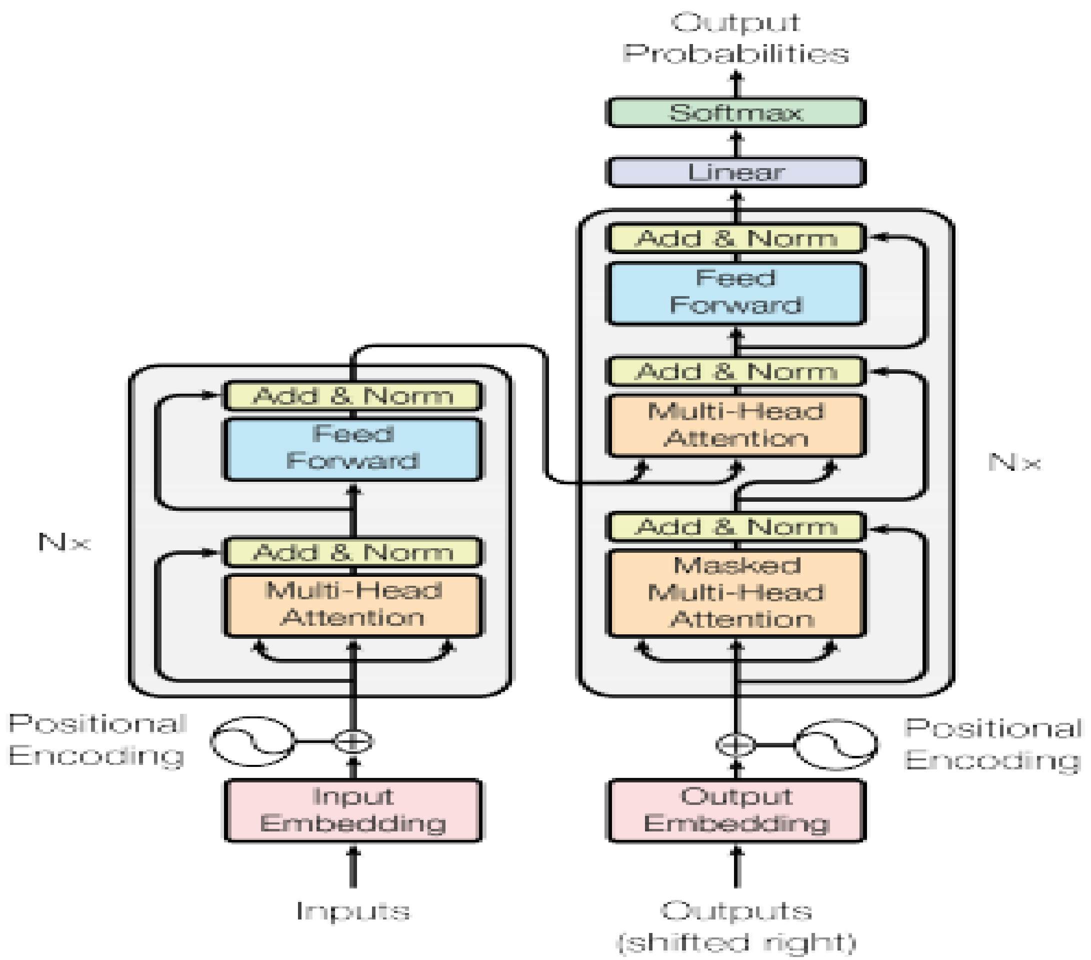
CS772: Deep Learning for Natural Language Processing

Attention and Transformer

Pushpak Bhattacharyya
Computer Science and Engineering Department
IIT Bombay

Week 12 of 21st Mar, 2022

A classic diagram and a classic paper



Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. "Attention is all you need." NeurIPS (2017).

http://nlp.seas.harvard.edu/2018/04/03/attention.html
http://jalammar.github.io/illustrated-transformer/

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

Attention

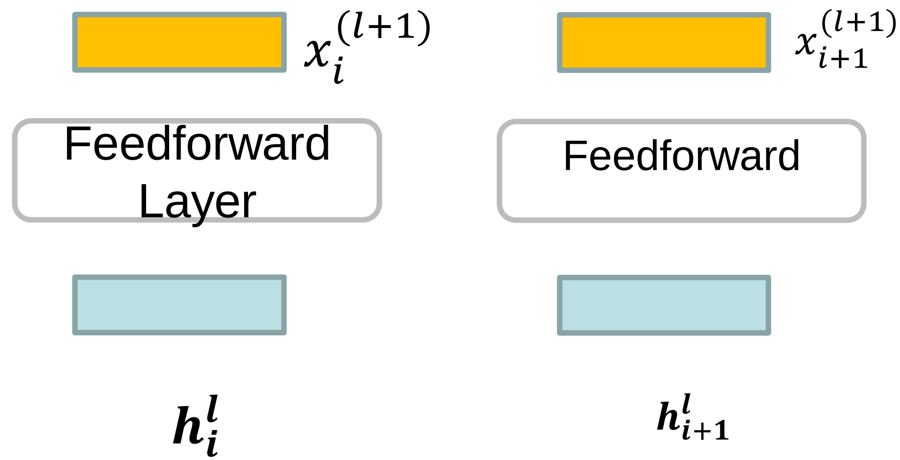
Compare every elements with all other elements

