l, m)$
- Challenge: we do not have alignments on our training examples
- For example:

  $e^{(100)}$ = And the program has been implemented

  $f^{(100)}$ = Le programme a ete mis en application

Parameter Estimation if Alignments are Observed

- If alignments are observed in the training data:

  $e^{(100)}$ = And the program has been implemented

  $f^{(100)}$ = Le programme a ete mis en application

  $a^{(100)}$ = {2, 3, 4, 5, 6, 6, 6}

- Training data is $(e^{(k)}, f^{(k)}, a^{(k)})$ for $k = 1 \ldots n$. Each $e^{(k)}$ is an English sentence, each $f^{(k)}$ is a French sentence, each $a^{(k)}$ is the alignment.

- Maximum likelihood parameter estimation in this case is easy:

$$t_{ML}(f \mid e) = \frac{\text{Count}(e, f)}{\text{Count}(e)} \qquad q_{ML}(j \mid i, l, m) = \frac{\text{Count}(j \mid i, l, m)}{\text{Count}(i, l, m)}$$

Parameter Estimation if Alignments are Observed

- Maximum likelihood parameter estimation in this case is easy:

$$t_{ML}(f \mid e) = \frac{\text{Count}(e, f)}{\text{Count}(e)} \qquad q_{ML}(j \mid i, l, m) = \frac{\text{Count}(j \mid i, l, m)}{\text{Count}(i, l, m)}$$

- Example:

Number of times "the" and "le" were aligned

$$t_{ML}(\text{le} \mid \text{the}) = \frac{\text{Count}(\text{le, the})}{\text{Count}(\text{the})}$$

Number of times "the" was aligned to anything

Number of times position 1 (in French) was aligned with position 3 (in English) for l = 6 and m = 7

$$q_{ML}(3 \mid 1, 6, 7) = \frac{\text{Count}(3 \mid 1, 6, 7)}{\text{Count}(1, 6, 7)}$$

Number of times position 1 (in French) was aligned with anything for l = 6 and m = 7

Algorithm

Input: A training corpus $(f^{(k)}, e^{(k)}, a^{(k)})$ for $k = 1 \dots n$, where $f^{(k)} = f_1^{(k)} \dots f_{m_k}^{(k)}$, $e^{(k)} = e_1^{(k)} \dots e_{l_k}^{(k)}$, $a^{(k)} = a_1^{(k)} \dots a_{m_k}^{(k)}$.

the index of training example

$$\delta(k, i, j)$$

index of French word

index of English word

Algorithm:

- Set all counts $c(\dots) = 0$
- For $k = 1 \dots n$
  - For $i = 1 \dots m_k$, For $j = 0 \dots l_k$,

$$\begin{aligned} c(e_j^{(k)}, f_i^{(k)}) &\leftarrow c(e_j^{(k)}, f_i^{(k)}) + \delta(k, i, j) \\ c(e_j^{(k)}) &\leftarrow c(e_j^{(k)}) + \delta(k, i, j) \\ c(j|i, l, m) &\leftarrow c(j|i, l, m) + \delta(k, i, j) \\ c(i, l, m) &\leftarrow c(i, l, m) + \delta(k, i, j) \end{aligned}$$

where $\delta(k, i, j) = 1$ if $a_i^{(k)} = j$, 0 otherwise.

Output: $t_{ML}(f|e) = \frac{c(e,f)}{c(e)}$, $q_{ML}(j|i, l, m) = \frac{c(j|i,l,m)}{c(i,l,m)}$

Algorithm

Input: A training corpus $(f^{(k)}, e^{(k)}, a^{(k)})$ for $k = 1 \ldots n$, where $f^{(k)} = f_1^{(k)} \ldots f_{m_k}^{(k)}$, $e^{(k)} = e_1^{(k)} \ldots e_{l_k}^{(k)}$, $a^{(k)} = a_1^{(k)} \ldots a_{m_k}^{(k)}$.

Algorithm:

- Set all counts $c(\ldots) = 0$
- For $k = 1 \ldots n$
  - For $i = 1 \ldots m_k$, For $j = 0 \ldots l_k$,

$$
\begin{aligned}
c(e_j^{(k)}, f_i^{(k)}) &\leftarrow c(e_j^{(k)}, f_i^{(k)}) + \delta(k, i, j) \\
c(e_j^{(k)}) &\leftarrow c(e_j^{(k)}) + \delta(k, i, j) \\
c(j|i, l, m) &\leftarrow c(j|i, l, m) + \delta(k, i, j) \\
c(i, l, m) &\leftarrow c(i, l, m) + \delta(k, i, j)
\end{aligned}
$$

where $\delta(k, i, j) = 1$ if $a_i^{(k)} = j$, 0 otherwise.

