Expectation Maximization

Machine Translation as a task

For Machine Translation to happen, one must put in place processes for:

1. Word translation - what do the words map to; could be to more than one word in either direction
2. Translation alignment - Movement of translated words to their correct positions in the target sentence
3. Word fertility management or phrase alignment management

Let’s look at some examples

Examples

E: Peter slept early today
H: पीटर आज जल्दी सोया
HT: piitar aaj jaldii soyaa
HG: Peter today early slept

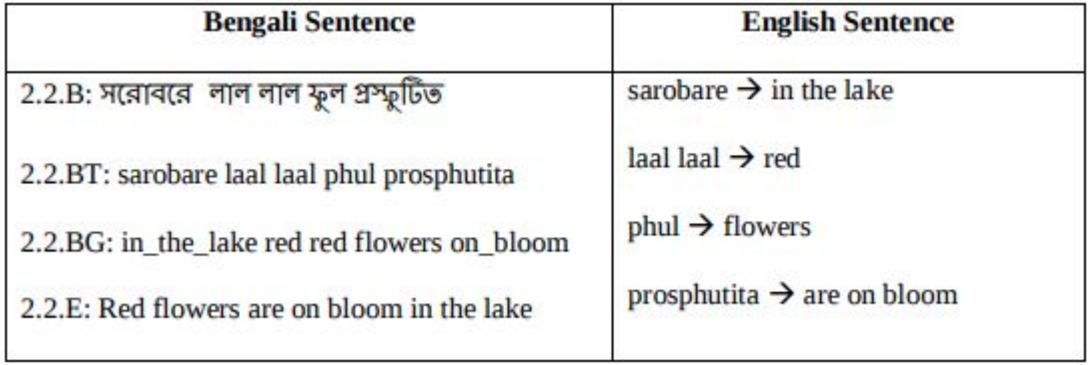
L -> Language [Eg: Hindi H, English E]
LT -> Its transliteration
LG -> Its gloss, ie word to word translation

In the above example, we can see that the mapping is as follows
a(1)=1, a(2)=4, a(3)=3, a(4)=2

The translation task is almost over when words are translated and the translated words are positioned.

Q: *What if a word from either side of the translation can map to more than one word or even to nothing on the other side?*

Examples



Bengali Sentence	English Sentence
2.2.B: সরোবরে লাল লাল ফুল প্রস্ফুটিত 2.2.BT: sarobare laal laal phul prosphutita 2.2.BG: in_the_lake red red flowers on_bloom 2.2.E: Red flowers are on bloom in the lake	sarobare → in the lake laal laal → red phul → flowers prosphutita → are on bloom

This many-to-many alignment is the foundation for the so called Phrase Based Statistical machine translation (PB-SMT) and will be discussed later.

If the many-to-many mappings are not allowed, NULL mappings should be accepted.

An example for the above would be ->

“Peter went to school” -> “piitar paathshaalaa gayaa”
where “to” -> NULL i.e., ‘to’ maps to nothing in Hindi.

The phenomenon of one word mapping to multiple words on the other side is called “fertility” which will be discussed under IBM models of word alignment.

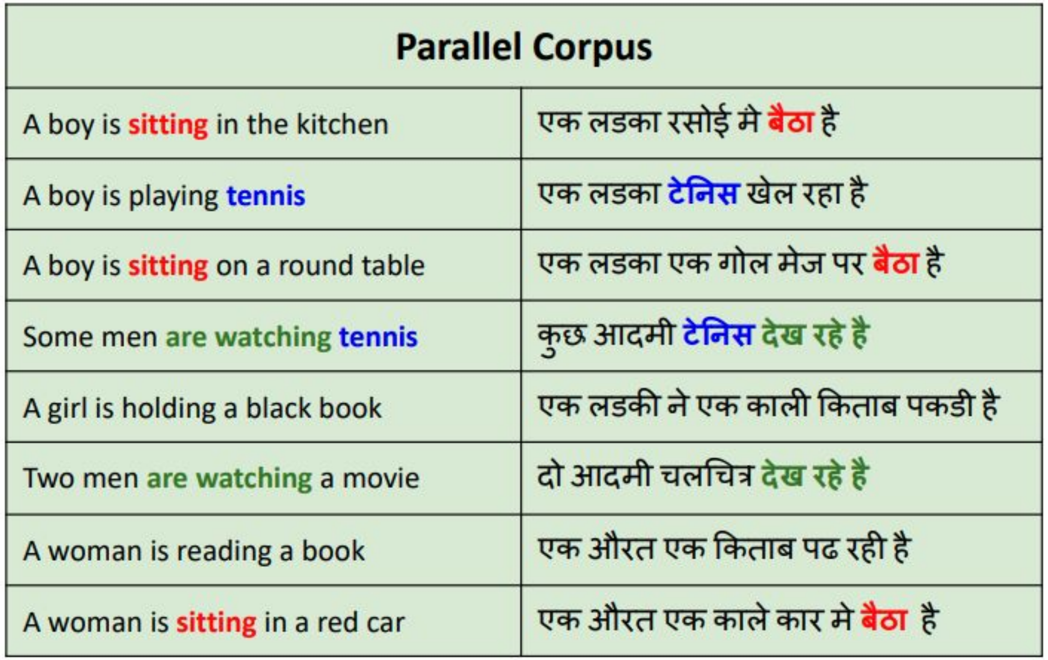
Key Idea

Co-occurrence of translated words

Words which occur together in the parallel sentence are likely to be translations (higher translational probability)

If we know the alignments, we can compute the translational probabilities

But, we can find the best alignment only if we know the word translation probabilities



Parallel Corpus	
A boy is **sitting** in the kitchen	एक लडका रसोई मे **बैठा** है
A boy is playing **tennis**	एक लडका **टेनिस** खेल रहा है
A boy is **sitting** on a round table	एक लडका एक गोल मेज पर **बैठा** है
Some men **are watching** **tennis**	कुछ आदमी **टेनिस** **देख रहे है**
A girl is holding a black book	एक लडकी ने एक काली किताब पकडी है
Two men **are watching** a movie	दो आदमी चलचित्र **देख रहे है**
A woman is reading a book	एक औरत एक किताब पढ रही है
A woman is **sitting** in a red car	एक औरत एक काले कार मे **बैठा** है

Why EM?