Represent the input context as *a* weighted average of input word embeddings

$$\boldsymbol{h_i^l} = \sum_{i=1}^{N} \boldsymbol{w_i} x_i$$

$$\boldsymbol{x_i^{l+1}} = \boldsymbol{FF}(h_i^l)$$

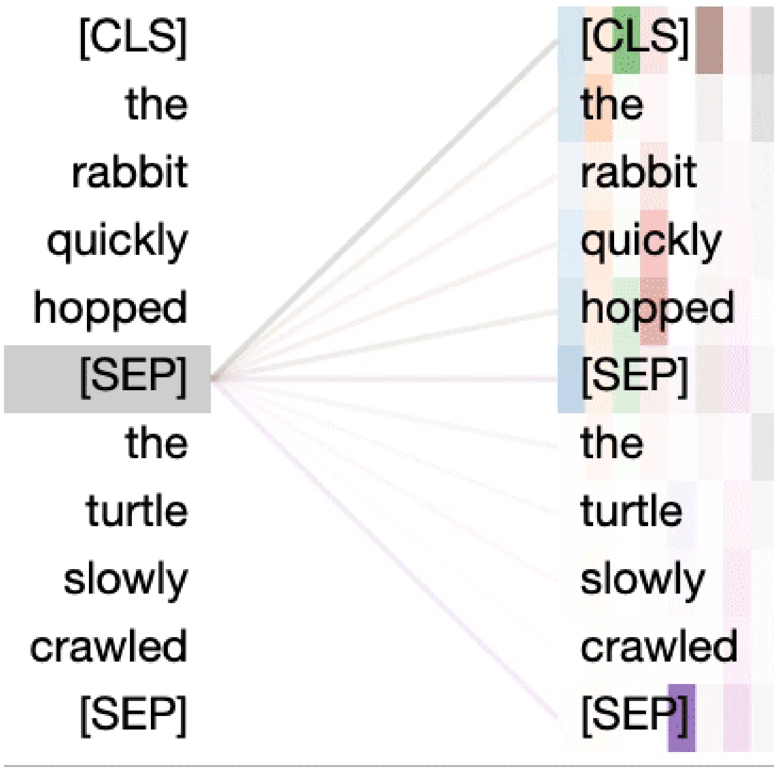
How do we compute weights ➔ Attention!



Non-recurrent ➔ this operation can be applied in parallel to all elements in the sequence

Self-Attention

Every word is compared with every other word in the same sentence



X ➔ query

$x_1, x_2\ x_3 \ldots x_n$ ➔ values

Direct comparison between arbitrary words ➔ long-range dependencies can be better modelled