Output: $t_{ML}(f|e) = \frac{c(e,f)}{c(e)}$, $q_{ML}(j|i, l, m) = \frac{c(j|i,l,m)}{c(i,l,m)}$

Example 90:

$e^{(90)}$ = the dog

$f^{(90)}$ = le chien

$a^{(90)} = \{1, 2\}$

$\delta(90, 1, 1) = 1$

$\delta(90, 2, 2) = 1$

$\delta(90, i, j) = 0$

c(the, le) ++

c(the) ++

$c(1|1,2,2)$ ++

$c(1,2,2)$ ++

c(dog, chien) ++

c(dog) ++

$c(2|2,2,2)$ ++

$c(2,2,2)$ ++

Parameter Estimation with the EM Algorithm

- The alignments are not observed in the training data:

  $e^{(100)}$ = And the program has been implemented

  $f^{(100)}$ = Le programme a ete mis en application

- Training data is $(e^{(k)}, f^{(k)})$ for $k = 1 \ldots n$. Each $e^{(k)}$ is an English sentence, each $f^{(k)}$ is a French sentence

- Related to previous algorithm, but with two key differences:
  - The algorithm is *iterative*. Start with some initial choice for *q* and *t*. At each iteration, compute some “counts” based on the data and current estimates. Re-estimate the parameters using the new counts
  - We use the following definition for $\delta(k, i, j)$ at each iteration:

$$\delta(k, i, j) = \frac{q(j|i, l_k, m_k)\, t(f_i^{(k)}|e_j^{(k)})}{\sum_{j=0}^{l_k} q(j|i, l_k, m_k)\, t(f_i^{(k)}|e_j^{(k)})}$$

EM Algorithm

Input: A training corpus $(f^{(k)}, e^{(k)})$ for $k = 1 \ldots n$, where $f^{(k)} = f_1^{(k)} \ldots f_{m_k}^{(k)}$, $e^{(k)} = e_1^{(k)} \ldots e_{l_k}^{(k)}$.

Initialization: Initialize $t(f|e)$ and $q(j|i, l, m)$ parameters (e.g., to random values).

10 – 20 iterations

EM Algorithm

For $s = 1 \dots S$

- Set all counts $c(\dots) = 0$
- For $k = 1 \dots n$
  - For $i = 1 \dots m_k$, For $j = 0 \dots l_k$

$$
\begin{aligned}
c(e_j^{(k)}, f_i^{(k)}) &\leftarrow c(e_j^{(k)}, f_i^{(k)}) + \delta(k,i,j) \\
c(e_j^{(k)}) &\leftarrow c(e_j^{(k)}) + \delta(k,i,j) \\
c(j|i,l,m) &\leftarrow c(j|i,l,m) + \delta(k,i,j) \\
c(i,l,m) &\leftarrow c(i,l,m) + \delta(k,i,j)
\end{aligned}
$$

Identical to previous algorithm

where

$$
\delta(k,i,j) = \frac{q(j|i,l_k,m_k)t(f_i^{(k)}|e_j^{(k)})}{\sum_{j=0}^{l_k} q(j|i,l_k,m_k)t(f_i^{(k)}|e_j^{(k)})}
$$

- Recalculate the parameters:

$$
t(f|e) = \frac{c(e,f)}{c(e)} \qquad q(j|i,l,m) = \frac{c(j|i,l,m)}{c(i,l,m)}
$$

EM Algorithm

$$\delta(k,i,j)=\frac{q(j|i,l_k,m_k)t(f_i^{(k)}|e_j^{(k)})}{\sum_{j=0}^{l_k} q(j|i,l_k,m_k)t(f_i^{(k)}|e_j^{(k)})}$$

$e^{(100)}$ = And the program has been implemented

$f^{(100)}$ = Le programme a ete mis en application

δ(100, 3, 0) = q(0|3,6,7) × t(a | NULL) / X

X = (q(0|3,6,7) × t(a | NULL) + q(1|3,6,7) × t(a | And) + q(2|3,6,7) × t(a | the) + …)