As we discussed, the steps for MT are
A. Word Translation
B. Translation Alignment
C. Word Fertility (or) Phrase Alignment Management

Steps (A) and (B) above are mutually supportive; knowing one, one knows the other.

Given the parallel sentence pair and word translations, one can get the alignments.
Similarly, given the parallel sentence pair and the alignments, one can get the translations.

The best alignment is the one that maximizes the sentence translation probability

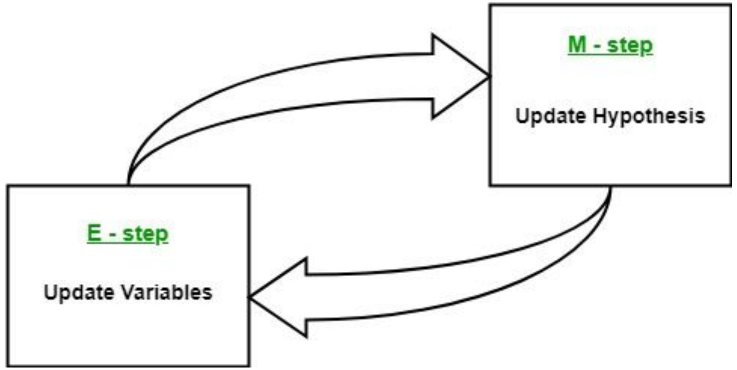
Such two way absence of knowledge, ie chicken-and-egg problem, is symptomatic of application of Expectation Maximization (EM).

Build up to the formulae based on EM

- A data or observation likelihood expression is set up, assuming an appropriate distribution.
- The parameters of the distribution are the quantities of interest (in our case alignment probabilities).
- Hidden variables are assumed for mathematical simplicity of the likelihood expression.
- The parameters and the hidden variables are estimated iteratively, starting with initial values of parameters.
- The iteration step to estimate the parameters is called the **M-step or maximization step.**
- The iteration step to estimate the hidden variables is called the **E-step or expectation step.**

Expectation Maximization Algorithm

1. Initialize model parameters (e.g. uniform)
2. Assign probabilities to the missing data
3. Estimate model parameters from completed data
4. Iterate steps 2–3 until convergence



Examples of EM

	HMM	**MT**	**Coin Toss**
X (observed)	Sentences	Parallel Data	Head-tail Sequences
Y (hidden)	State Sequences	Word Alignment	Coin id Sequences
θ	a_{ij}, b_{ijk}	t(f|e), d(a_j|j, l, m), …	p1, p2, λ
Algorithm	Forward - backward	IBM Models	N/A

Coin-toss Experiment

Situation-1: Throw of a single coin

- Given a biased coin, P(H) = P, therefore P(T) = 1 – P
- Experiment is in Bernoulli Trial Setting (It has only two outcomes- Head or Tail)
- The experiment is repeated N times.
- Problem: Given an Observation Sequence ‘**D**’ of H and T, what is the likelihood of D, ie P(D)?

- Indicator Variable[1] is used to help in calculating the likelihood.
- Indicator Variable x_i is defined as:

$$x_i = \begin{cases} 1 & \textit{if } i^{th} \textit{ observation} = H, \\ 0 & \textit{otherwise} \end{cases} \tag{1}$$

Hence,

$$P(D|Q) = \prod_{i=1}^{N} (P^{x_i}(1 - P)^{1-x_i}) \tag{2}$$

where $P = \frac{\sum_{i=1}^{N} x_i}{N}$, which is total number of occurrences of heads divided by total number of experiments (coin tosses).

[1] https://en.wikipedia.org/wiki/Dummy_variable_(statistics)

Situation-2: Throw of two coins

- Now there are 3 parameters, probabilities, $P(H|Coin_1) = P_1$ and $P(H|Coin_2) = P_2$, of heads of the two coins, the probability $P(Coin_1) = P$ of choosing the first coin, and implying,
  $P(T\ |Coin_1) = 1 - P_1$; $P(T\ |Coin_2) = 1 - P_2$;and $P(Coin_2) = 1 - P$
- Experiment is in Bernoulli Trial Setting (Choice of coins has two outcomes - $Coin_1$ or $Coin_2$. Toss too has two outcomes – Head or Tail)
- Same as single coin toss, we have N tosses and observations of heads and tails.
- However, we do not know which observation comes from which coin.

- An indicator variable z_i is introduced to capture coin choice

$$z_i = \begin{cases} 1 & \text{if } i^{th} \text{ coin tossed is Coin1} \\ 0 & \text{otherwise} \end{cases}$$

- This variable is hidden, i.e., we do not know its values. However, without it, the likelihood expression would have been very cumbersome.
- Problem: Given an Observation Sequence D of H, T, what is likelihood of D, that is P(D)?

