CS626: Speech, NLP and the Web

Expectation Maximization, Alignment, Machine Translation

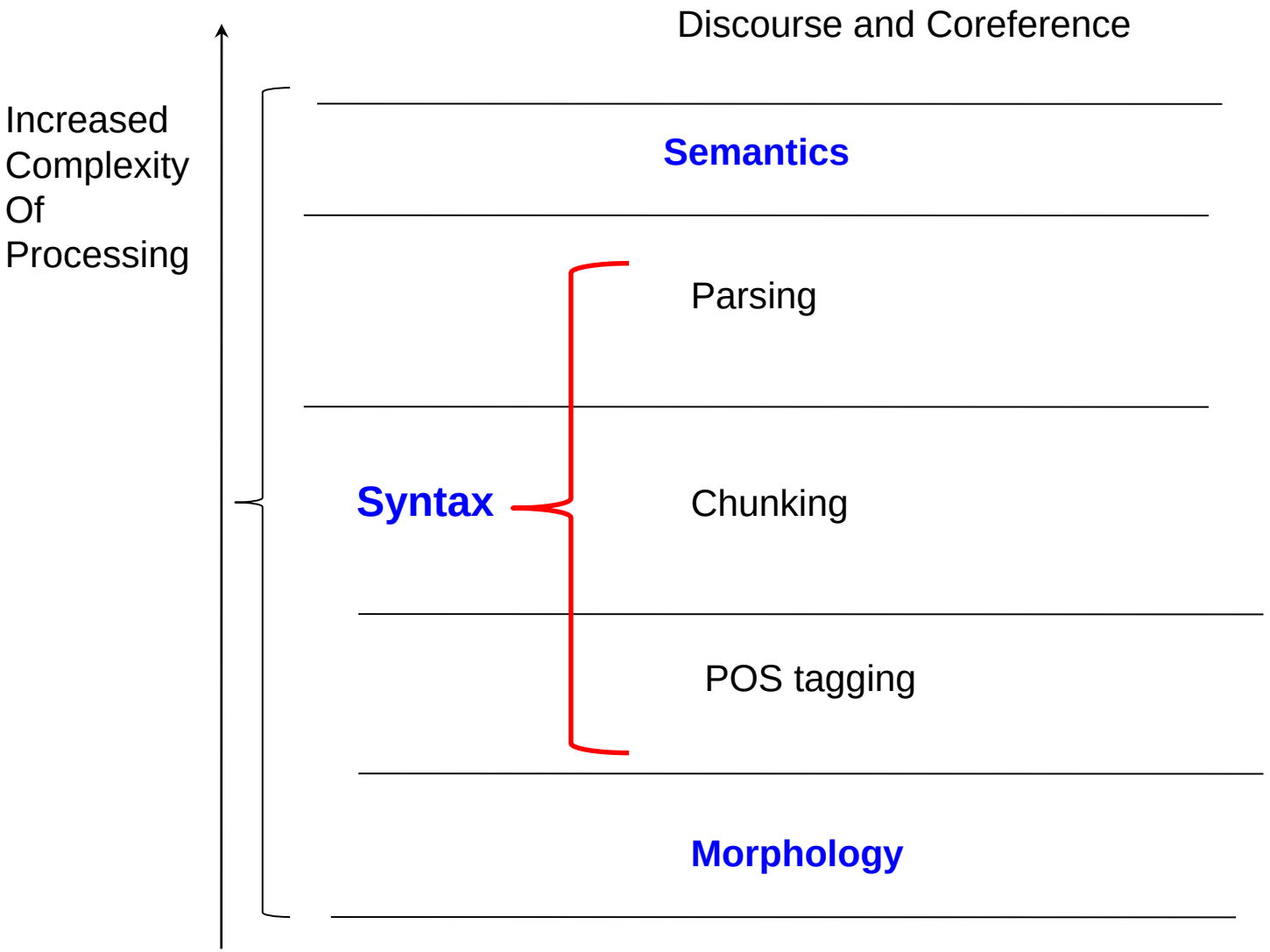
Pushpak Bhattacharyya

Computer Science and Engineering Department

IIT Bombay

Week 12 of 10th October, 2022

NLP Layer and Linguistics



Expectation Maximization

- A very important technique for parameter estimation in presence of hidden variables
- Application in
  - Machine Translation- word alignment
  - HMM- combined transition and emission probabilities
  - PCFG- probabilities of CFG rules

