Machine Translation

Using Sequence-to-Sequence Modeling

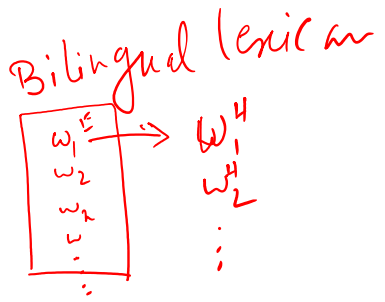
Machine Translation

- Machine Translation (MT) is the task of translating a sentence x from one language (the source language) to a sentence y in another language (the target language).

Be the change you want to see in the world

वह परिवर्तन बनो जो संसार में देखना चाहते हो

1950s: Early Machine Translation

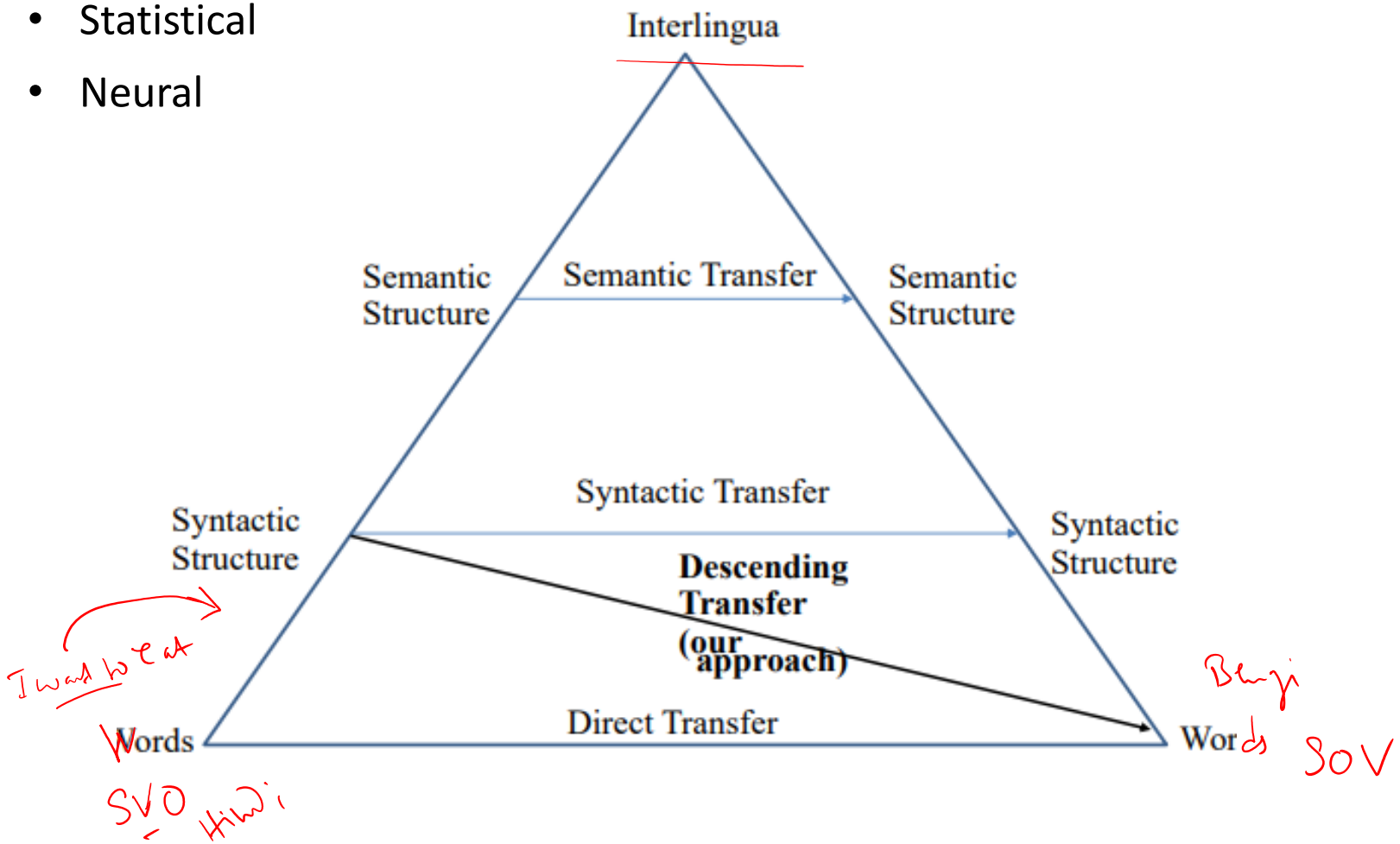


- Machine Translation research began in the early 1950s.
- Russian → English (motivated by the Cold War!)
- Systems were mostly rule-based, using a bilingual dictionary to map Russian words to their English counterparts

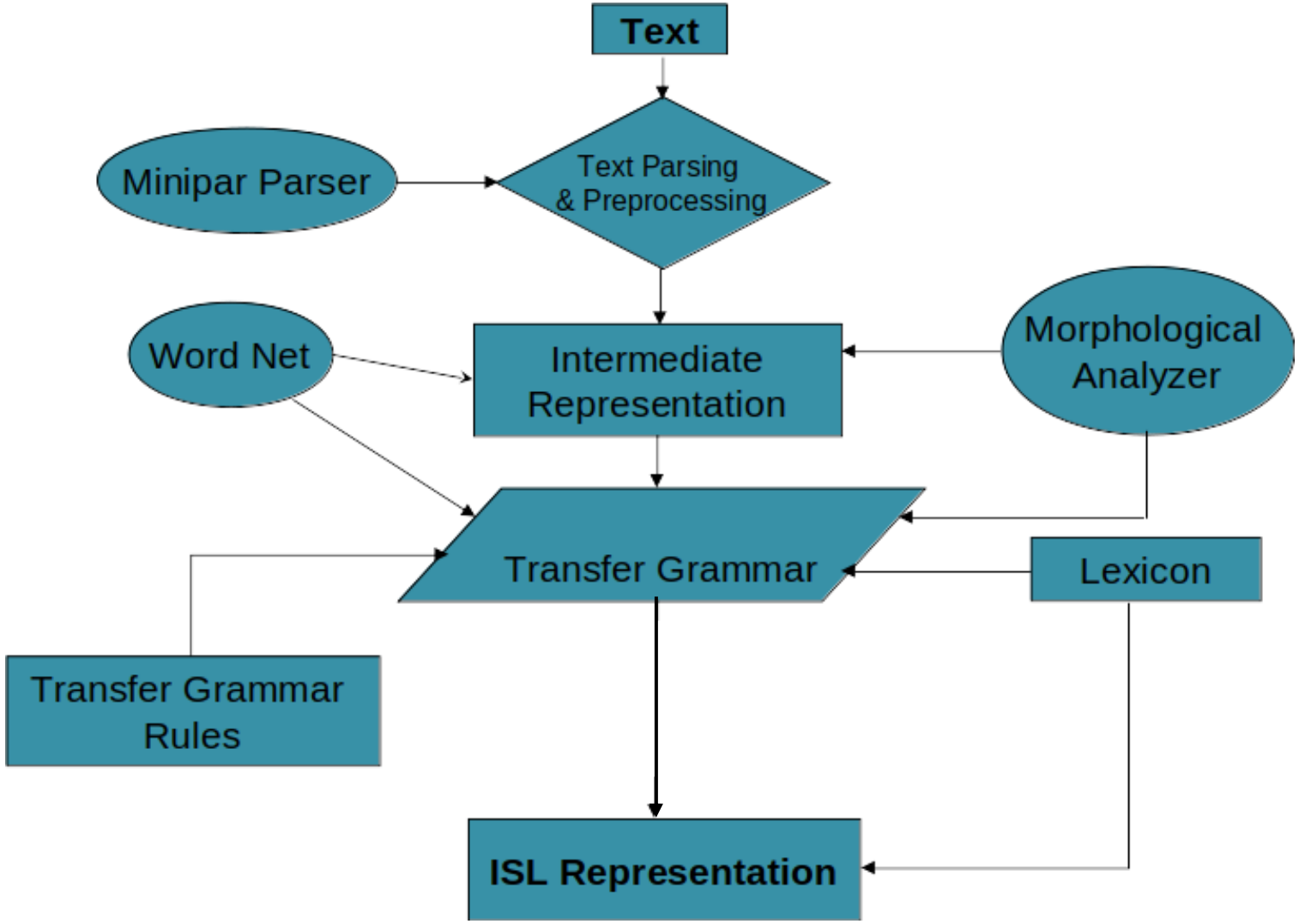
https://www.youtube.com/watch?v=K-HfpsHPmvw