Complete Data of observations to calculate likelihood looks like:

$D_{complete} = <x_1, z_1>, <x_2, z_2>, \ldots <x_N, z_N>$

Data Likelihood, D = $P_\theta(X)$

And, $P_\theta(X,Z)=\prod_{i=1}^{N}\left(pp_1^{x_i}(1-p_1)^{1-x_i}\right)^{z_i}\left((1-p)p_2^{x_i}(1-p_2)^{1-x_i}\right)^{1-z_i}$,

$$\text{s.t. } x_i = 1 \text{ or } 0,\ z_i = 1 \text{ or } 0,\ \theta =< p, p_1, p_2 >$$

The choice of each coin is a Bernoulli trial. After a choice is made, we have another Bernoulli trial for the outcome of the toss. For mathematical convenience we work with expectation of log likelihood:

$$\begin{aligned} E(LL(D)) = \sum_{i=1}^{N} [&E(z_i)(\log p + x_i \log p_1 + (1-x_i)\log(1-p_1)) \\ &+ (1-E(z_i))(\log(1-p) + x_i \log p_2 + (1-x_i)\log(1-p_2))] \end{aligned}$$

$$X :< x_1, x_2, x_3, \ldots, x_{N-1}, x_N > \ \textit{and}\ \ Z :< z_1, z_2, z_3, \ldots, z_{N-1}, z_N >$$

Note: How has z_i turned into expectation of z_i, ie $E(z_i)$? The reasoning is as follows. We can introduce the hidden variable Z only through marginalization: $P_\theta(X) = \sum_Z P_\theta(X,Z)$

We would like to work with log(Pθ(X)). However, then we will have a Σ expression inside log.

This is cumbersome. Fortunately log is a concave function, so that

$$\log\left(\sum_{i=1}^{n} \lambda_i y_i\right) \geq \sum_{i=1}^{n} \lambda_i \log(y_i), \quad \textit{with} \sum_{i=1}^{n} \lambda_i = 1$$

This is the application of **Jensen's inequality**, which is briefly explained in the next slide.

Quick Recap: A function is concave if it's second derivative is negative.

Jensen’s Inequality

- For any convex function f (x), $f(\lambda_1 * x_1 + \lambda_2 * x_2) \leq \lambda_1 * f(x_1) + \lambda_2 * f(x_2)$
- For any concave function f (x), $f(\lambda_1 * x_1 + \lambda_2 * x_2) \geq \lambda_1 * f(x_1) + \lambda_2 * f(x_2)$

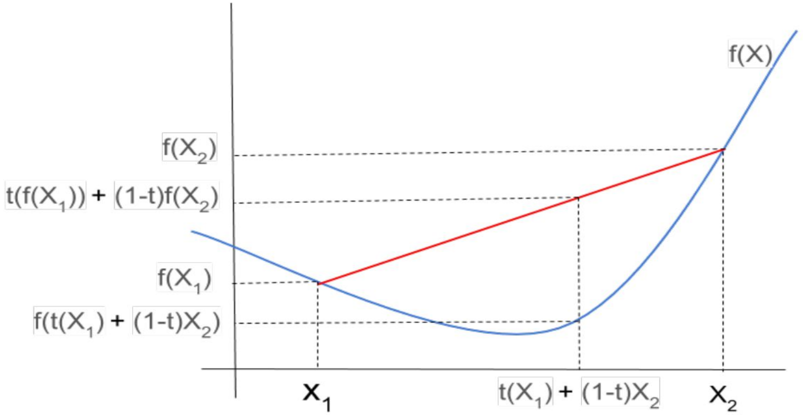


Figure: Jensen’s Inequality for a convex function $f(X)$

Expectation Step:

Thus **log** can move inside past **Σ**, provided we can find appropriate **λ**s. In this case **λ**s are constructed from **P_θ(X,Z)** at particular values of **Z** and **θ**. A bit of manipulation with these probability values leads to:

$$
\begin{aligned}
&LL(D) = \log likelihoood\ \ of\ \ data \\
&= \log(P_\theta(X)) \\
&= \log(\sum_z P_\theta(X,Z)) \\
&\geq E_{Z,\theta}(\log(P(X,Z)))
\end{aligned}
$$

where **$E_{Z,\theta}$(log(P(X,Z))** is the expectation of log likelihood of the data at particular values of **Z** and **θ**.

This gives rise to the above log likelihood expression (**θ** is dropped, being clear from the context).

Maximization Step:
We want to maximize likelihood, based on the parameters. Hence, we differentiate wrt P, P_1 and P_2 and equate them to 0. We get:

$$p_1 = \frac{\sum_{i=1}^{N} E(z_i)x_i}{\sum_{i=1}^{N} E(z_i)}$$

$$p_2 = \frac{M - \sum_{i=1}^{N} E(z_i)x_i}{N - \sum_{i=1}^{N} E(z_i)}, \text{ } M\text{= observed no. of heads}$$

$$p = \frac{\sum_{i=1}^{N} E(z_i)x_i}{N}$$

$$E(z_i)$$
$$= P(z_i = 1 \mid x = x_i) = P(z_i = 1).P(x = x_i \mid z_i = 1) / P(x = x_i)$$
$$= \frac{PP_1^{x_i}(1-P_1)^{(1-x_i)}}{PP_1^{x_i}(1-P_1)^{(1-x_i)} + (1-P)P_2^{x_i}(1-P_2)^{(1-x_i)}}$$

Generalization:

Throw of more than 1 ‘something’, where that ‘something’ has more than 1 outcome.

This gives rise to a multinomial which is extremely useful in many NLP and ML situations.

- Observation sequence: N observations, with each observation having L outcomes such that anyone of the L outcomes is possible for each of the observation, i.e., $\sum_{k=1}^{L} x_{ik} = 1$ since each x_{ik}=1/0, and one and only one of them is 1.

$$D:(x_{11},x_{12},x_{13},\dots,x_{1L}),(x_{21},x_{22},x_{23},\dots,x_{2L}),\dots,(x_{N1},x_{N2},x_{N3},\dots,x_{NL})$$

- Hidden variable for M sources

$$Z : (z_{11}, z_{12}, z_{13}, \ldots, z_{1M}), (z_{21}, z_{22}, z_{23}, \ldots, z_{2M}), \ldots, (z_{N1}, z_{N2}, z_{N3}, \ldots, z_{NM})$$

Since one and only one source is chosen for each observation, i.e., $\sum_{j=1}^{M} z_{ij}$nce each z_{ij}= 1/0, and one and only one of them is 1.