More computations than Recurrent models: $O(n^2)$

Important observations on self attention

- In the input sequence, pairs of words differ in their strength of association

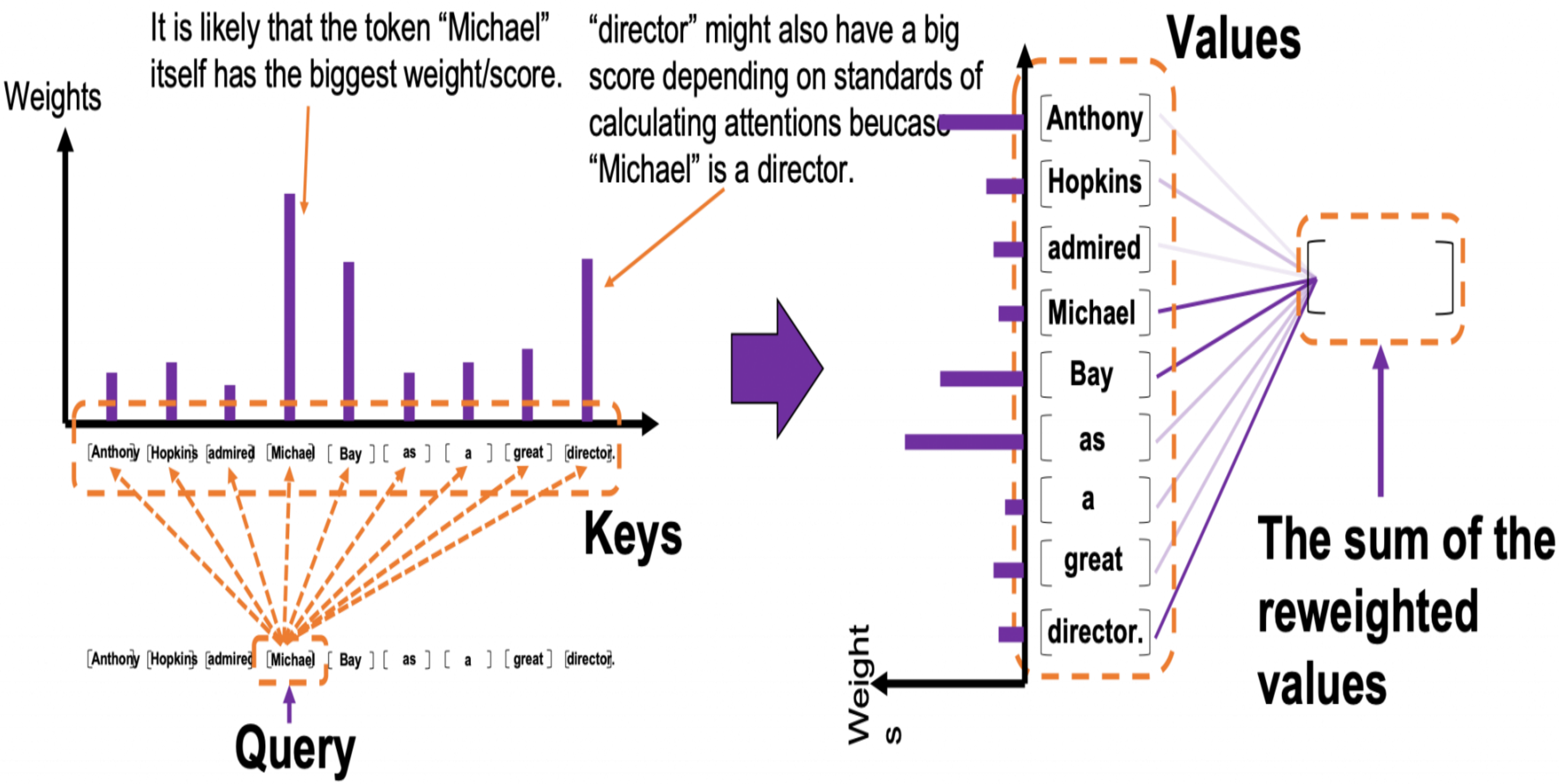
- For example for an adjective-noun combination, adjective’s attention should be stronger for the noun than for other words in the sentence

- So the key questions are:
  - What to attend to
  - With how much attention to attend to

Attention that is non-self

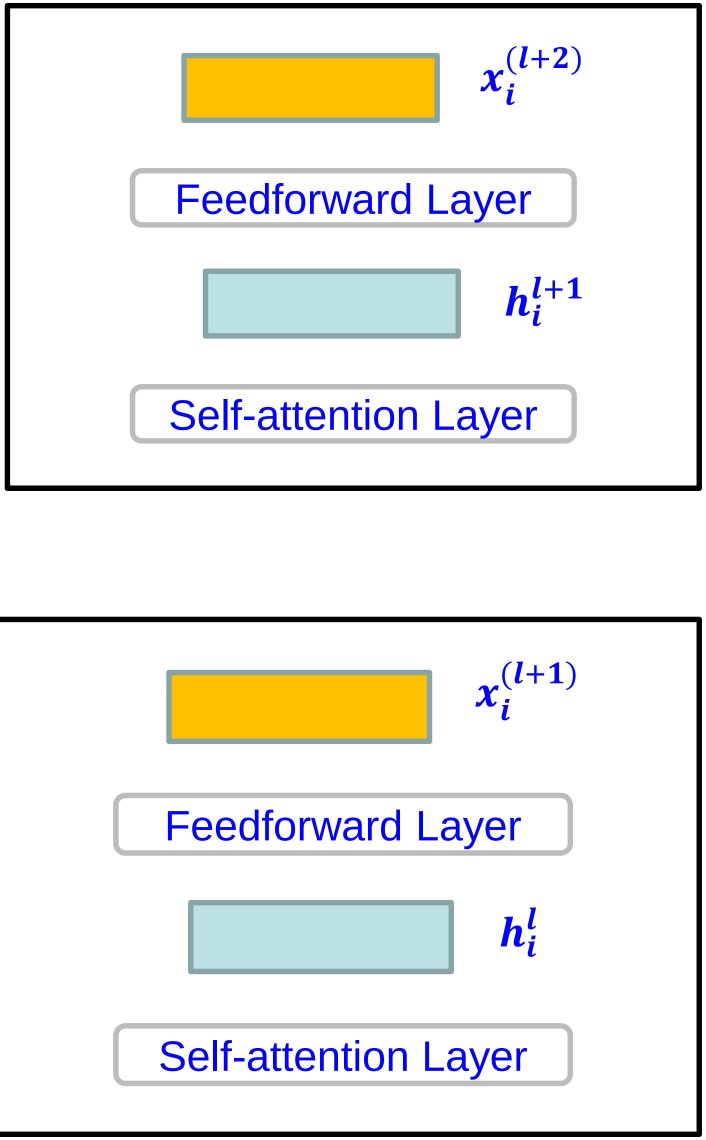
- When the decoder generates the output sequence, attention is a 2-part attention
- Each output token should attend to whatever token has been output before
- Additionally, it should attend to the tokens in the input sequence