MT Approaches

- Rule Based
- Statistical
- Neural



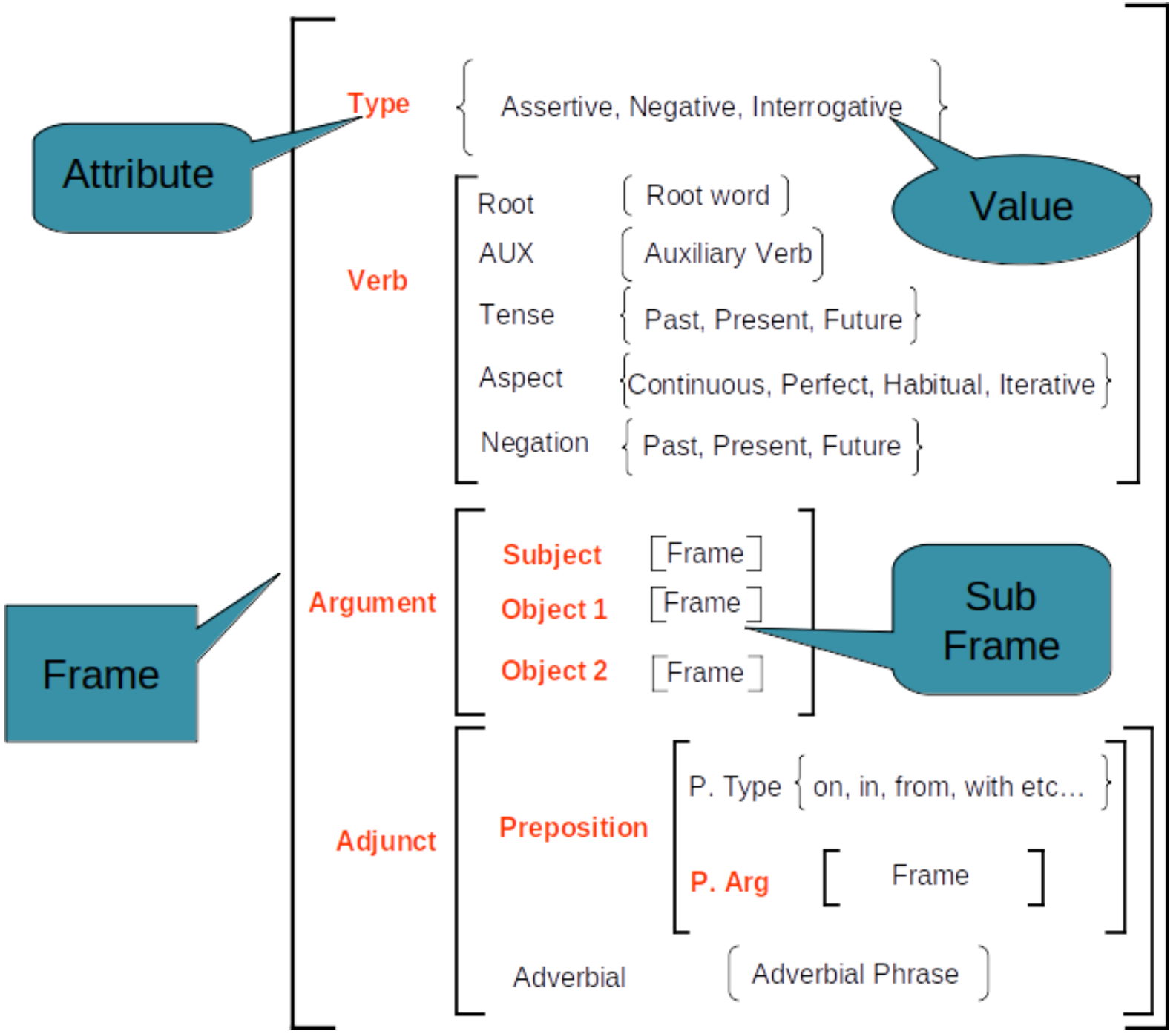
Typical Rule-based MT Systems



Intermediate Representation

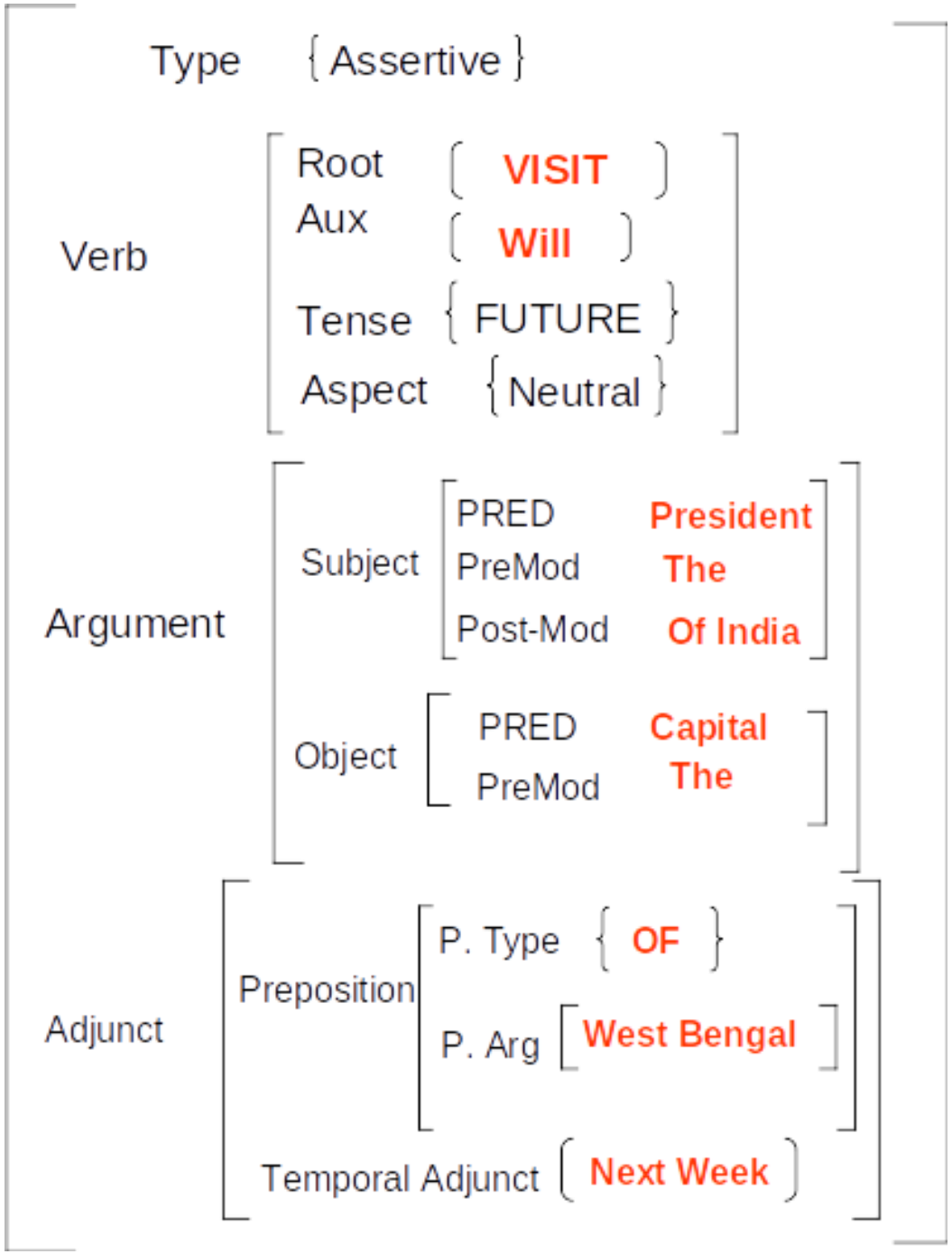
- The dependency structure is represented by recursive frame based structure.
- Each frame corresponds to the higher syntactic and functional information of a sentence
- This higher syntactic and functional information is represented as a set of attribute-value pairs.

Recursive Frame



Example:

“The President of India will visit the capital of west Bengal next week”

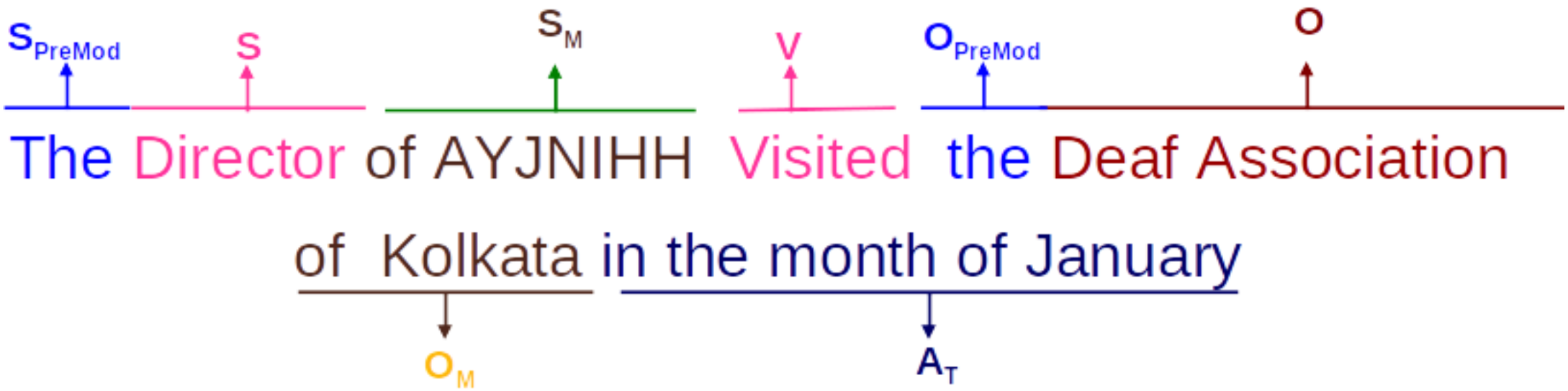


The Transfer Phase

- Mapping of source language frame (s-frame) to target language frame (t-frame)
- Lexical selection
- Structural Association
  - Many languages are free word order
    - SOV pattern is preferred
  - We define generic mapping rule for English to <TL> syntactic transfer
  - Rules were constructed with the help of <TL> experts
  - Rules may vary based on sentence type

Example Rules

- Rule 1: [SPre-Mod][S][SM][V][VM][O][OM][AT] → [A'T][S'M][S'][SPre-Mod][O'M][O'][V'M][V']
- Rule 2: [Aux][S][SM][V][VM][O][OM][AT] → [A'T][S'M][S'][O'M][O'][V'M][V'][Yes/no-Marker]
- Rule 3: [Wh-Phrase][S][SM][V][VM][O][OM][AT] → [A'T][S'M][S'][O'M][O'][V'M][V'][Wh-Phrase]