- Parameters θ:
  π_j: Probability of choosing source j
  p_{jk}: Probability of observing ´jth outcome from ꞉kth source

Assuming the observations to be i.i.d. (independently and identically distributed, in this case- multinomial distribution), the likelihood, the log likelihood and the expectation of the log likelihood are respectively:

$$L(D;\theta) = \prod_{i=1}^{N} \prod_{j=1}^{M} \left(\pi_j \prod_{k=1}^{L} (p_{jk}^{x_{ik}})\right)^{z_{ij}} \qquad (2.1)$$

$$LL(D;\theta) = \sum_{i=1}^{N} \sum_{j=1}^{M} z_{ij} \left(\pi_j + \sum_{k=1}^{L} (x_{ik} \log p_{jk})\right) \qquad (2.2)$$

$$E(LL(D;\theta)) = \sum_{i=1}^{N} \sum_{j=1}^{M} E(z_{ij}) \left(\pi_j + \sum_{k=1}^{L} (x_{ik} \log p_{jk})\right) \qquad (2.3)$$

Maximize (2.3) subject to the constraints:

$$\sum_{j=1}^{M} \pi_j = 1 \qquad (2.4)$$

$$\sum_{k=1}^{L} P_{jk} = 1,\ j = 1,\dots,M \qquad (2.5)$$

(2.4) is true since one of the sources is certain to be chosen. (2.5) is true, since given a source, one of the outcomes is certain.

Introducing one Lagrangian for (2.4) and M Lagrangians for each of the M sources as per (2.5), the dual of (2.3) to be optimized is:

$$\begin{aligned} Q(D;\theta) = \sum_{i=1}^{N}\sum_{j=1}^{M} E(z_{ij})\left(\pi_j + \sum_{k=1}^{L}(x_{ik}\log p_{jk})\right) - \\ \alpha\left(\sum_{j=1}^{M}\pi_j - 1\right) - \sum_{j=1}^{M}\beta_j\left(\sum_{k=1}^{L}P_{jk} - 1\right) \end{aligned}$$

We want to maximize likelihood, based on the parameters. Hence, we differentiate wrt π_j and p_{jk}, and equate them to 0. We get the following:

E-Step:

$$E(z_{ij}) = \frac{\pi_j \prod_{k=1}^{L} (p_{jk}^{x_{ik}})}{\sum_{j=1}^{M} \pi_j \prod_{k=1}^{L} (p_{jk}^{x_{ik}})}$$

M-Step:

$$\pi_j = \frac{\sum_{i=1}^{N} E(z_{ij})}{\sum_{j=1}^{M} \sum_{i=1}^{N} E(z_{ij})}$$

$$p_{jk} = \frac{\sum_{i=1}^{N} E(z_{ij}) x_{ik}}{\sum_{i=1}^{N} E(z_{ij})}$$

Derivation of alignment probabilities

Key Notations:

English vocabulary: V_E
French vocabulary: V_F
No. of sentence pairs (observations): $\dot{S}$
Data D which consists of $\dot{S}$ pairs of parallel sentences looks like:

$$e_1^1, e_2^1, e_3^1, \ldots, e_{l_1}^1 \Leftrightarrow f_1^1, f_2^1, f_3^1, \ldots, f_{m_1}^1 \qquad \text{- (pair-1: } \boldsymbol{E^1}, \boldsymbol{F^1}\text{)}$$

$$e_1^2, e_2^2, e_3^2, \ldots, e_{l_1}^2 \Leftrightarrow f_1^2, f_2^2, f_3^2, \ldots, f_{m_2}^2 \qquad \text{- (pair-2: } \boldsymbol{E^2}, \boldsymbol{F^2}\text{)}$$

$$\ldots$$

$$e_1^s, e_2^s, e_3^s, \ldots, e_{l_1}^s \Leftrightarrow f_1^s, f_2^s, f_3^s, \ldots, f_{m_s}^s \qquad \text{- (}s^{th}\text{ pair: } \boldsymbol{E^s}, \boldsymbol{F^s}\text{)}$$

$$\ldots$$

$$e_1^S, e_2^S, e_3^S, \ldots, e_{l_1}^S \Leftrightarrow f_1^S, f_2^S, f_3^S, \ldots, f_{m_S}^S \qquad \text{- (last pair, }S^{th}\text{ pair: } \boldsymbol{E^S}, \boldsymbol{F^S}\text{)}$$

No. words on English side in s^{th} sentence: l_s

No. words on French side in s^{th} sentence: m_s

$\text{index}_E(e_i)$ = Index of English word e_i in English vocabulary/dictionary

$\text{index}_F(f_j)$ = Index of French word fj in French vocabulary/dictionary

Hidden Variables (A; the alignment variables):

Total no. of hidden variables = $\sum_{s=1}^{S} l^s m^s$
where each hidden variable is as follows:

a_{pq}^{s} = 1, if in sth sentence, pth English word is mapped to qth French word
= 0, otherwise

Parameters (θ) :

Total no. of parameters= $|V_E| \times |V_F|$, where each parameter is as follows:
P_{ij}= Probability that ith word in English dictionary is mapped to jth word in foreign language dictionary, and this is the parameter set θ

Data Likelihood L(D; θ) = $\prod_{s=1}^{S} P(f^s \mid e^s)$

Data Likelihood L(D; θ), marginalized over A :

$$L(D;\theta) = \sum_{A} L(D, A;\theta)$$

$$and,$$

$$L(D, A;\theta) = \prod_{s=1}^{S} \prod_{m=1}^{m^s} \prod_{l=1}^{l^s} (P_{index_E(e_p^s), index_F(f_q^s)})^{a_{pq}^s}$$

