

CMP462: Natural Language Processing



Lecture 09: Machine Translation I

Mohamed Alaa El-Dien Aly
Computer Engineering Department
Cairo University
Spring 2014

Agenda

- Introduction
- Challenges of Machine Translation (MT)
- Introduction to Statistical Machine Translation (SMT)
- IBM Models
- Training of IBM Models

Acknowledgment: Most slides adapted from Michael Collins NLP class on [Coursera](#).

Introduction

+You Search Images Maps Play YouTube News Gmail Drive Calendar More ▾

Sign in

Try a new browser with automatic translation. [Download Google Chrome](#) [Dismiss](#)

Translate

From: Arabic - detected ▾

To: English ▾

Translate

English Spanish French **Arabic - detected**

كما أوضح أن الإنفاق الاستهلاكي كان المحرك الرئيسي للاقتصاد الذي تضرر جراء عامين من الاضطرابات السياسية

وأشار إلى أن هناك شبه غياب للاستثمارات الأجنبية المباشرة في النصف الأول من السنة المالية، وأنه لتحقيق نمو اقتصادي بنسبة 7% تحتاج البلاد إلى معدل استثمار لا يقل عن 22%

English Spanish Arabic

He also explained that consumer spending was the main engine of the economy that has been hit by two years of political turmoil

He pointed out that there is a near absence of foreign direct investment (FDI) in the first half of the fiscal year, and that to achieve economic growth of 7% country needs investment rate of at least 22%

Challenges: Lexical Ambiguity

Example1:

book the flight reservar
read the book libro

Example2:

the box was in the pen
the pen was on the table

Example3:

kill a man matar
kill a process acabar

Challenges: Differing Word Orders

English word order is

subject-verb-object

Japanese word order is

subject-object-verb

English: IBM bought Lotus

Japanese: *IBM Lotus bought*

English: Sources said that IBM bought Lotus yesterday

Japanese: *Sources yesterday IBM Lotus bought*

Challenges: Syntactic Structure Not Preserved Across Translation

(Example from Dorr et al. 1999)

The bottle floated into the cave



La botella entro a la cuerva flotando
(the bottle entered the cave floating)

Challenges: Syntactic Ambiguity

(Example from Dorr et al. 1999)

John hit the dog with the stick



John golpeo el perro con el palo/que tenia el palo

(hit with the stick OR the dog with the stick)

Challenges: Pronoun Resolution

(Example from Dorr et al. 1999)

The computer outputs the data; it is fast.



La computadora imprime los datos; **es** rapida

The computer outputs the data; it is stored in ascii.



La computadora imprime los datos; **estan** almacenados en ascii

Introduction to Statistical MT

- Parallel corpora are available in several language pairs
- **Basic idea:** use a parallel corpus as a training set of translation examples
- Classic example: IBM work on French-English translation, using the Canadian Hansards. (1.7 million sentences of 30 words or less in length).
- Idea goes back to Warren Weaver (1949): suggested applying statistical and cryptanalytic techniques to translation.

Introduction to Statistical MT

... one naturally wonders if the problem of translation could conceivably be treated as a problem in cryptography. When I look at an article in Russian, I say: This is really written in English, but it has been coded in some strange symbols. I will now proceed to decode."

(Warren Weaver, 1949, in a letter to Norbert Wiener)

The Noisy Channel Model

- **Goal:** translation system from French to English
- Have a model $p(e | f)$ which estimates conditional probability of any English sentence e given the French sentence f . Use the training corpus to set the parameters.
- A Noisy Channel Model has two components:
 - $p(e)$ the language model
 - $p(f | e)$ the translation model

- Which gives us:

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

and

$$\operatorname{argmax}_e p(e | f) = \operatorname{argmax}_e p(e) p(f | e)$$

The Noisy Channel Model

- The language model $p(e)$ could be a trigram model, estimated from any data (parallel corpus *not* needed to estimate the parameters)
- The translation model $p(f|e)$ is trained from a parallel corpus of French/English pairs.
- Note:
 - The translation model is backwards!
 - The language model can make up for deficiencies of the translation model.
 - Later we'll talk about how to build $p(f|e)$
 - Decoding, i.e. finding $\operatorname{argmax}_e p(e) p(f|e)$ is also a challenging problem.