Mathematics of EM

From

Pushpak Bhattacharyya, *Machine Translation*, CRC Press, 2015

Maximum Likelihood of Observations

- ***Situation 1: Throw of a Single Coin***
- The parameter is the probability *p* of getting heads in a single toss. Let *N* be the number of tosses. Then the observation *X* and the data or observation likelihood *D* respectively are:

$$X :< x_1, x_2, x_3, ..., x_{N-1}, x_N >$$

$$D = \prod_{i=1}^{N} p^{x_i} (1-p)^{1-x_i}, \text{ s.t. } x_i = 1 \text{ or } 0, \text{ and } 0 \le p \le 1$$

where x_i is an indicator variable assuming values 1 or 0 depending on the *ith* observation being heads or tail. Since there are *N* identically and independently distributed (*i.i.d.*) observations, *D* is the product of probabilities of individual observations each of which is a Bernoulli trial.

Single coin

Since exponents are difficult to manipulate mathematically, we take log of *D*, also called log likelihood of data, and maximize with regard to *p*. This yields

$$p = \frac{\sum_{i=1}^{N} x_i}{N} = \frac{M}{N}; \; M = \#Heads, N = \#tosses$$

Throw of 2 coins

- Three parameters: probabilities p_1 and p_2 of heads of the two coins and the probability p of choosing the first coin (automatically, $1-p$ is the probability of choosing the second coin).
- N tosses and observations of heads and tails. Only, we do not know which observation comes from which coin.
- Indicator variable z_i is introduced to capture coin choice ($z_i=1$ if coin 1 is chosen, else 0). This variable is hidden, *i.e.*, we do not know its values.
- However, without it the likelihood expression would have been very cumbersome.

Data Likelihood

Data Likelihood,

$D = P_{<p1,p2,p>}(X) = P_\theta(X),\ \theta = <p, p_1, p_2>$

$= \Sigma_Z P_\theta(X,Z))$

$$X :< x_1, x_2, x_3, ..., x_{N-1}, x_N >$$

$$Z :< z_1, z_2, z_3, ..., z_{N-1}, z_N >$$

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

$$\text{s.t. } z_i, x_i = 1 \text{ or } 0, \text{ and } 0 \le p, p_1, p_2 \le 1$$

Invoke Jensen Inequality

We would like to work with $logP_{\theta}(X)$. However, there will be a Σ inside *log*. Fortunately, *log* is a concave function, so that

$$\log\left(\sum_{i=1}^{K} \lambda_i y_i\right) \geq \left(\sum_{i=1}^{K} \lambda_i \log(y_i)\right); \sum_{i=1}^{K} \lambda_i = 1$$

Log likelihood of Data

$LL(D)$= log likelihood of data

$= log(P_\theta(X))= log(\Sigma_Z P_\theta(X,Z))$

$= log[\Sigma_Z \lambda_Z (P_\theta(X,Z)/\lambda_Z)]; \Sigma_Z \lambda_Z = 1$

$>= \Sigma_Z [\lambda_Z log[(P_\theta(X,Z)/\lambda_Z)]$

After a number of intricate mathematical steps

$LL(D) >= E_{Z|X,\theta} log(P_\theta(X,Z))$, where $E(.)$ is the expectation function; note that the expectation is conditional on X.