Marginalized Data Log-Likelihood LL(D,A; θ) :

$$LL(D, A;\theta) = \sum_{s=1}^{S} \sum_{q=1}^{m^s} \sum_{p=1}^{l^s} \log(P_{index_E(e_p^s), index_F(f_q^s)})^{a_{pq}^s}$$

Expectation of data log likelihood E(LL(D; θ)) :

$$E(LL(D, A;\theta)) = \sum_{s=1}^{S} \sum_{q=1}^{m^s} \sum_{p=1}^{l^s} E(a_{pq}^s) \log(P_{index_E(e_p^s), index_F(f_q^s)})$$

Need to find parameter values such that E(LL(D;θ) is maximized w.r.t. the following $|V_E|$ constraints:

$$(\sum_{j=1}^{|V_F|} P_{ij} = 1), \forall i$$

Introduce Lagrangian for the constraint. Let the Lagrange multiplier corresponding to the ith English word's constraint be λ_i

$$E(LL(D, A;\theta)) = \sum_{s=1}^{S} \sum_{q=1}^{m^s} \sum_{p=1}^{l^s} E(a_{pq}^s) \log(P_{index_E(e_p^s), index_F(f_q^s)}) - \sum_{i=1}^{|V_E|} \lambda_i (\sum_{j=1}^{|V_F|} P_{ij} = 1)$$

Differentiating wrt P_{ij}, and by using the above constraint, we get:

$$P_{ij} = \frac{\sum_{s=1}^{S} \sum_{q=1}^{m^s} \sum_{p=1}^{l^s} \delta_{index_E(e_p^s),i}\, \delta_{index_F(f_q^s),j}\, E(a_{pq}^s)}{\sum_{j=1}^{|V_F|} \sum_{s=1}^{S} \sum_{q=1}^{m^s} \sum_{p=1}^{l^s} \delta_{index_E(e_p^s),i}\, \delta_{index_F(f_q^s),j}\, E(a_{pq}^s)} \qquad -M-step$$

Expectation-step:

$E(a^s_{pq})$ is the expectation that given the sth parallel sentence the pth word from the English side aligns with the qth word in the foreign side.

By definition, expectation of a random variable is the sum of product of the values of the r.v. and the corresponding probability. The random variable a^s_{pq} takes values 0 or 1.

Hence, we have $E(a^s_{pq}) = P(a^s_{pq} \mid e^s, f^s) = P(e^s_p \leftrightarrow f^s_q \mid e^s \leftrightarrow f^s)$

$$E(a^s_{pq}) = \frac{P_{index_E(e^s_p),index_F(f^s_q)}}{\sum_{x=1}^{m^s} P_{index_E(e^s_p),index_F(f^s_x)}}$$

where $P_{index_E(e^s_p),index_F(f^s_q)}$ is the probability of mapping between the dictionary entry that matches e^s_p and

the dictionary entry that matches f^s_q.

Summarizing:

The alignment algorithm will first initialize the Pij values. From these it will get the z^s_{pq} values for s, p,q through the E-step.

This will then update the P_{ij} values through the M-step. These new values of P_{ij}s will be used to get new E^s_{pq} values.

M-Step:

$$P_{ij} = \frac{\sum_{s=1}^{S}\sum_{q=1}^{m^s}\sum_{p=1}^{l^s}\delta_{index_E(e_p^s),i}\,\delta_{index_F(f_q^s),j}\,E(a_{pq}^s)}{\sum_{j=1}^{|V_F|}\sum_{s=1}^{S}\sum_{q=1}^{m^s}\sum_{p=1}^{l^s}\delta_{index_E(e_p^s),i}\,\delta_{index_F(f_q^s),j}\,E(a_{pq}^s)}$$

E-Step:

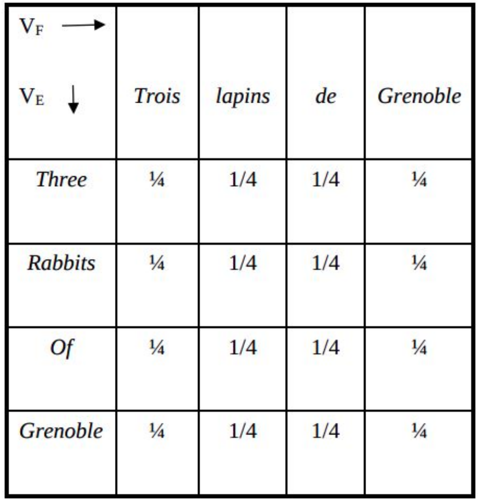
$$E(a_{pq}^s) = \frac{P_{index_E(e_p^s),index_F(f_q^s)}}{\sum_{x=1}^{m^s}P_{index_E(e_p^s),index_F(f_x^s)}}$$

Such alternations between E-step and M-Step will continue, eventually converging to stable P_{ij} values. These values will correspond to the global minimum, since the data likelihood expression is convex.

Explanation with an example

Initialization and Iteration-1 of EM:
We start with uniform probability values for the alignments. In absence of any information other than the parallel corpora, all alignments are equally likely. *Three*, for example, can map to any one of *trois*, *lapins*, *de* and *Grenoble* with equal probability.

So, the Initial alignment probabilities; with Average entropy=2



$V_F \rightarrow$ $V_E \downarrow$	*Trois*	*lapins*	*de*	*Grenoble*
Three	¼	1/4	1/4	¼
Rabbits	¼	1/4	1/4	¼
Of	¼	1/4	1/4	¼
Grenoble	¼	1/4	1/4	¼

From this we get the “expected counts” of alignments.

For the expected count of three *trois* alignment from the first parallel sentence, we note the count of three and *trois* in the sentence pair and weigh the count by current Pr(*trois*|*three*) normalized.