Fundamental concepts- “Attention”, “query”, “key”, “value”



Transformer Architecture

Stack self-attention blocks to create deep networks



Positional Embeddings

The ICICI bank branch is on the bank of the river

The self-attention model has no notion of position,
➔ same words will have same representations irrespective of their position/syntactic role in the sentence

Create positional embeddings that uniquely and deterministically identify a position

Add it to the word embedding at the bottom layer

https://kazemnejad.com/blog/transformer_architecture_positional_encoding/

Digression

Phrase Based SMT (PBSMT) and distortion

Governing equation

$$e_{best} = \arg\max_e P(e \mid f) = \arg\max_e [P(f \mid e) P_{LM}(e)]$$

where e and f have their usual meaning of output and input respectively; the translation with the highest score is e_{best}. $P(f|e)$ and $P_{LM}(e)$ are the translation model and language model, respectively.

Modelling *P(f|e)*

$$P(\overline{f}_1^I \mid \overline{e}_1^I) = P(\overline{f_1}, \overline{f_2}, \ldots, \overline{f_I} \mid \overline{e_1}, \overline{e_2}, \ldots, \overline{e_I})$$

$$= \prod_{i=1}^{I} \Phi(\overline{f_i} \mid \overline{e_i})\, d(start_i - end_{i-1} - 1)$$

LHS is the probability of sequence of *I* phrases in the sentence *f*, given *I* phrases in sentence *e*. *Φ* is called the *phrase translation probability* and *d(.)* is the *distortion probability.*

Distortion Probability

- $d(start_i\text{-}end_{i-1}\text{-}1)$
- $start_i$: starting position of the translation of the i^{th} phrase of *e* in *f*
- end_{i-1}: end position of the translation of the $(i\text{-}1)^{th}$ phrase of *e* in *f*

- The quantity $start_i\text{-}end_{i-1}\text{-}1$ is a measure of the distance between the translation of i^{th} phrase and the translation of the $(i\text{-}1)^{th}$ phrase of *e* as they appear as the *end*th and *start*th phrase in *f*.

- It is, thus, also a measure of the *reordering* of phrases induced by the translation.

Need for phrases (linguistic phrases and non-linguistic "phrases")

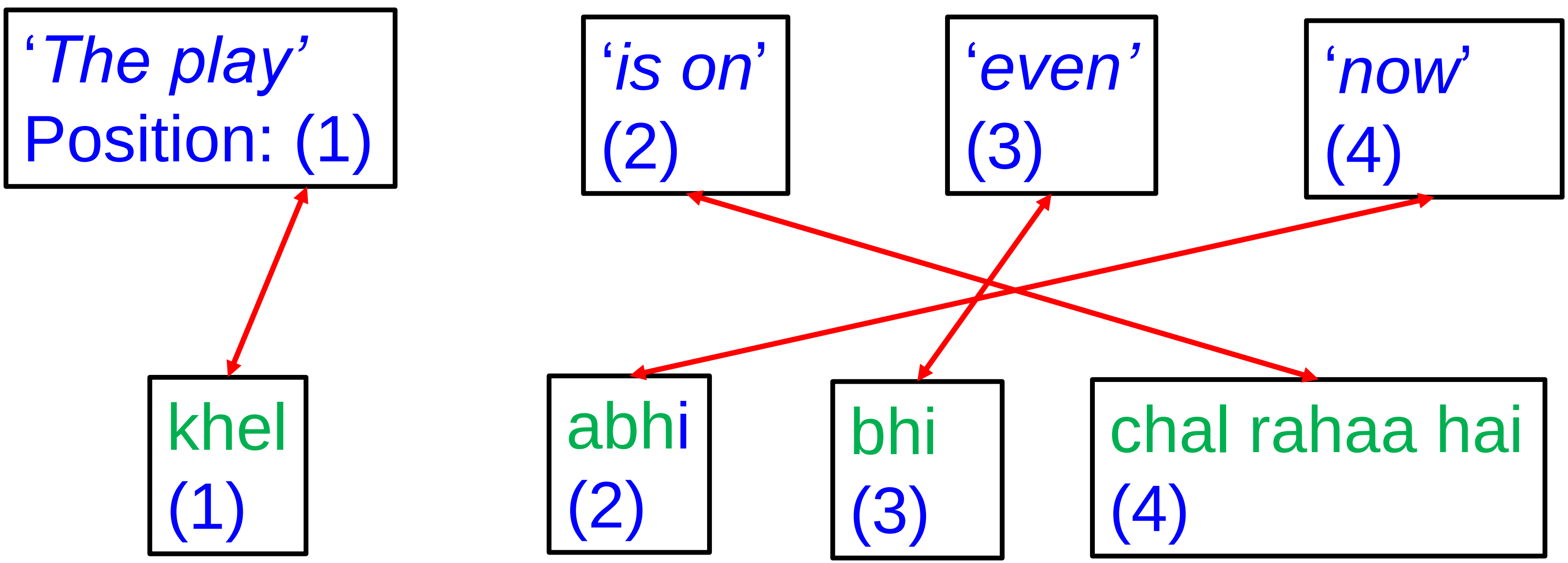
- *"The play is on"* ←→ *"khel chal rahaa hai"*
- *"is on"* ←→ *"chal rahaa hai"*
- IMP: treat '*is*' and '*on*' together and NOT separately