δ(100, 3, 1) = q(1|3,6,7) × t(a | And) / X

δ(100, 3, 2) = q(2|3,6,7) × t(a | the) / X

…

δ(100, 3, 6) = q(6|3,6,7) × t(a | implemented) / X

EM Algorithm

$$\delta(k,i,j)=\frac{q(j|i,l_k,m_k)\,t(f_i^{(k)}|e_j^{(k)})}{\sum_{j=0}^{l_k} q(j|i,l_k,m_k)\,t(f_i^{(k)}|e_j^{(k)})}$$

$e^{(100)}$ = And the program has been implemented

$f^{(100)}$ = Le programme a ete mis en application

$\sum_{j=0}^{l_k} \delta(k,i,j)=1$ They form a probability distribution over *j*

$$\delta(k,i,j)=P(a_i^{(k)}=j\,|\,e^{(k)},f^{(k)};t,q)$$

Probability that i^{th} French word is aligned with j^{th} English word under the current estimation parameter values

So we are trying to estimate the “best” alignment and use that because we don't have the actual alignment

EM Algorithm

For $s = 1 \dots S$

- Set all counts $c(\dots) = 0$
- For $k = 1 \dots n$
  - For $i = 1 \dots m_k$, For $j = 0 \dots l_k$

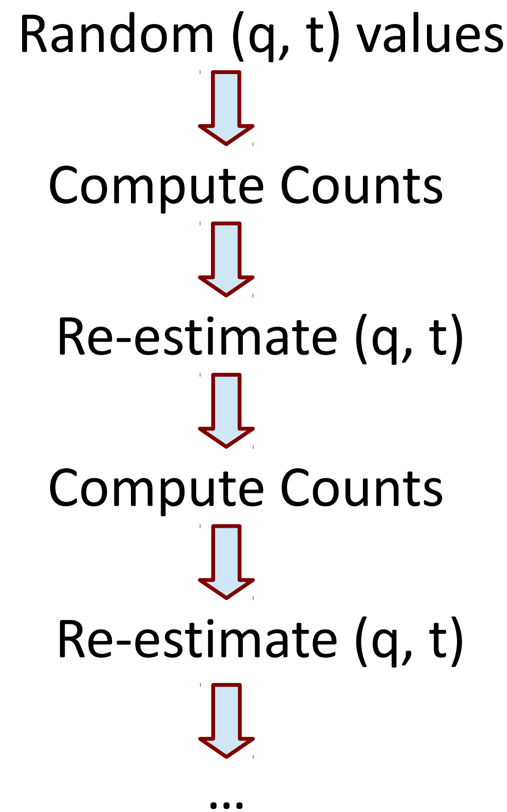
$$
\begin{aligned}
c(e_j^{(k)}, f_i^{(k)}) &\leftarrow c(e_j^{(k)}, f_i^{(k)}) + \delta(k,i,j) \\
c(e_j^{(k)}) &\leftarrow c(e_j^{(k)}) + \delta(k,i,j) \\
c(j|i,l,m) &\leftarrow c(j|i,l,m) + \delta(k,i,j) \\
c(i,l,m) &\leftarrow c(i,l,m) + \delta(k,i,j)
\end{aligned}
$$

  where

$$
\delta(k,i,j) = \frac{q(j|i,l_k,m_k)t(f_i^{(k)}|e_j^{(k)})}{\sum_{j=0}^{l_k} q(j|i,l_k,m_k)t(f_i^{(k)}|e_j^{(k)})}
$$

- Recalculate the parameters:

$$
t(f|e) = \frac{c(e,f)}{c(e)} \qquad q(j|i,l,m) = \frac{c(j|i,l,m)}{c(i,l,m)}
$$



Justification for the EM Algorithm

- Training data is $(e^{(k)}, f^{(k)})$ for $k = 1 \ldots n$. Each $e^{(k)}$ is an English sentence, each $f^{(k)}$ is a French sentence
- The log-likelihood function

$$L(q,t) = \sum_{k=1}^{n} \log p\left(f^{(k)} | e^{(k)}\right) = \sum_{k=1}^{n} \log \sum_{a} p\left(f^{(k)}, a | e^{(k)}\right)$$

- The maximum likelihood estimates:

$$\text{argmax}_{q,t} L(q,t)$$

- The EM algorithm converges to a *local* maximum of the likelihood function (it is not *convex*)

Summary

- Use alignments to simplify model
- Once parameters are estimated, we can recover the most probable alignment
- Iterative EM algorithm for estimating parameters
- IBM Model 2 no longer used for translation, but rather for recovering alignments, which are used in other MT methods

Recap

- IBM Model 1
- IBM Model 2
- Training of Models 1 and 2