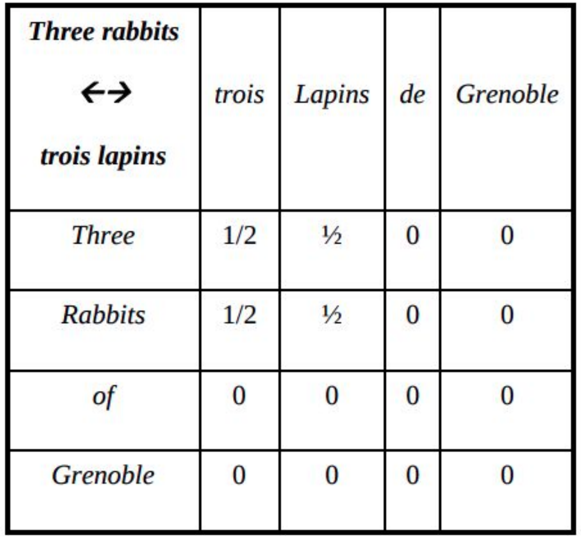
$$expected\_count\ [three \leftarrow\rightarrow trois;\ (three\ rabbits) \leftarrow\rightarrow (trois\ lapins)]$$

$$\begin{aligned}
&= \frac{\Pr(trois \mid three)}{\Pr(trois \mid three) + \Pr(lapins \mid three)} * (\#three) * (\#trois) \\
&= \frac{1/4}{1/4 + 1/4} * 1 * 1 \\
&= 1/2
\end{aligned}$$

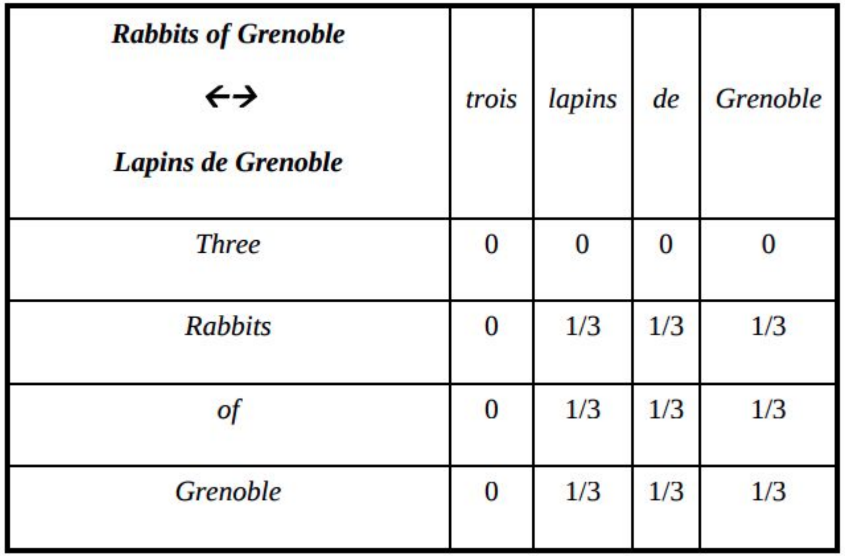
Three and *trois* appear once each in the first parallel sentence pair.

The current Pr(*troi*|*three*) value is ¼. In the parallel sentence pair “three rabbits trois lapins”, *three* can map to *lapins* also. Hence the weightage factor is ½.

Carrying out the above procedure, we get two “count” tables for the two sentences. This completes iteration-1.



Three rabbits ←→ trois lapins	*trois*	*Lapins*	*de*	*Grenoble*
Three	1/2	½	0	0
Rabbits	1/2	½	0	0
of	0	0	0	0
Grenoble	0	0	0	0



Rabbits of Grenoble ←→ Lapins de Grenoble	*trois*	*lapins*	*de*	*Grenoble*
Three	0	0	0	0
Rabbits	0	1/3	1/3	1/3
of	0	1/3	1/3	1/3
Grenoble	0	1/3	1/3	1/3

Expected counts of mappings in “three rabbits ←→ troi lapins” ınts of mappings in “rabbits of grenoble lapins de Grenoble”

Iteration 2:

From these expected counts we get the “revised” probabilities Pr(trois|three), Pr(lapins|three), Pr(lapins|rabbit) etc.

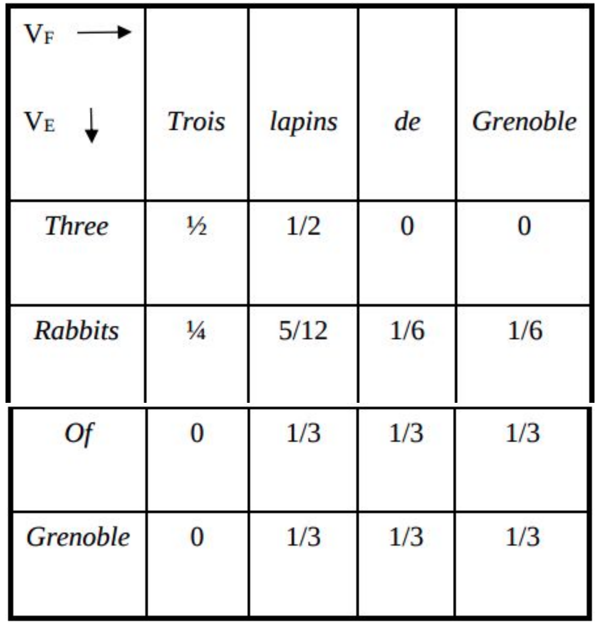
Since the current value of Pr(lapins|rabbit) is ¼=0.25, it will be interesting to see what it gets updated to:

$$
\begin{aligned}
& P_{revised}(lapins \mid rabbit) \\
&= \frac{count(rabbits \rightarrow lapins \text{ in the corpus})}{count(rabbits \rightarrow anything\_else \text{ in the corpus})} \\
&= \frac{1/2+1/3}{(1/2+1/2)+(1/3+1/3+1/3)} \\
&= \frac{5}{12} \approx 0.4
\end{aligned}
$$

The numerator is the total “count” in the corpus of rabbits lapins which from tables above is (1/2+1/3). The denominator is the total count of rabbits X mapping, where X is any other word. That count from the first parallel sentence is (1/2+1/2) and from the second sentence is (1/3+1/3+1/3).