- Otherwise, '*on*' might map to '*rahaa*' which will take away some probability mass of '*on*' onto Hindi word mappings like 'on' ←→ {'par', 'upar', …}
- May produce non-fluent translations like

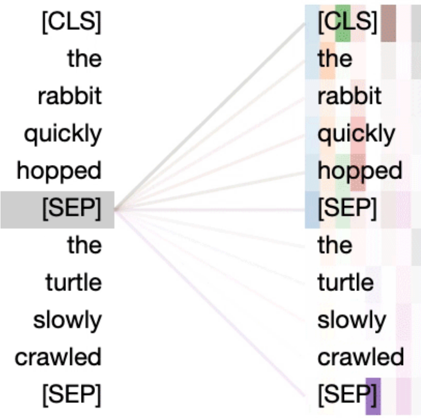
  "*the book is on the table*" ←→ "*kitaab mej rahaa hai*" instead of "*kitaab mej par hai*"

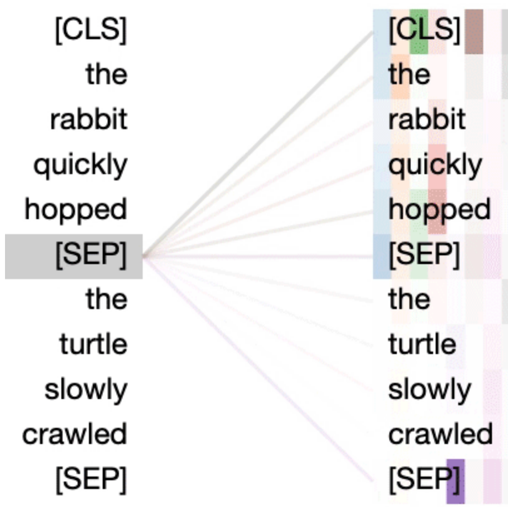
Back to distortion

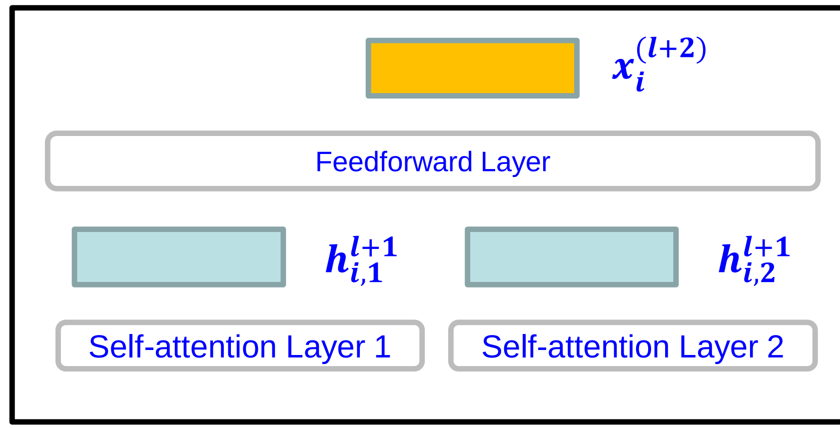
- *“The play is on even now”* ←→ *“khel abhii bhii chal rahaa hai”*
- Mappings:



Multiple self-attention heads



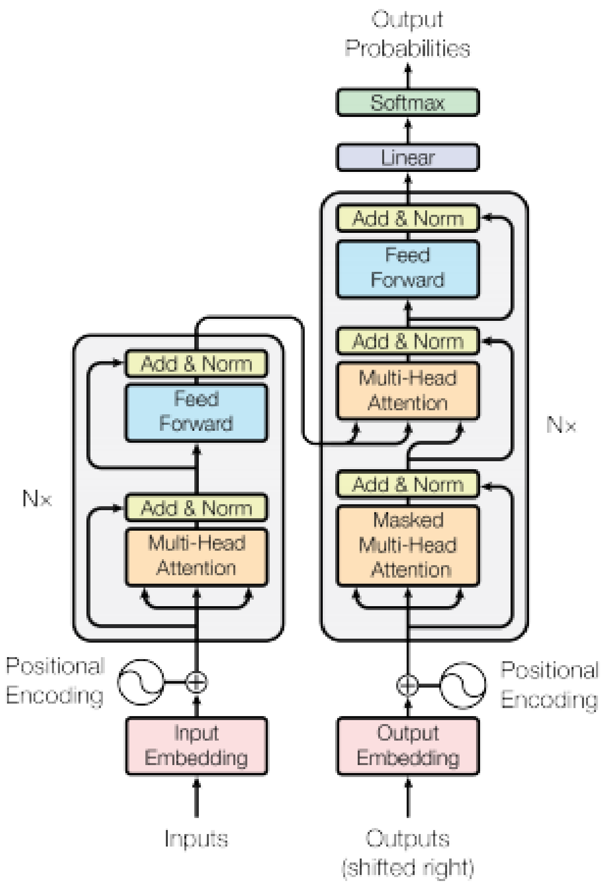




Multiple self-attention networks at each layer

Each head learns different kinds of dependencies

Putting it all together



Decoder layer also has a cross-attention layer

Decoder ➔ masking for future time-steps while computing self-attention

There are residual connections & layer-normalization between layers

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. "Attention is all you need." NeurIPS (2017).

http://nlp.seas.harvard.edu/2018/04/03/attention.html
http://jalammar.github.io/illustrated-transformer/

Transformer has led to tremendous advances in MT

Encoder architectures like BERT based on Transformer have yielded large improvements in NLU tasks

Transformer models are the de-facto standard models for many NLP tasks

Back to attention

What is “Attention”

- **Attention** enhances the important parts of the input data and fades out the rest

- The network should devote more computing power on that small part of the data that matters

Sentence-1

- Ram who is a good student and lives in London which is a large metro, will go to the University for higher studies.
- राम जो एक अच्छा छात्र है और लंदन में रहता है जो एक बड़ी मेट्रो है, उच्च अध्ययन के लिए विश्वविद्यालय जाएगा।

Sentence-2

- Sita who is a good student and lives in London which is a large metro, will go to the University for higher studies.
- सीता जो एक अच्छी छात्रा है और लंदन में रहती है जो एक बड़ी मेट्रो है, उच्च अध्ययन के लिए विश्वविद्यालय जाएगी।

Learning “Attention”