Expectation of log likelihood

$$E_{Z|X}[\log(P_\theta(X,Z)]$$

$$= E_{Z|X}\left[\log\prod_{i=1}^{N}\left[\left(pp_1{}^{x_i}\left(1-p_1\right)^{1-x_i}\right)^{z_i}\left((1-p)\,p_2{}^{x_i}\left(1-p_2\right)^{1-x_i}\right)^{1-z_i}\right]\right]$$

$$= E_{Z|X}\left[\sum_{i=}^{N} z_i\left(\log p + x_i\log p_1 + (1-x_i)\log\left(1-p_1\right)\right) + (1-z_i)\left(\log(1-p) + x_i\log p_2 + (1-x_i)\log\left(1-p_2\right)\right)\right]$$

$$= \sum_{i=1}^{N}\left[E(z_i\mid x_i)\left(\log p + x_i\log p_1 + (1-x_i)\log\left(1-p_1\right)\right) + (1-E(z_i\mid x_i))\left(\log(1-p) + x_i\log p_2 + (1-x_i)\log\left(1-p_2\right)\right)\right]$$

$$\text{s.t. } z_i, x_i = 1 \text{ or } 0, \text{ and } 0 \le p, p_1, p_2 \le 1$$

Derivation of E and M steps for 2 coin problem (1/2)- M step

Take partial derivative of $E_{Z|X,\theta}(.)$ (prev. slide) wrt p, p_1, p_2 and equate to 0.

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

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

$$p_2 = \frac{M - \sum_{i=1}^{N} E(z_i \mid x_i) x_i}{N - \sum_{i=1}^{N} E(z_i \mid x_i)}; M = \#Heads, N = \#tosses$$

Derivation of E and M steps for 2 coin problem (2/2)- E step

$E(z_i|x_i) = 1.P(z_i=1|x_i) + 0.P(z_i=0|x_i)$

$= P(z_i=1|x_i)$

$$P(z_i = 1 | x_i) = \frac{P(z_i = 1, x_i)}{P(x_i)}$$

$$= \frac{pp_1^{x_i}(1-p_1)^{1-x_i}}{P(x_i, z_i = 1) + P(x_i, z_i = 0)}$$

$$= \frac{pp_1^{x_i}(1-p_1)^{1-x_i}}{pp_1^{x_i}(1-p_1)^{1-x_i} + (1-p)p_2^{x_i}(1-p_2)^{1-x_i}}$$

Generalization into N “throws” using M “things” each having L outcomes

From

Pushpak Bhattacharyya, *Machine Translation*, CRC Press, 2015

Multiple outcomes from multiple entities

- “Throw” of “something” where that something has more than 2 outcomes, e.g., throw of multiple dice
- The observation sequence has a sequence of 1 to 6s
- But we do not know which observation came from which dice
- Gives rise to a multinomial that is extremely useful in NLP ML.

Observation Sequence

- N ‘throws’, 1 of L outcomes from each throw, 1 of the M ‘things’ (called ‘sources’) chosen
- $\Sigma_{k=1,L} x_{ik}=1$, since each x_{ik} is either 1 or 0 and one and only one of them is1.
- D (data):

$$<x_{11}/x_{12}/\ldots x_{1L}>, <x_{21}/x_{22}/\ldots x_{2L}>, \ldots$$
$$<x_{N1}/x_{N2}/\ldots x_{NL}>$$

Hidden Variable

- Hidden variable for M sources
- $\Sigma_{j=1,M} z_{ij} = 1$, since each z_{ij} is either 1 or 0 and one and only one of them is 1.
- Z:

$$<z_{11}/z_{12}/\ldots z_{1M}>, <z_{21}/z_{22}/\ldots z_{2M}>, \ldots$$
$$<z_{N1}/z_{N2}/\ldots z_{NM}>$$

Parameters

- Parameter set θ:
  - π_j: probability of choosing source j
  - p_{jk}: probability of observing k^{th} outcome from the j^{th} source

This will be elaborated next week; only expressions are given now

M-step

M-Step:

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

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

E-step

E-Step:

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

Machine Learning Based MT

Alignment is the crux of the matter

Chronology

- IBM Models of Alignment- Brown *et al.* 1990, 1993
- Phrase Based MT- Koehn 2003
- Encoder Decoder- Sutskever *et al.* 2014, Cho *et al.* 2014
- Attention- Bahadanu *et al.* 2015
- Transformer- Vaswani *et al.* 2017

Czeck-English data

- [nesu] “I carry”
- [ponese] “He will carry”
- [nese] “He carries”
- [nesou] “They carry”
- [yedu] “I drive”
- [plavou] “They swim”

To translate …

- I will carry.
- They drive.
- He swims.
- They will drive.