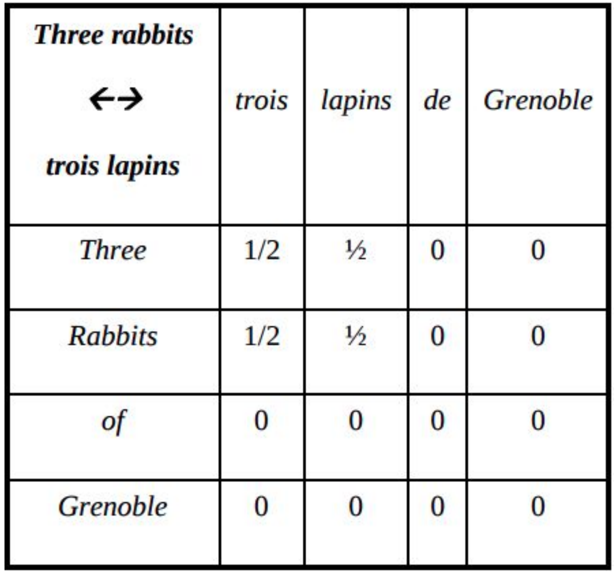
Thus after the first iteration the probability P(lapins|rabbit) seems to be moving in the right direction. The value has increased from 0.25 to 0.4.

Revised alignment probabilities after iteration-1; average entropy= 1.9

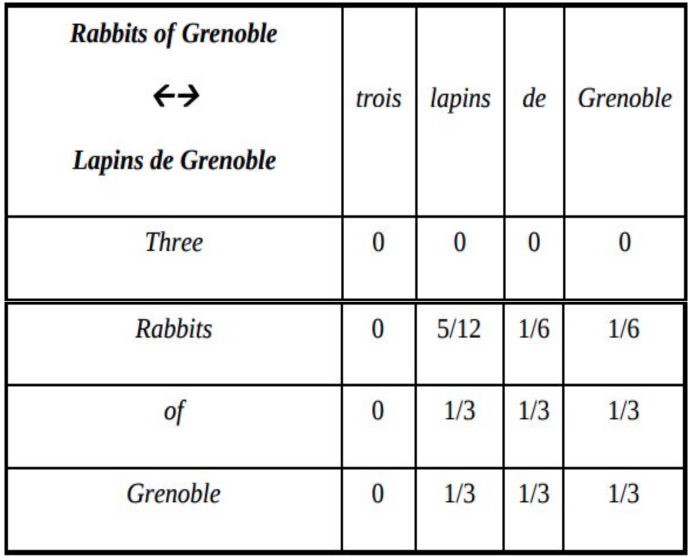


$V_F \longrightarrow$ $V_E \downarrow$	*Trois*	*lapins*	*de*	*Grenoble*
Three	½	1/2	0	0
Rabbits	¼	5/12	1/6	1/6
Of	0	1/3	1/3	1/3
Grenoble	0	1/3	1/3	1/3

Revising the counts now:



Three rabbits* ←→ *trois lapins	*trois*	*lapins*	*de*	*Grenoble*
Three	1/2	½	0	0
Rabbits	1/2	½	0	0
of	0	0	0	0
Grenoble	0	0	0	0



Rabbits of Grenoble* ←→ *Lapins de Grenoble	*trois*	*lapins*	*de*	*Grenoble*
Three	0	0	0	0
Rabbits	0	5/12	1/6	1/6
of	0	1/3	1/3	1/3
Grenoble	0	1/3	1/3	1/3

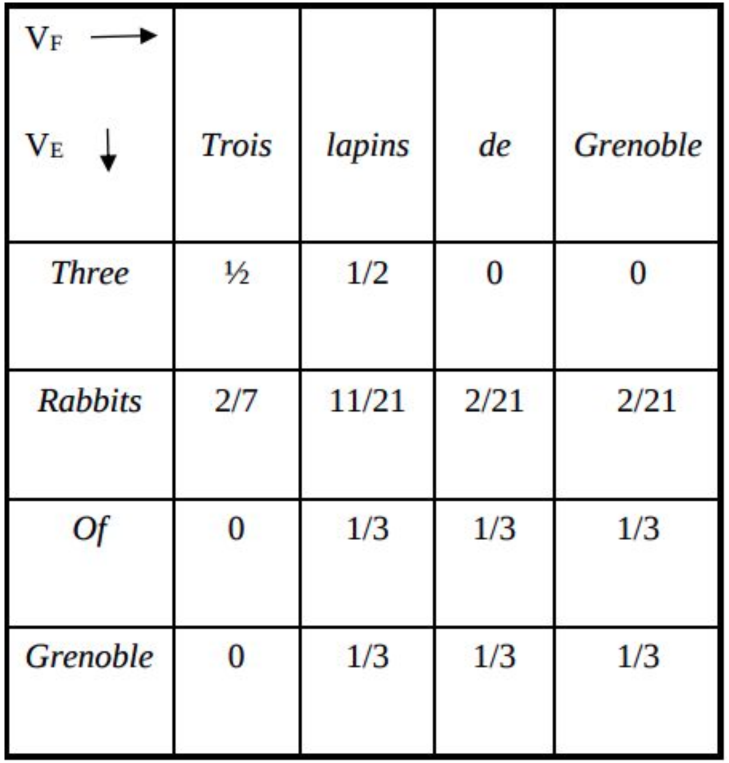
Iteration 3:

The alignment probabilities will get revised as

We have an average entropy = 1.4

Thus P(lapins|rabbit) value goes on increasing (now it is about 0.5) and the average entropy value is already less than the average entropy value with heuristic alignment.

