CS626: Speech, NLP and the Web

Expectation Maximization, Alignment, Machine Translation

Pushpak Bhattacharyya

Computer Science and Engineering Department

IIT Bombay

Week 13 of 17^{th} October, 2022

EM: 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:

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

SMT

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

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

SMT: Language Model

- To detect *good* English sentences
- Probability of an English sentence $w_1w_2 \ldots\ldots w_n$ can be written as

$$Pr(w_1w_2 \ldots\ldots w_n) = Pr(w_1) * Pr(w_2|w_1) * \ldots * Pr(w_n|w_1 w_2 \ldots w_{n-1})$$

- Here $Pr(w_n|w_1\, w_2 \ldots w_{n-1})$ is the probability that word w_n follows word string $w_1\, w_2 \ldots w_{n-1}$.
  - N-gram model probability

- Trigram model probability calculation

$$p(w_3|w_1w_2) = \frac{count(w_1w_2w_3)}{count(w_1w_2)}$$

SMT: Translation Model

- How to assign the values to $p(e|f)$?
  - Sentences are infinite, not possible to find pair(e,f) for all sentences

$$p(f|e) = \frac{count(f, e)}{count(e)}$$ ← Sentence level

- Introduce a hidden variable ***a***, that represents alignments between the individual words in the sentence pair

$$\Pr(\boldsymbol{f}|\boldsymbol{e}) = \sum_{\boldsymbol{a}} \Pr(\boldsymbol{f}, \boldsymbol{a}|\boldsymbol{e})$$ ← Word level

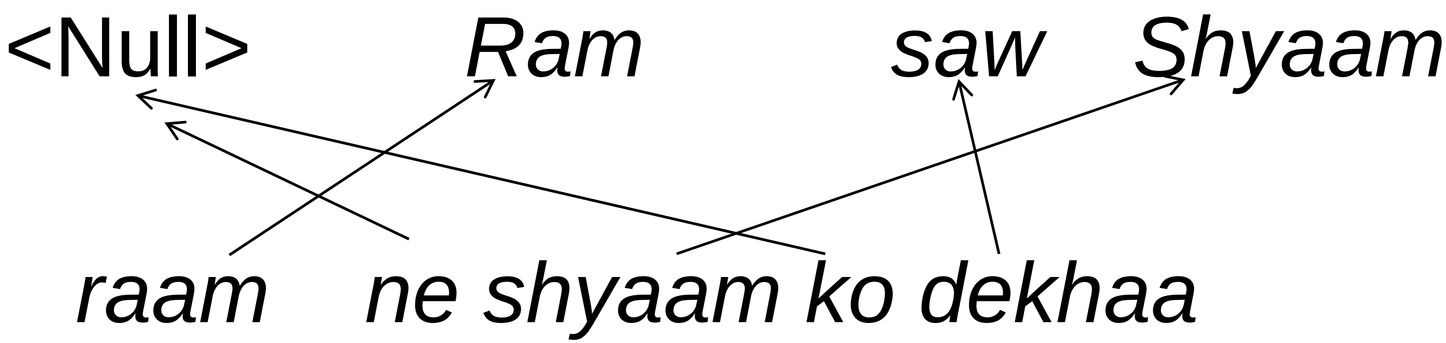
Alignment

- If the string, $e= e_1^l= e_1\ e_2\ \ldots e_l$, has l words, and the string, $f= f_1^m=f_1f_2\ldots f_m$, has m words,
- then the alignment, a, can be represented by a series, $\boldsymbol{a_1^m= a_1a_2\ldots a_m}$, of m values, each between 0 and l such that if the word in position j of the f-string is connected to the word in position i of the e-string, then
  - $\boldsymbol{a_j= i}$, and
  - if it is not connected to any English word, then $\boldsymbol{a_j= O}$

Example of alignment

English (e): *Ram saw Shyam*

Hindi (f): *raam ne shyaam ko dekhaa*



Translation Model: Exact expression

$$\Pr(\boldsymbol{f}, \boldsymbol{a} | \boldsymbol{e}) = \Pr(m|\boldsymbol{e}) \prod_{j=1}^{m} \Pr\left(a_j \,|a_1^{j-1}, f_1^{j-1}, m, \boldsymbol{e}\right) \Pr\left(f_j | a_1^{j}, f_1^{j-1}, m, \boldsymbol{e}\right)$$

Choose the length of foreign language string given *e*	Choose alignment given *e* and *m*	Choose the identity of foreign word given *e, m, a*

- Five models for estimating parameters in the expression [2]
- Model-1, Model-2, Model-3, Model-4, Model-5