Hindi-English data

- [DhotA huM] “I carry”
- [DhoegA] “He will carry”
- [DhotA hAi] “He carries”
- [Dhote hAi] “They carry”
- [chalAtA huM] “I drive”
- [tErte hEM] “They swim”

Bangla-English data

- [bai] “I carry”
- [baibe] “He will carry”
- [bay] “He carries”
- [bay] “They carry”
- [chAlAi] “I drive”
- [sAMtrAy] “They swim”

To translate … (repeated)

- I will carry.
- They drive.
- He swims.
- They will drive.

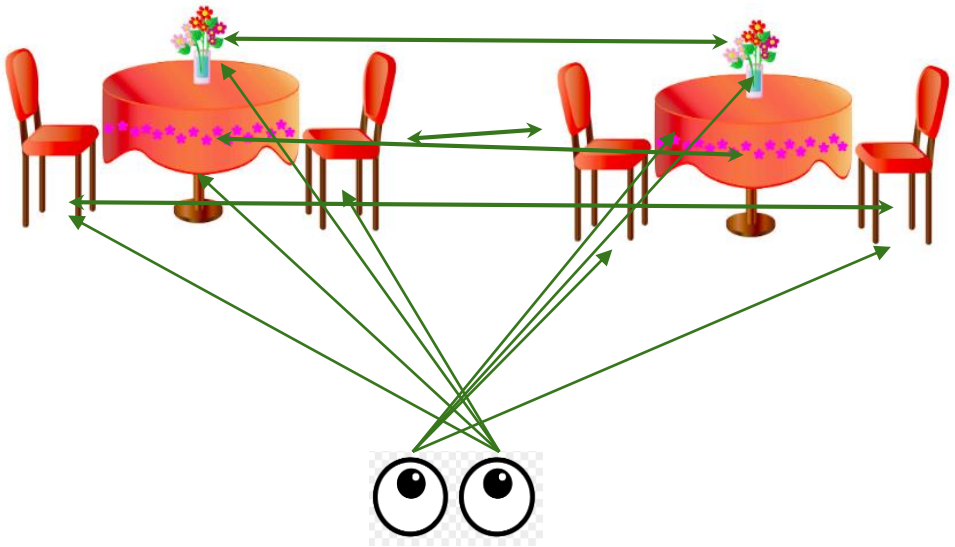
Foundation

- Data driven approach
- Goal is to find out the English sentence e given foreign language sentence f whose p(e|f) is maximum.
- Translations are generated on the basis of statistical model
- Parameters are estimated using bilingual parallel corpora

$$\tilde{e} = \underset{e \in e^*}{\operatorname{argmax}}\, p(e|f) = \underset{e \in e^*}{\operatorname{argmax}}\, p(f|e)p(e)$$

How to build part alignment from whole alignment

- Two images are in alignment: images on the two retina
- Need to find alignment of parts of it



EM for word alignment from sentence alignment: example

English	French
(1) three rabbits	(1) trois lapins
a b	w x
(2) rabbits of Grenoble	(2) lapins de Grenoble
b c d	x y z

Initial Probabilities:
each cell denotes *t(a ←→ w), t(a ←→ x) etc.*

	a	b	c	d
w	1/4	1/4	1/4	1/4
x	1/4	1/4	1/4	1/4
y	1/4	1/4	1/4	1/4
z	1/4	1/4	1/4	1/4

Example of expected count

$C[w \leftarrow\rightarrow a;\ (a\ b) \leftarrow\rightarrow (w\ x)]$

$$= \frac{t(w \leftarrow\rightarrow a)}{t(w \leftarrow\rightarrow a)+t(w \leftarrow\rightarrow b)} \times \#(a \text{ in } \text{`a b'}) \times \#(w \text{ in } \text{`w x'})$$

$$= \frac{1/4}{1/4+1/4} \times 1 \times 1 = 1/2$$

“counts”

a b ←→ ***w x***	a	b	c	d
w	1/2	1/2	0	0
x	1/2	1/2	0	0
y	0	0	0	0
z	0	0	0	0

b c d ←→ ***x y z***	a	b	c	d
w	0	0	0	0
x	0	1/3	1/3	1/3
y	0	1/3	1/3	1/3
z	0	1/3	1/3	1/3