- Rule 4: Pred (X, Premod, Prep) Pred (X)
- Rule 5: Pred (X, Premod, Indef Det) → Pred (X, Premod, INDEX (?))
- Rule 6: Pred (X, AUX) → 0
- Rule 7: Tense (X, Future), Verb(X) → Verb (X+<Future>)
- Rule 8: Tense (X, Past), Verb (X) → Verb (X+<Past>)
- Rule 9: Tense (X, Present), Prog (X) → 0
- Rule 10: Pred (X, Pronoun, person=1,number=1) → Pred (INDEXIPSI)
- Rule 11: Pred (X, Premod, Adj) → Pred (X, Postmod, Adj)



Rule 1: Pred (X, Det)➔ 0

Rule 2: Pred (X, Prep List)➔ 0

Rule 3: Tense (X, Past), Verb (X) ➔ Verb (X+<FINISH>)

Rule 4: [$S_{Pre\text{-}Mod}$] [S] [S_M] [V] [V_M] [O] [O_M] [A_T]➔ [A'_T] [S'_M][S'_M] [$S_{Pre\text{-}Mod}$] [O'_M] [O'] [V'_M] [V']

1990s-2010s: Statistical Machine Translation

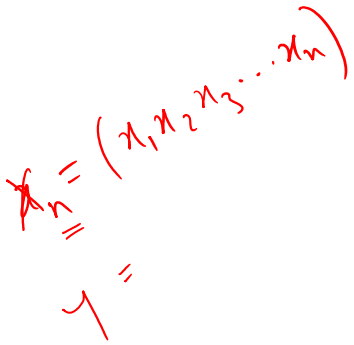
- Core idea: Learn a probabilistic model from data
- Suppose we're translating Hindi → English.
- We want to find best English sentence y, given Hindi sentence x

$$\text{argmax}_y P(y|x)$$

- Use Bayes Rule to break this down into two components to be learnt separately

$$= \text{argmax}_y P(x|y)P(y)$$

- Question: How to learn translation model ?
- First, need large amount of parallel data (e.g. pairs of human-translated Hindi/English sentences)