- Which part of the data is more important than others depends on the context

- Learned through training data by **gradient descent**

Two kinds of Attention

- Dot Product Attention

- Multihead Attention

Dependency Parse- Attention by Parsing

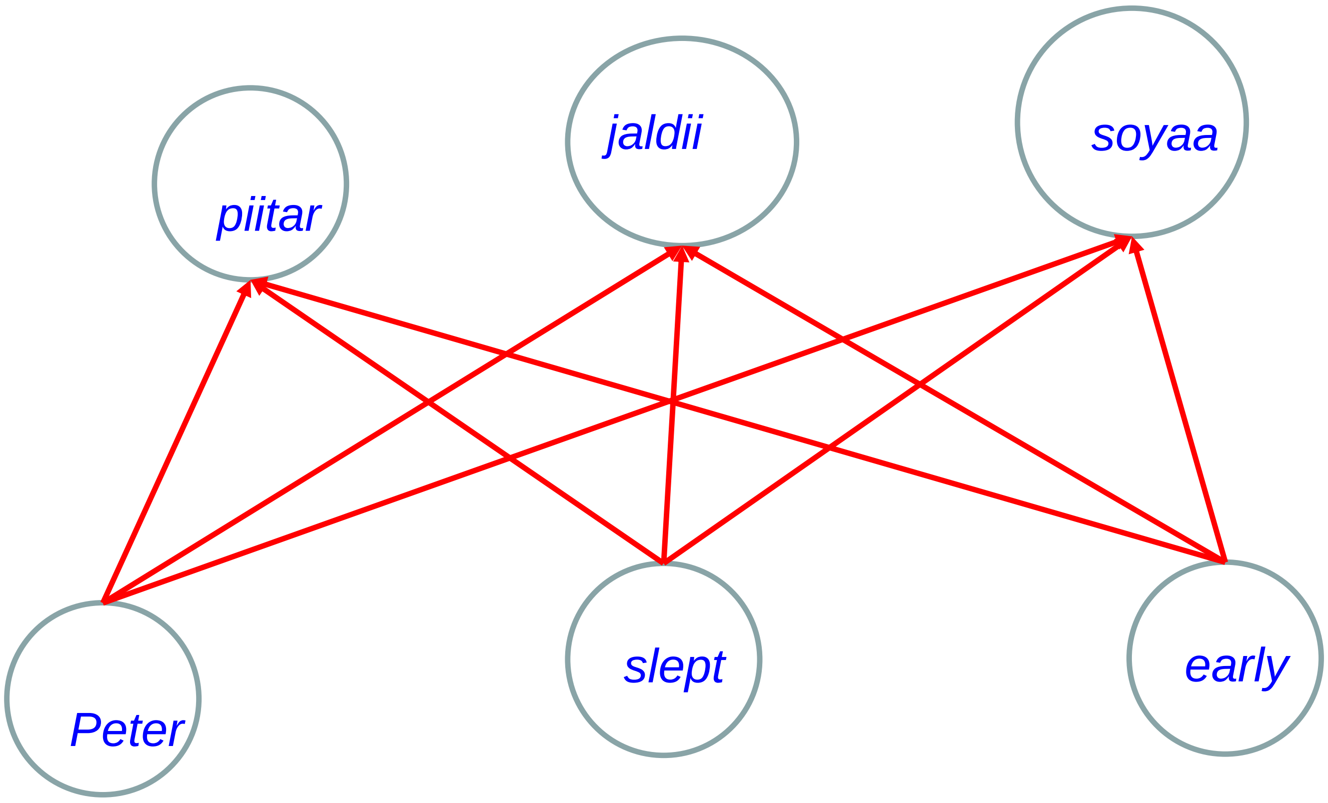
- root(ROOT-0, go-18)
- nsubj(go-18, Ram-1)
- nsubj(student-6, who-2)
- cop(student-6, is-3)
- det(student-6, a-4)
- amod(student-6, good-5)
- acl:relcl(Ram-1, student-6)
- cc(lives-8, and-7)
- conj(student-6, lives-8)
- case(London-10, in-9)
- nmod(lives-8, London-10)
- nsubj(metro-15, which-11)
- cop(metro-15, is-12)
- det(metro-15, a-13)
- amod(metro-15, large-14)
- acl:relcl(student-6, metro-15)
- aux(go-18, will-17)
- case(University-21, to-19)
- det(University-21, the-20)
- obl(go-18, University-21)
- case(studies-24, for-22)
- amod(studies-24, higher-23)
- nmod(University-21, studies-24)

Attention and Alignment

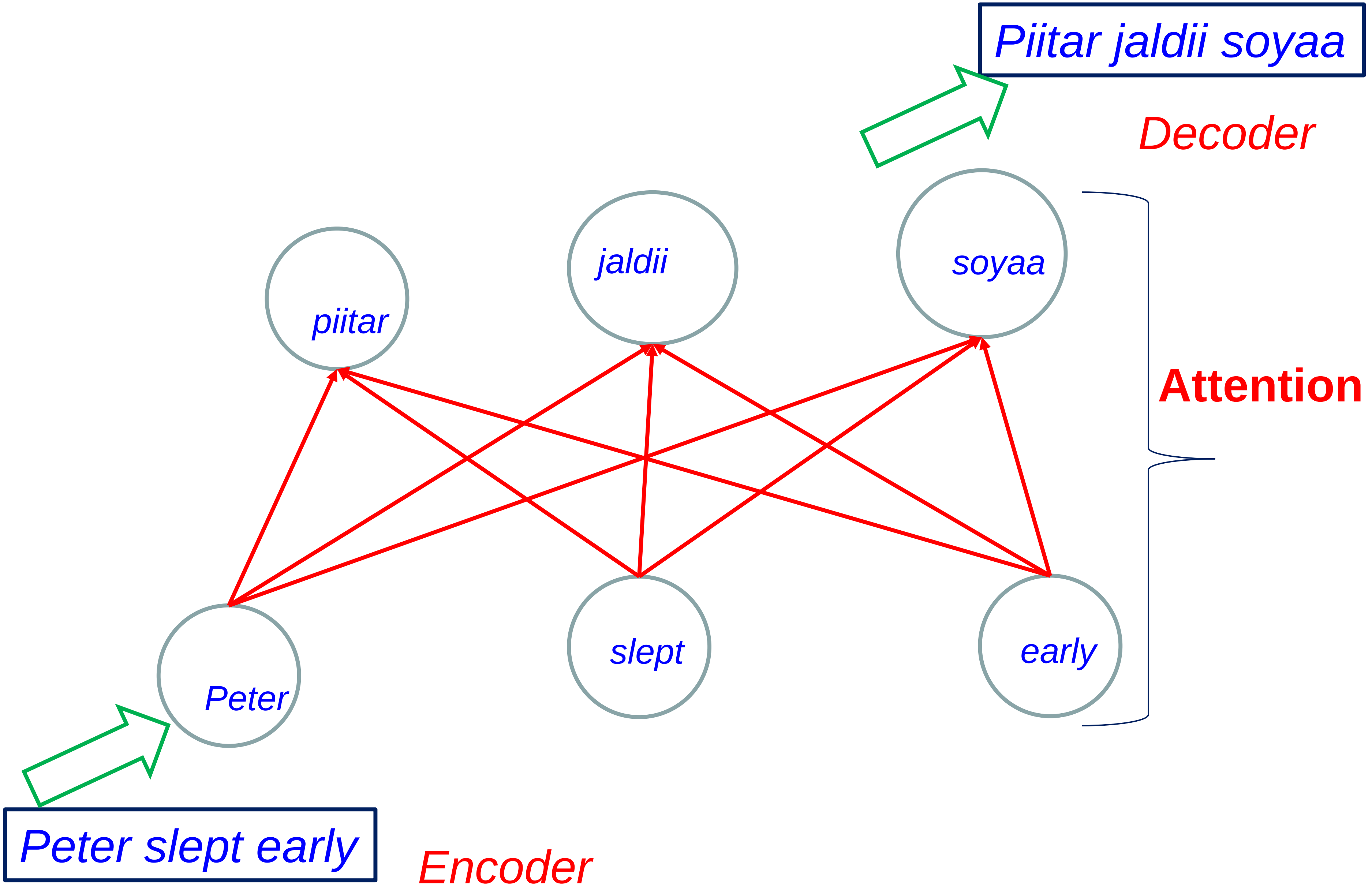
Hindi (col) --> English (row) | \/	PIITAR (पीटर)		JALDII (जल्दी)		SOYAA (सोया)
PETER	1		0		0
SLEPT	0		0		1
EARLY	0		1		0