Revised probability: example

$$t_{revised}(a \leftrightarrow w)$$

$$= \frac{1/2}{(1/2+1/2+0+0)_{(a\ b)\leftrightarrow(w\ x)} + (0+0+0+0)_{(b\ c\ d)\leftrightarrow(x\ y\ z)}}$$

Revised probabilities table

	a	b	c	d
w	1/2	1/2	0	0
x	1/4	5/12	1/6	1/6
y	0	1/3	1/3	1/3
z	0	1/3	1/3	1/3

“revised counts”

a b ←→ ***w x***	a	b	c	d
w	1/2	3/8	0	0
x	1/2	5/8	0	0
y	0	0	0	0
z	0	0	0	0

b c d ←→ ***x y z***	a	b	c	d
w	0	0	0	0
x	0	5/9	1/3	1/3
y	0	2/9	1/3	1/3
z	0	2/9	1/3	1/3

Re-Revised probabilities table

	a	b	c	d
w	1/2	1/2	0	0
x	3/16	**85/144**	1/9	1/9
y	0	1/3	1/3	1/3
z	0	1/3	1/3	1/3

Continue until convergence; notice that (b,x) binding gets progressively stronger; b=rabbits, x=lapins

Derivation of EM based Alignment Expressions

$$V_E = \text{vocalbulary of language } L_1 \text{ (Say English)}$$

$$V_F = \text{vocabulary of language } L_2 \text{ (Say Hindi)}$$

E^1 *what is in a name ?*
नाम में क्या है ?

F^1 *naam meM kya hai ?*
name in what is ?

E^2 *That which we call rose, by any other name will smell as sweet.*
जिसे हम गुलाब कहते हैं, और भी किसी नाम से उसकी कुशबू समान मीठा होगी

F^2 *Jise hum gulab kahte hai, aur bhi kisi naam se uski khushbu samaan mitha hogii*
That which we rose say , any other name by its smell as sweet
That which we call rose, by any other name will smell as sweet.

Vocabulary mapping

Vocabulary

V_E	V_F
what , is , in, a , name , that, which, we , call ,rose, by, any, other, will, smell, as, sweet	*naam, meM, kya, hai, jise, ham, gulab, kahte, aur, bhi, kisi, bhi, uski, khushbu, saman, mitha, hogii*

Key Notations

English vocabulary : V_E
French vocabulary : V_F
No. of observations / sentence pairs : S
Data D which consists of S observations looks like,

$$e^1{}_1, e^1{}_2, \ldots, e^1{}_{l^1} \Leftrightarrow f^1{}_1, f^1{}_2, \ldots, f^1{}_{m^1}$$

$$e^2{}_1, e^2{}_2, \ldots, e^2{}_{l^2} \Leftrightarrow f^2{}_1, f^2{}_2, \ldots, f^2{}_{m^2}$$

.....

$$e^s{}_1, e^s{}_2, \ldots, e^s{}_{l^s} \Leftrightarrow f^s{}_1, f^s{}_2, \ldots, f^s{}_{m^s}$$

.....

$$e^S{}_1, e^S{}_2, \ldots, e^S{}_{l^S} \Leftrightarrow f^S{}_1, f^S{}_2, \ldots, f^S{}_{m^S}$$

No. words on English side in s^{th} sentence : l^s
No. words on French side in s^{th} sentence : m^s
$index_E(e^s{}_p)$ =Index of English word $e^s{}_p$in English vocabulary/dictionary
$index_F(f^s{}_q)$ =Index of French word $f^s{}_q$in French vocabulary/dictionary

(Thanks to Sachin Pawar for helping with the maths formulae processing)