Proof of Translation Model: Exact expression

$$\Pr(f \mid e) = \sum_a \Pr(f, a \mid e)$$; marginalization

$$\Pr(f, a \mid e) = \sum_m \Pr(f, a, m \mid e)$$; marginalization

$$= \sum_m \Pr(m \mid e) \Pr(f, a \mid m, e)$$

$$= \sum_m \Pr(m \mid e) \Pr(f, a \mid m, e)$$

$$= \sum_m \Pr(m \mid e) \prod_{j=1}^{m} \Pr(f_j, a_j \mid a_1^{j-1}, f_1^{j-1}, m, e)$$

$$= \sum_m \Pr(m \mid e) \prod_{j=1}^{m} \Pr(a_j \mid a_1^{j-1}, f_1^{j-1}, m, e) \Pr(f_j \mid a_1^{j}, f_1^{j-1}, m, e)$$

m is fixed for a particular *f*, hence

$$= \Pr(m \mid e) \prod_{j=1}^{m} \Pr(a_j \mid a_1^{j-1}, f_1^{j-1}, m, e) \Pr(f_j \mid a_1^{j}, f_1^{j-1}, m, e)$$

Notion of Fertility

Fertility is the number of words in the target sentence that each word in the source sentence produces

English (e): *Ram is speaking*

Bengali (f): *raam bolchhe*

fertility(raam)=1;

corresponds (not aligns) *to "Ram"*

fertility(bolchhe)=2; to "is speaking"

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,

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

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

.....

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

.....

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

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)) :

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

cs: ml, for nt: pushpak

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

Phrase Based MT

An Example

English--> Hindi ↓	*People*	*Of*	*Mumbai*
Mumbai			X
Ke		X	
Log	X		

Table X.5: creation of phrase alignments from word alignments through grow-dia algorithm

Phrase Alignment Process

- Run IBM model 3 in both directions- source to target and target to source- to create what are called *alignment sets*. There are two alignment sets: one in each direction.

- Then apply a process called *symmetrisation* to obtain phrase alignments.

Alignments: “People of Mumbai”←→”Mumbai ke logoM”

- *A1: {<Mumbai, Mumbai>, <of, ke>, <people, log>}*

- *A2: {< mumbai, Mumbai>, < ke, of>, < log, people>}*

Grow Diagonal Process

- *People of --> ke log* (blue square)
- *of Mumbai --> mumbai ke* (yellow square)
- *People of Mumbai --> mumbai ke log* (red square)

Illustration with “people of…”

English--> Hindi ↓	*People*	*Of*	*Mumbai*
Mumbai			X
Ke		X	
Log	X		

Linguistic and Non-linguistic Phrases

- ‘*People of Mumbai*’ → ‘*mumbai ke log*’
  - noun phrase (NP) alignment

- ‘*of Mumbai*’ → ‘*mumbai ke*’
  - preposition phrase (PP),

- ‘*people of*’ → ‘*ke log*’ is not a linguistic phrase (headedness property violated)

Case of Null Alignment

English--> Hindi ↓	*Meet*	*The*	*People*	*Of*	*Mumbai*
Mumbai					X
Ke				X	
LogoM			X		
Se					
Miliye	X				

Table X.6: phrase alignments in case of null alignment

Growing Bigger Alignments

English--> Hindi ↓	*Meet*	*The*	*People*	*Of*	*Mumbai*
Mumbai					X
Ke				X	
LogoM			X		
Se					
Miliye	X				

Table X.6: phrase alignments in case of null alignment

Grow-Diag with null alignment

- The red box will expand into the cell <*se, the*> and create the alignment '*the people of Mumbai*'-->'*mumbai ke logoM se*'.
- The alignment '*the people'-->'logoM*' too will be created, merging the <*ke, the*> cell with <*logoM, people*>.
- Cells from null rows or null columns can be merged upwards or downwards, thereby associating the row-word/column-word with the next phrase or the previous phrase.

Consequence of Grow-Diag with null alignment

- Due to the null alignment of ‘*the*’, all of the following phrase alignments are possible:

  ‘*meet the*’--> ‘*se miliye*’

  ‘*the people*’ <-> ’*logoM*’

  ‘*the people*’ --> ’*logoM se*’

‘the’ and ‘se’ both can be *both* prefix and suffix of phrases.

Influence of Data (1/2)

- Q: Which phrase amongst the above will be retained?
- A: ALL! But with different probabilities.
- '*the people*'--> '*logoM*' should have higher probability than '*the people*'--> '*logoM se*',
- Because '*people*' is likely to seen more in the company of '*logoM*' than '*logoM se*'
- as in *'tell the people'--> 'logoM ko bolo'*, '*have faith in the people*'--> '*logoM pe viswaas rakho*' and so on