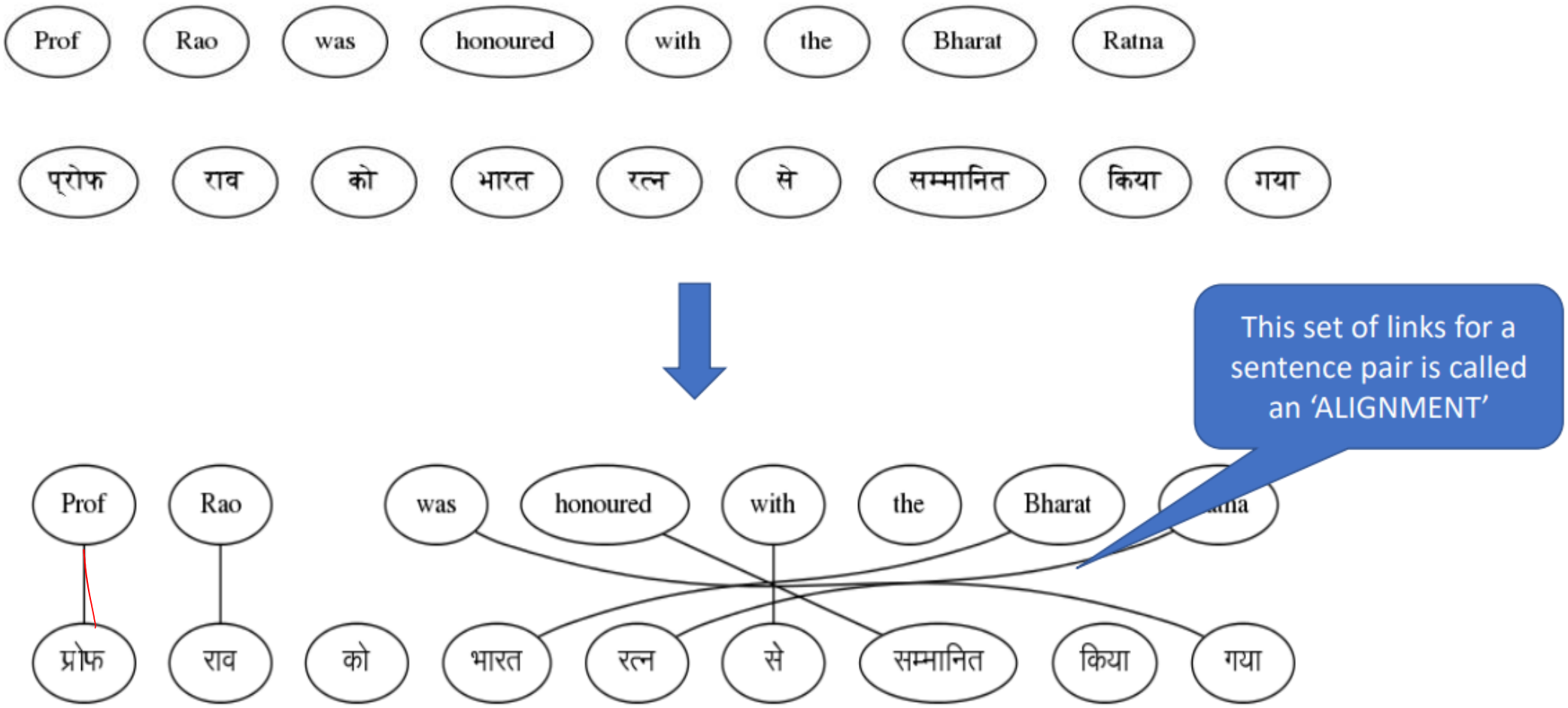
Parallel Corpus	
A boy is sitting in the kitchen	एक लडका रसोई मे बैठा है
A boy is playing tennis	एक लडका टेनिस खेल रहा है
A boy is sitting on a round table	एक लडका एक गोल मेज पर बैठा है
Some men are watching tennis	कुछ आदमी टेनिस देख रहे है
A girl is holding a black book	एक लडकी ने एक काली किताब पकडी 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 बैठी है

Learning alignment for SMT

- Question: How to learn translation model from the parallel corpus?
  - Break it down further: Introduce a latent variable into the model:

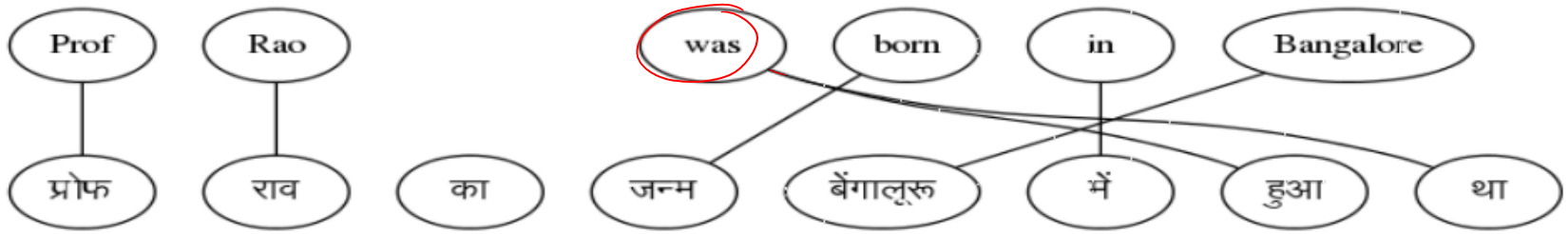
$$P(x, a|y)$$

  - where a is the alignment, i.e. word-level correspondence between source sentence x and target sentence y


Prof
Rao
was
honoured
with
the
Bharat
Ratna
प्रोफ
राव
को
भारत
रत्न
से
सम्मानित
किया
गया
This set of links for a sentence pair is called an 'ALIGNMENT'
Prof
Rao
was
honoured
with
the
Bharat
प्रोफ
राव
को
भारत
रत्न
से
सम्मानित
किया
गया

What is alignment?

- Alignment is the correspondence between particular words in the translated sentence pair.
  - Typological differences between languages lead to complicated alignments!
  - Note: Some words have no counterpart



Alignment can be many-to-one

Alignment can be one-to-many

Alignment can be many-to-many (phrase-level)

Learning alignment for SMT

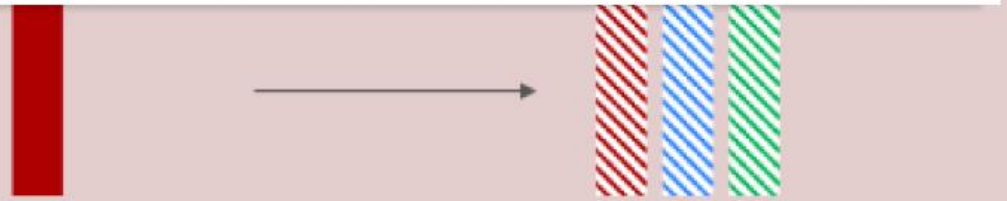
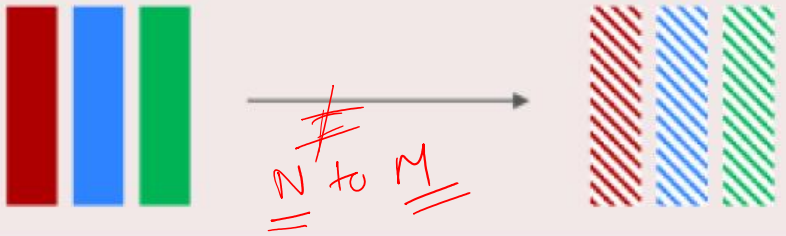
- We learn P(x,a|y) as a combination of many factors, including:
  - Probability of particular words aligning (also depends on position in sent)
  - Probability of particular words having particular fertility (number of corresponding words)
- Alignments '*a*' are latent variables: They aren't explicitly specified in the data!