Hidden variables and parameters

Hidden Variables (Z) :
Total no. of hidden variables $= \sum_{s=1}^{S} l^s m^s$ where each hidden variable is as follows:
$z_{pq}^s = 1$, if in s^{th} sentence, p^{th} English word is mapped to q^{th} French word.
$z_{pq}^s = 0$, otherwise

Parameters (Θ) :
Total no. of parameters $= |V_E| \times |V_F|$, where each parameter is as follows:
$P_{i,j}$ = Probability that i^{th} word in English vocabulary is mapped to j^{th} word in French vocabulary

Likelihoods

Data Likelihood *L(D; Θ)* :

$$L(D;\Theta) = \prod_{s=1}^{S} \prod_{p=1}^{l^s} \prod_{q=1}^{m^s} \left(P_{index_E(e_p^s), index_F(f_q^s)} \right)^{z_{pq}^s}$$

Data Log-Likelihood LL(D; Θ) :

$$LL(D;\Theta) = \sum_{s=1}^{S} \sum_{p=1}^{l^s} \sum_{q=1}^{m^s} z_{pq}^s \log \left(P_{index_E(e_p^s), index_F(f_q^s)} \right)$$

Expected value of Data Log-Likelihood E(LL(D; Θ)) :

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

Constraint and Lagrangian

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

$$\sum_{s=1}^{S} \sum_{p=1}^{l^s} \sum_{q=1}^{m^s} E(z_{pq}^s) \, log \left(P_{index_E(e_p^s), index_F(f_q^s)} \right) - \sum_{i=1}^{|V_E|} \lambda_i \left(\sum_{j=1}^{|V_F|} P_{i,j} - 1 \right)$$

Differentiating wrt P_{ij}

$$\sum_{s=1}^{S}\sum_{p=1}^{l^s}\sum_{q=1}^{m^s}\delta_{index_E(e_p^s),i}\,\delta_{index_F(f_q^s),j}\left(\frac{E(z_{pq}^s)}{P_{i,j}}\right)-\lambda_i=0$$

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

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

Final E and M steps

M-step

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

E-step

$$E(z_{pq}^s) = \frac{P_{index_E(e_p^s),index_F(f_q^s)}}{\sum_{q'=1}^{m^s}P_{index_E(e_p^s),index_F(f_{q'}^s)}}, \forall s,p,q$$

Tools that implement word alignment

- Giza++ which comes with Moses
- Berkeley Aligner

Indian Language SMT *(LREC 2014)*

	hi	ur	pa	bn	gu	mr	kK	ta	te	ml	en
hi		61.28	68.21	34.96	51.31	39.12	37.81	14.43	21.38	10.98	29.23
ur	61.42		52.02	29.59	39.00	27.57	28.29	11.95	16.61	8.65	22.46
pa	73.31	56.00		29.89	43.85	30.87	30.72	10.75	18.81	9.11	23.83
bn	37.69	32.08	31.38		28.14	22.09	23.47	10.94	13.40	8.10	18.76
gu	55.66	44.12	45.14	28.50		32.06	30.48	12.57	17.22	8.01	19.78
mr	45.11	32.60	33.28	23.73	32.42		27.81	10.74	12.89	7.65	17.62
kK	41.92	34.00	34.31	24.59	31.07	27.52		10.36	14.80	7.89	17.07
ta	20.48	18.12	15.57	13.21	16.53	11.60	11.87		8.48	6.31	11.79
te	28.88	25.07	25.56	16.57	20.96	14.94	17.27	8.68		6.68	12.34
ml	14.74	13.39	12.97	10.67	9.76	8.39	9.18	5.90	5.94		8.61
en	28.94	22.96	22.33	15.33	15.44	12.11	13.66	6.43	6.55	4.65	

Baseline PBSMT - % BLEU scores (S1)

- **Clear partitioning of translation pairs by language family pairs**, based on translation accuracy.
  - Shared characteristics within language families make translation simpler
  - Divergences among language families make translation difficult

(Anoop Kunchukuttan, Abhijit Mishra, Pushpak Bhattacharyya, LREC 2014)