Thus, we are progressing towards a better probability distribution.



$V_F \rightarrow$ $V_E \downarrow$	*Trois*	*lapins*	*de*	*Grenoble*
Three	½	1/2	0	0
Rabbits	2/7	11/21	2/21	2/21
Of	0	1/3	1/3	1/3
Grenoble	0	1/3	1/3	1/3

Generalized Expectation Maximization

Goal:

- Given D = { x_1 , x_2 , x_3 .. x_n }, find $\hat{\theta}$ that maximizes likelihood of sequence D.
- So,

$$\begin{aligned}\hat{\theta} &= \underset{\theta}{\operatorname{argmax}}\, P_X(D|\theta) \\ &= \underset{\theta}{\operatorname{argmax}}\, P_X(D|\theta) \\ &= \underset{\theta}{\operatorname{argmax}} \sum_Z P_{X|Z}(X|Z\theta)P_Z(Z|\theta)\end{aligned}$$

- E-step = given current values of θ and D, guess z_i
- M-step = with new z_i compute new $\hat{\theta}$

Main Idea is that we do not know the complete data likelihood, but compute its expected value given observed data.

- So,

$$Q(\theta|\theta^n) = E_{X|Z\theta^n}[log(P_{XZ}(XZ|\theta))]$$

- Tricky because we want to compute θ but need θ^n
- E-step: Given estimates θ^n

Compute

$$Q(\theta|\theta^n) = E_{X|Z\theta^n}(log(P_{XZ}(XZ|\theta)))$$

- M-step:

$$\theta^{n+1} = \underset{\theta}{\operatorname{argmax}}\, Q(\theta|\theta^n) \tag{11}$$

How do we know it converges to something useful?

- By Jensen’s Inequality

  Note: ln (natural log) is strictly convex

Proof of Convergence

EM Convergence

$$P_{XZ}(D,Z|\theta) = P_{Z|X}(Z|D,\theta)P_X(D|\theta) \tag{12}$$

then

$$log(P_x(D|\theta)) = log(P_{XZ}(DZ|\theta)) - log(P_{Z|X}(Z|D,\theta)) \tag{13}$$

Expectation on both sides

$$log(P_x(D|\theta)) = E_{Z|X\theta^n}[log(P_{XZ}(DZ|\theta))|D] - E_{Z|X|\theta^n}[log(P_{Z|X}(Z|D,\theta))|D] \tag{14}$$

Note that

$$Q(\theta|\theta^n) = E_{Z|X|\theta}[log(P_{XZ}(D, Z|\theta))|D] \tag{15}$$

and let

$$H(\theta|\theta^n) = -E_{Z|X|\theta}[log(P_{Z|X}(Z|D, \theta))|D] \tag{16}$$

then,

$$log(P_X(D|\theta)) = Q(\theta|\theta^n) + H(\theta|\theta^n) \tag{17}$$

Integrating

$$\tag{18}$$

$$
\begin{aligned}
log(P_X(D|\theta^{n+1})) - log(P_X(D|\theta^n)) = Q(\theta^{n+1}|\theta^n) - Q(\theta^n|\theta^n) \\
+H(\theta^{n+1}|\theta^n) - H(\theta^n|\theta^n)
\end{aligned} \tag{19}
$$

but

$$\theta^{n+1} = \operatorname*{argmax}_{\theta} Q(\theta|\theta^n) \tag{20}$$

therefore,

$$Q(\theta^{n+1}|\theta^n) \geq Q(\theta^n|\theta^n) \tag{21}$$

So

$$log(P_X(D|\theta^{n+1})) \geq log(P_X(D|\theta^n)) \tag{22}$$

If

$$H(\theta^{n+1}|\theta^n) \geq H(\theta^n|\theta^n) \tag{23}$$

then,

$$H(\theta^{n+1}|\theta^n) - H(\theta^n|\theta^n) = -E_{Z|X\theta^n}[log(\frac{P_{Z|X}(Z|D\theta^{n+1})}{P_{Z|X}(Z|D\theta^n)})] \tag{24}$$

$$\geq -log(E_{Z|X\theta^n}[log(\frac{P_{Z|X}(Z|D\theta^{n+1})}{P_{Z|X}(Z|D\theta^n)})]) \tag{25}$$

$$= -log \int_2 (P(Z|X\theta^n))\frac{P(Z|D\theta^{n+1})}{P(Z|D\theta^n)}dz \tag{26}$$

$$= -log(1) = 0 \tag{27}$$

Convergence

The Expectation Maximisation Algorithm will always converge.

However, the problem is highly convex.

EM typically converges to a local optimum, not necessarily the global optimum, with no bound on the convergence rate in general.

So, good modelling assumptions are necessary to ensure a good solution.

Alternatives to Expectation Maximisation

As EM can converge to a local optima, a need exists for alternative methods for guaranteed learning, especially in the high-dimensional setting.

Alternatives to EM exist with better guarantees for consistency, which are termed **moment-based approaches** [more info] or the so-called **spectral techniques** [more info].

Moment-based approaches to learning the parameters of a probabilistic model are of increasing interest recently since they enjoy guarantees such as global convergence under certain conditions unlike EM which is often plagued by the issue of getting stuck in local optima. Algorithms with guarantees for learning can be derived for a number of important models such as mixture models, HMMs etc. For these spectral methods, no spurious local optima occur, and the true parameters can be consistently estimated under some regularity conditions. [more comparisons here]