What is Neural Machine Translation?

- Neural Machine Translation (NMT) is a way to do Machine Translation with a single neural network

- The neural network architecture is called sequence-to-sequence (aka seq2seq) and it involves two RNNs.

Types of RNN

1-to-1: single input and single output	1-to-N: single input and sequence output
	
A RNN models sequential interactions through a hidden state, or memory. It can take up to N inputs and produce up to N outputs.	An input could be a single image, and the output could be a sequence of words corresponding to the description of the image.
N-to-1: sequence input and single output	**N-to-N: sequence input and sequence output**
	
An input could be a sentence, and the output is the sentiment classification of the sentence	An input sequence may be a sentence with the outputs being the part-of-speech tag for each word

- Each rectangle represents a feature vector

Recall the Language Model

Generating Novel text with a n-gram Language Model

- You can also use a Language Model to generate text.

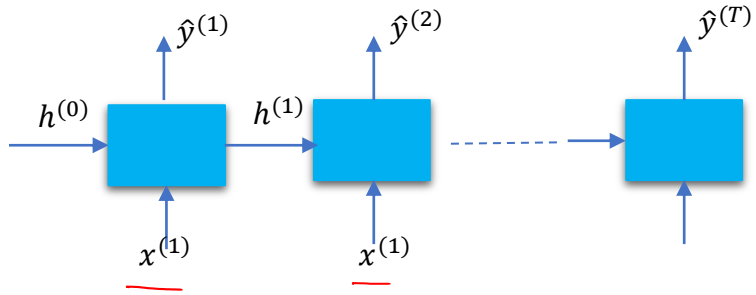
today the _____

company	0.153
bank	0.153
price	0.077
italian	0.039
emirate	0.039

Of	0.308
for	0.050
it	0.046
To	0.046
is	0.031

the	0.072
18	0.043
oil	0.043
its	0.036
gold	0.018

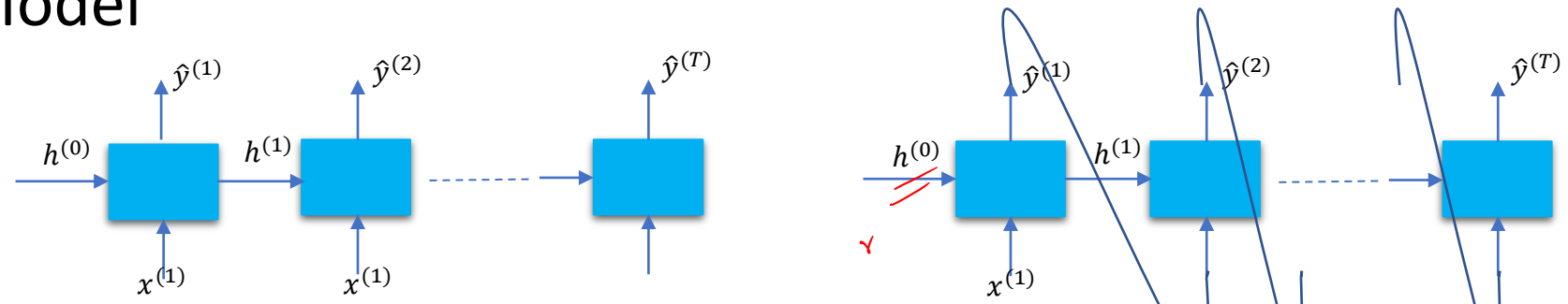
...



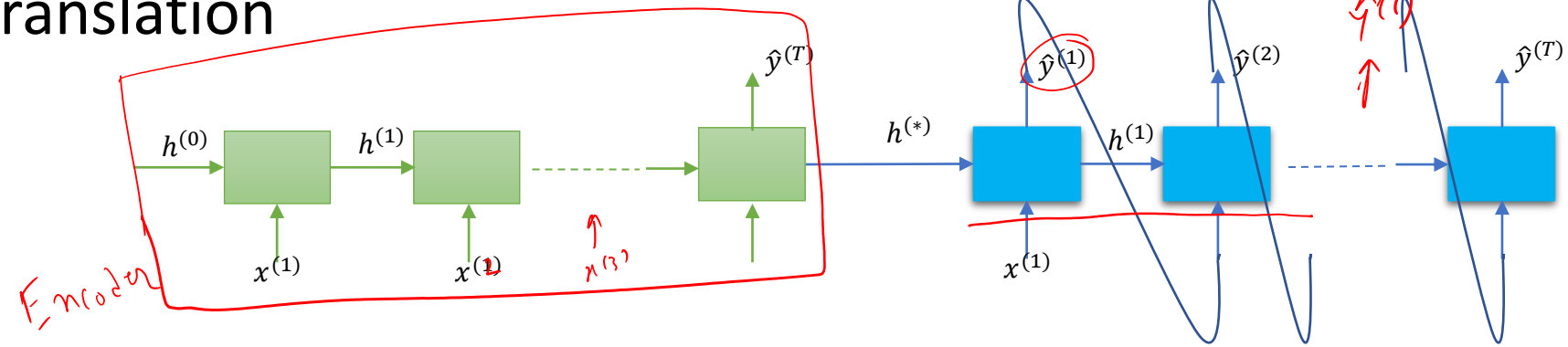
today the price of gold per ton , while production of shoe lasts and shoe industry , the bank intervened just after it considered and rejected an imf demand to rebuild depleted european stocks , sept 30 end primary 76 cts a share .