Example from Koehn and Knight tutorial

Translation from Spanish to English, candidate translations based on $p(\textit{Spanish} | \textit{English})$ alone:

Que hambre tengo yo

What hunger have $p(s|e) = 0.000014$

Hungry I am so $p(s|e) = 0.000001$

I am so hungry $p(s|e) = 0.0000015$

Have i that hunger $p(s|e) = 0.000020$

Example from Koehn and Knight tutorial

With $p(\text{Spanish} | \text{English}) \times p(\text{English})$:

Que hambre tengo yo

What hunger have $p(s|e)p(e) = 0.000014 \times 0.000001$

Hungry I am so $p(s|e)p(e) = 0.000001 \times 0.0000014$

I am so hungry $p(s|e)p(e) = 0.0000015 \times 0.0001$

Have i that hunger $p(s|e)p(e) = 0.000020 \times 0.00000098$

IBM Models

- The earliest statistical machine translation models
- Used to help us model $p(f|e)$
- We will discuss only two models:
 - IBM Model 1
 - IBM Model 2
- They can be trained without supervision from a parallel corpus

Alignments

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

English: The dog eats

French: Le chien mange

- English sentence e has l words e_1, \dots, e_l
- French sentence f has m words f_1, \dots, f_m
- An alignment a identifies the source of each *french* word
 - Above: $\{1, 2, 3\}$
- Formally, an alignment a is $\{a_1, \dots, a_m\}$ where each $a_i \in \{0, \dots, l\}$.
Why 0?
 - French words with no English equivalent (*NULL* word)
- How many possible alignments?
 - $(l+1)^m$

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

$\{1, 1, 1, 1, 1, 1, 1\}$

By-product: Most Likely Alignments

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

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

for any alignment a .

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

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

- 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



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

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

IBM Model 1: The Generative Process

To generate a French string f from an English string e

- *Step 1*: Pick an alignment with probability

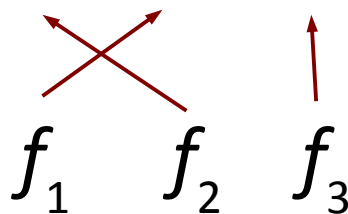
$$p(a|e, m) = \frac{1}{(l+1)^m}$$

- *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:

the dog eats



generate f_1 from $t(- | \text{dog})$, f_2 from $t(- | \text{the})$... 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	$t(- \text{position})$
position	vue	0.0169303	
position	point	0.0124795	
position	attitude	0.0108907	

... 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

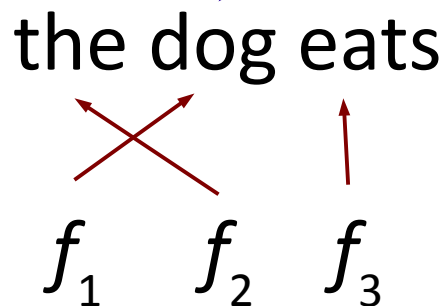
- Define $p(a|e, m) = \prod_{i=1}^m q(a_i|i, l, m)$

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

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

- Example:

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



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

IBM Model 2

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

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

Probability that 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 $a = \{a_1, \dots, a_m\}$

- 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

$$l = 6$$

$$m = 7$$

$e =$ And the program has been implemented

$f =$ Le programme a ete mis en application

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

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

Another Example

$$l = 6$$

$$m = 7$$

$e =$ And the program has been implemented

$f =$ Le programme a ete mis en application

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

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

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, \dots, e_l and f_1, \dots, f_m , define