FFNN for alignment:

Peter slept early → piitar jaldii soyaa

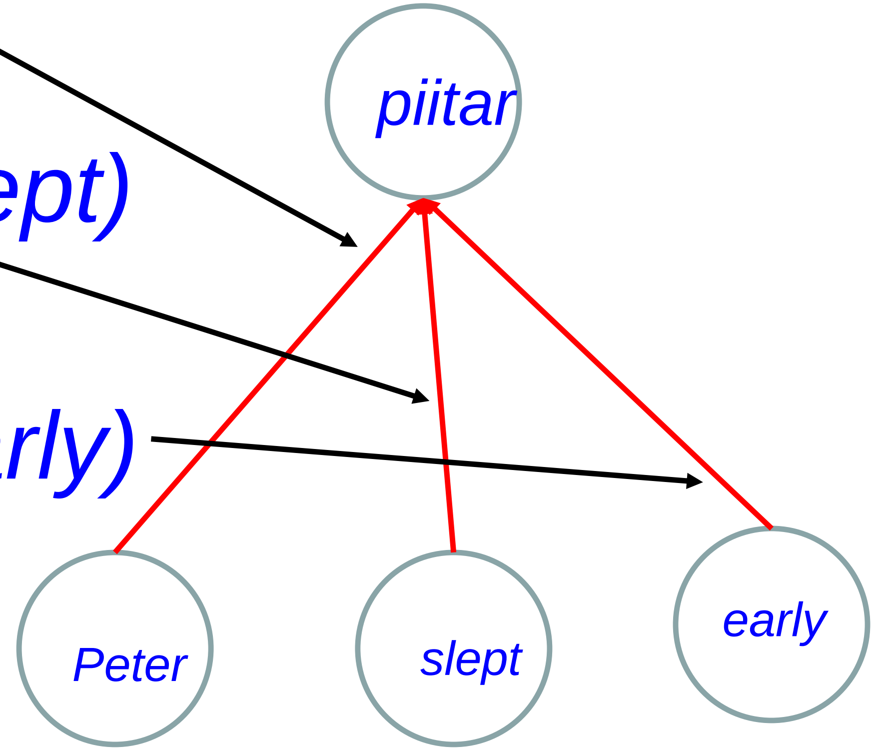


Introduce Attention Layer between Encoder and Decoder



How to learn the weights- attention weights?

- Weight (*piitar*, *peter*)
- Weight (*piitar*, *slept*)
- Weight (*piitar*, *early*)



Statistical Alignment Learning

Non-neural

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

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

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

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

“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

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

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

E1 *what is in a name ?*

नाम में क्या है ?

F1 *naam meM kya hai ?*

name in what is ?

E2 *That which we call rose, by any other name will smell as sweet.*

जिसे हम गुलाब कहते हैं, किसी और नाम से पुकारने पर भी उसकी खुशबू समान मीठा होगी

F2 *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 Hindi 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

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

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

P_{ij} is "asymmetric" in the sense that the dictionary mapping is obtained by "looking" from the English side, i^{th} English word mapping to SOME HIndi word; we can "look" from the Hindi side too;
Then we take the average of P_{ij} and P_{ji}

Aligners like GIZA++, Moses, Berkley etc. do this

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

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