LM vs MT

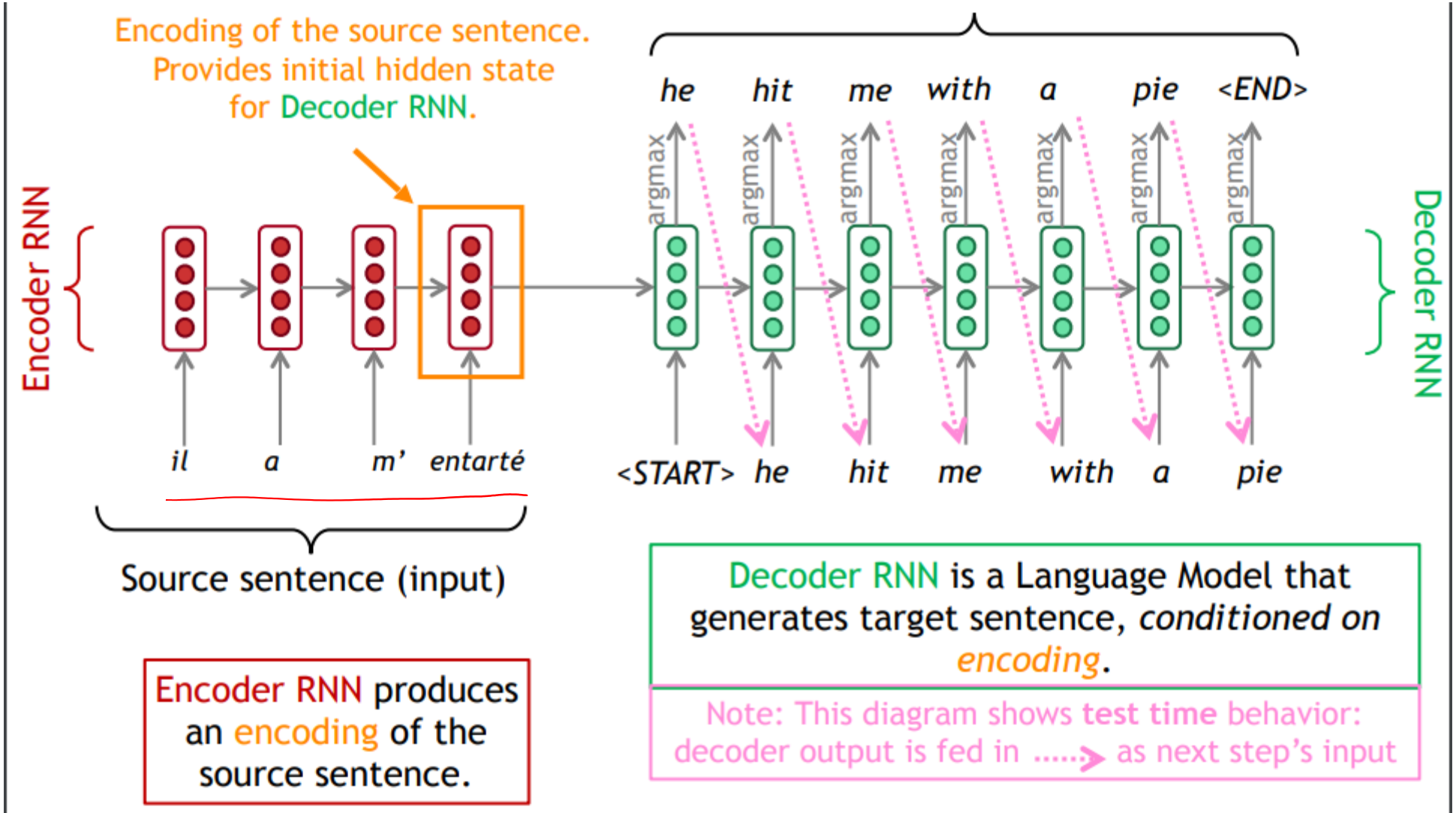
- Language Model



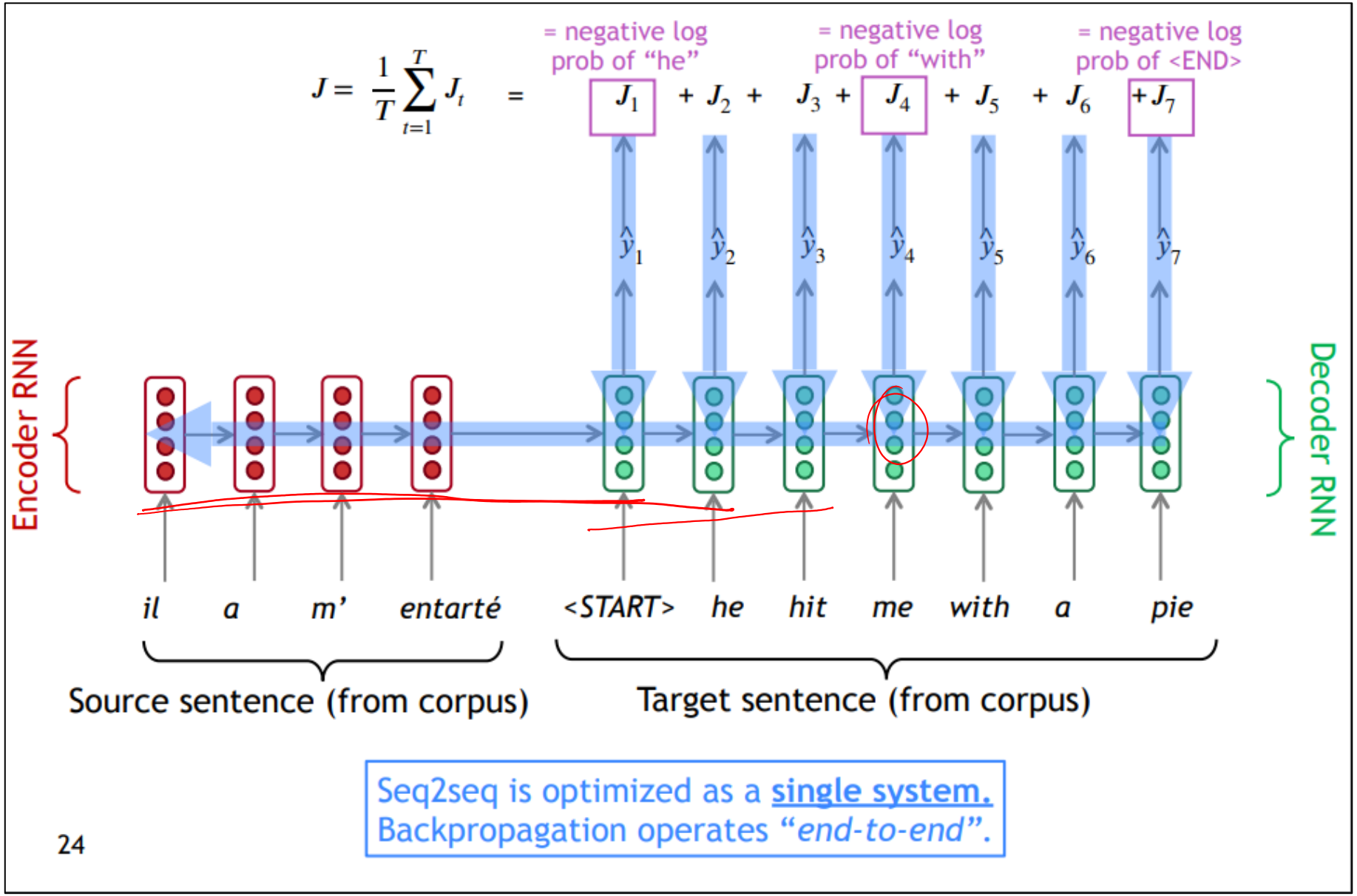
- Machine Translation



Neural Seq-Seq Generation Model



Training a Neural Machine Translation system



Exhaustive search decoding

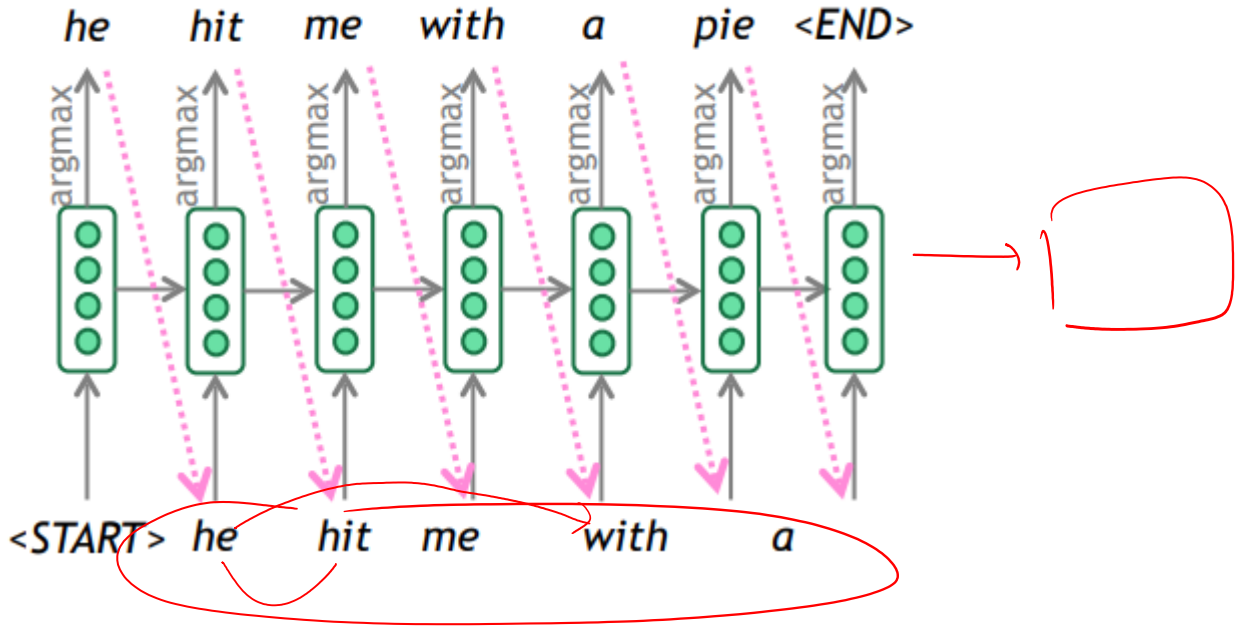
- Ideally we want to find a (length T) translation y that maximizes

$$P(y|x) = P(y_1|x)\,P(y_2|y_1,x)\,P(y_3|y_1,y_2,x)\ldots,P(y_T|y_1,\ldots,y_{T-1},x)$$
$$= \prod_{t=1}^{T} P(y_t|y_1,\ldots,y_{t-1},x)$$

- We could try computing all possible sequences y
  - This means that on each step t of the decoder, we're tracking V^t possible partial translations, where V is vocab size
  - This $O(V^T)$ complexity is far too expensive!