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

for $i = 1, \dots, m$

- Recover the best alignment for each word *independently*

Recovering Alignments

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

e = And the program has been implemented

f = Le programme aete mis en application

Focus on computing a_3

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

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 $k = 1 \dots n$. Each $e^{(k)}$ is an English sentence, each $f^{(k)}$ is a French sentence
- **Output:** parameters $t(f|e)$ and $q(j | i, l, m)$
- **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 $k = 1 \dots n$. 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:

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

Parameter Estimation if Alignments are Observed

- Maximum likelihood parameter estimation in this case is easy:

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

- Example:

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

Number of times “the” and “le” were aligned

Number of times “the” was aligned to anything

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

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

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
 $f^{(k)} = f_1^{(k)} \dots f_{m_k}^{(k)}$, $e^{(k)} = e_1^{(k)} \dots e_{l_k}^{(k)}$, $a^{(k)} = a_1^{(k)} \dots a_{m_k}^{(k)}$.

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

the index of training example

$\delta(k, i, j)$

index of French word

index of English word

Algorithm

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

Algorithm:

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

$$c(e_j^{(k)}, f_i^{(k)}) \leftarrow c(e_j^{(k)}, f_i^{(k)}) + \delta(k, i, j)$$

$$c(e_j^{(k)}) \leftarrow c(e_j^{(k)}) + \delta(k, i, j)$$

$$c(j|i, l, m) \leftarrow c(j|i, l, m) + \delta(k, i, j)$$

$$c(i, l, m) \leftarrow c(i, l, m) + \delta(k, i, j)$$

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

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

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

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

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

$c(\text{the, le}) ++$

$c(\text{the}) ++$

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

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

$c(\text{dog, chien}) ++$

$c(\text{dog}) ++$

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

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

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 $k = 1 \dots n$. 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 $k = 1 \dots n$, where
 $f^{(k)} = f_1^{(k)} \dots f_{m_k}^{(k)}$, $e^{(k)} = e_1^{(k)} \dots e_{l_k}^{(k)}$.

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 $j = 0 \dots l_k$

$$c(e_j^{(k)}, f_i^{(k)}) \leftarrow c(e_j^{(k)}, f_i^{(k)}) + \delta(k, i, j)$$

$$c(e_j^{(k)}) \leftarrow c(e_j^{(k)}) + \delta(k, i, j)$$

$$c(j|i, l, m) \leftarrow c(j|i, l, m) + \delta(k, i, j)$$

$$c(i, l, m) \leftarrow c(i, l, m) + \delta(k, i, j)$$

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:

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

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

$$\delta(100, 3, 0) = q(0|3,6,7) \times t(a | \text{NULL}) / X$$

$$X = (q(0|3,6,7) \times t(a | \text{NULL}) + q(1|3,6,7) \times t(a | \text{And}) + q(2|3,6,7) \times t(a | \text{the}) + \dots)$$

$$\delta(100, 3, 1) = q(1|3,6,7) \times t(a | \text{And}) / X$$

$$\delta(100, 3, 2) = q(2|3,6,7) \times t(a | \text{the}) / X$$

...

$$\delta(100, 3, 6) = q(6|3,6,7) \times t(a | \text{implemented}) / X$$

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

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

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

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 $j = 0 \dots l_k$

$$c(e_j^{(k)}, f_i^{(k)}) \leftarrow c(e_j^{(k)}, f_i^{(k)}) + \delta(k, i, j)$$

$$c(e_j^{(k)}) \leftarrow c(e_j^{(k)}) + \delta(k, i, j)$$

$$c(j|i, l, m) \leftarrow c(j|i, l, m) + \delta(k, i, j)$$

$$c(i, l, m) \leftarrow c(i, l, m) + \delta(k, i, j)$$

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:

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

Random (q, t) values



Compute Counts



Re-estimate (q, t)



Compute Counts



Re-estimate (q, t)



...

Justification for the EM Algorithm

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

- The log-likelihood function

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

- The maximum likelihood estimates:

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

- 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

- Introduction
- Challenges of Machine Translation (MT)
- Introduction to Statistical Machine Translation (SMT)
- IBM Models
- Training of IBM Models