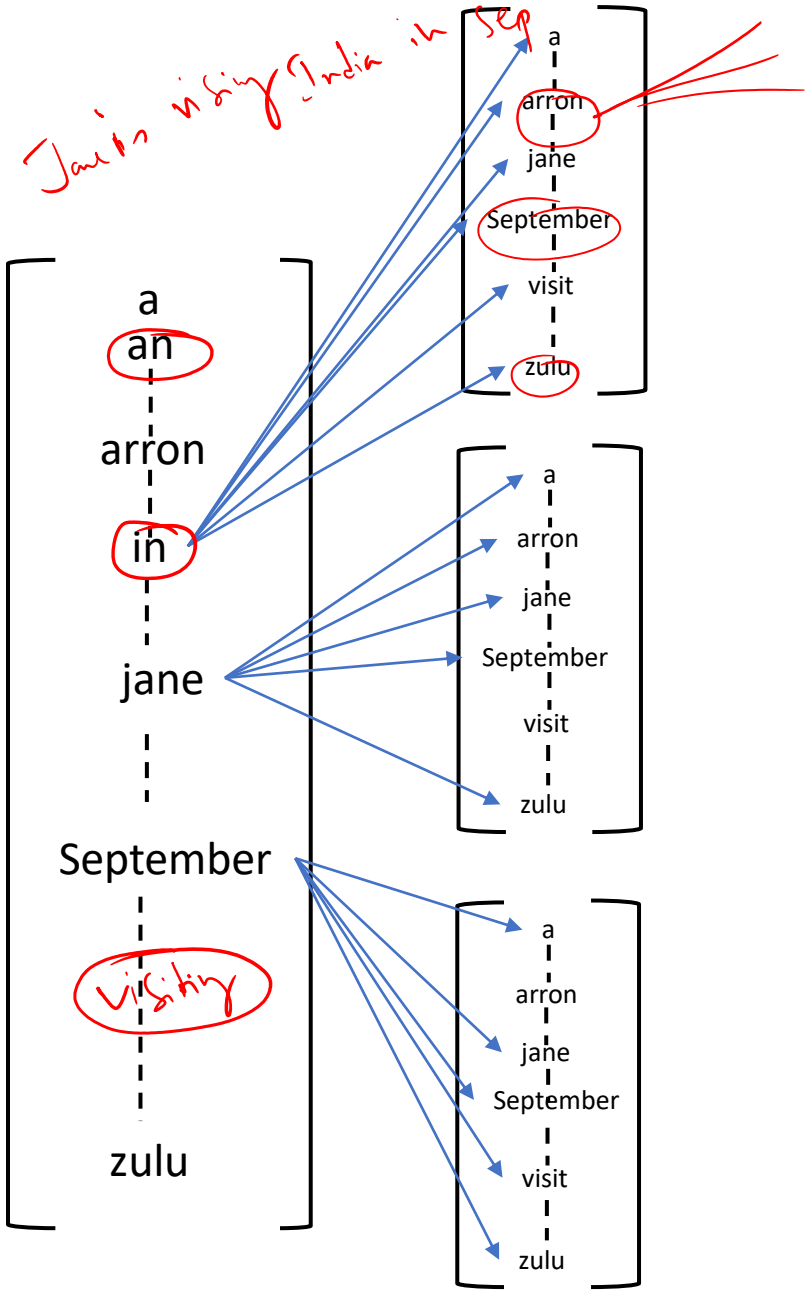
Greedy decoding

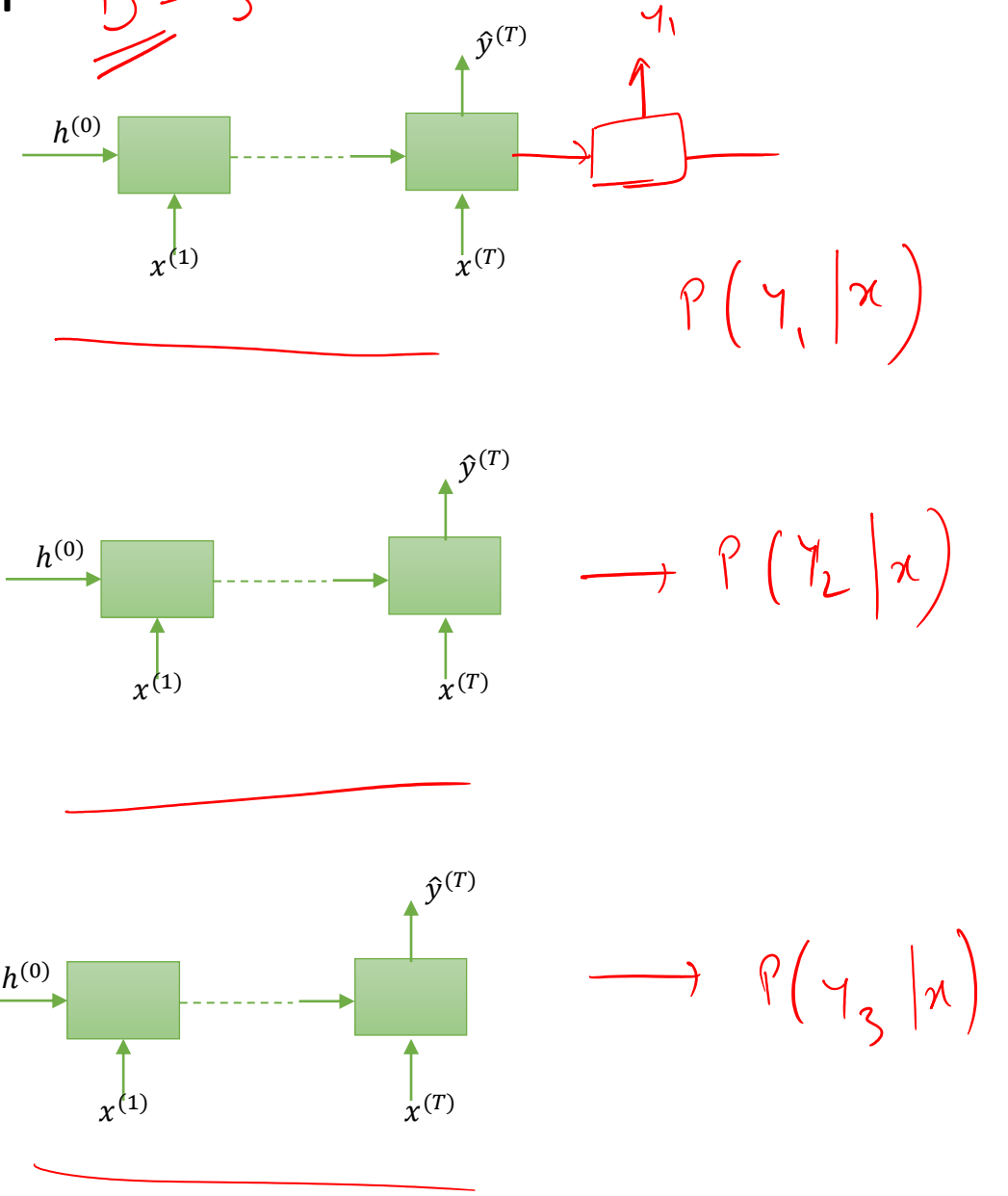
- We saw how to generate (or "decode") the target sentence by taking argmax on each step of the decoder



- This is greedy decoding (take most probable word on each step)
- Problems with this method?

Beam Search Algorithm



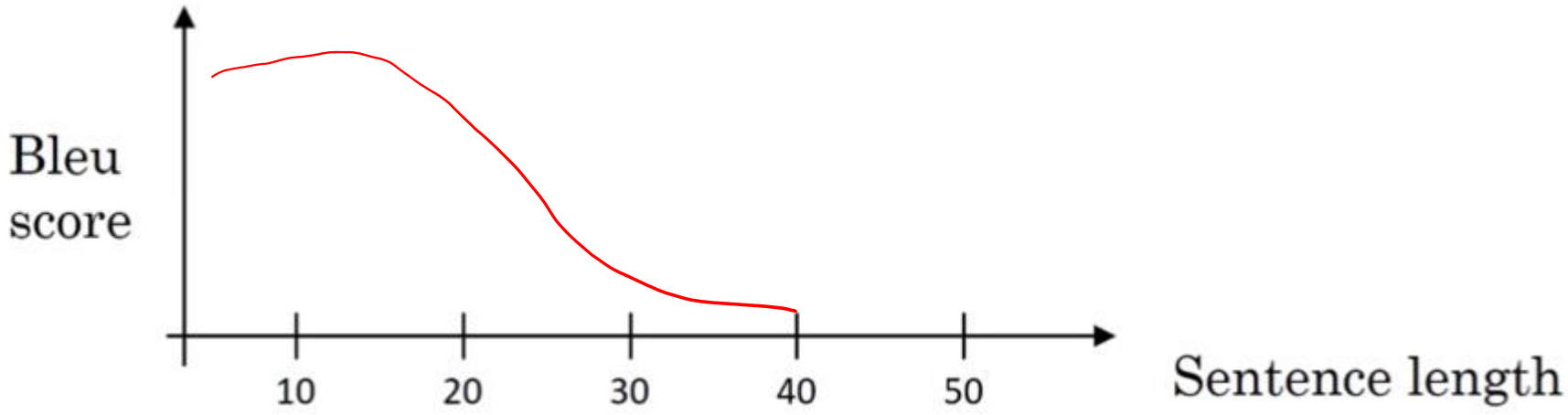


Jane visits Africa in September. <EOS>

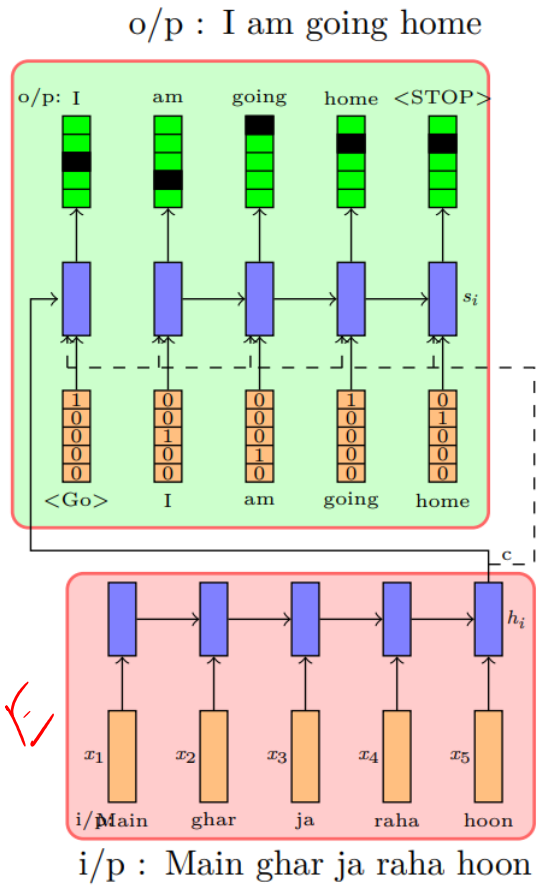
The problem of Long Sequences

His long, sleek hair was tied in a tight braid, and despite the cold and wind he wore only a long-sleeved sweater that hugged the muscles of his arms and shoulders beneath a down vest.

उसके लंबे, चिकने बाल एक तंग चोटी में बंधे हुए थे, और ठंड और हवा के बावजूद उसने केवल एक लंबी बाजू का स्वेटर पहना था जो नीचे की बनियान के नीचे उसकी बाहों और कंधों की मांसपेशियों को गले लगाता था।



Attention Model



- Let us motivate the task of attention with the help of MT
- The encoder reads the sentences only once and encodes it
- At each timestep the decoder uses this embedding to produce a new word
- Is this how humans translate a sentence ?

Humans try to produce each word in the output by focusing only on certain words in the input

Attention Model

o/p : I am going home

t_1 : [1 0 0 0 0]

i/p : Main ghar ja raha hoon

- Humans try to produce each word in the output by focusing only on certain words in the input
- Essentially at each time step we come up with a distribution on the input words

Attention Model

o/p : I am going home

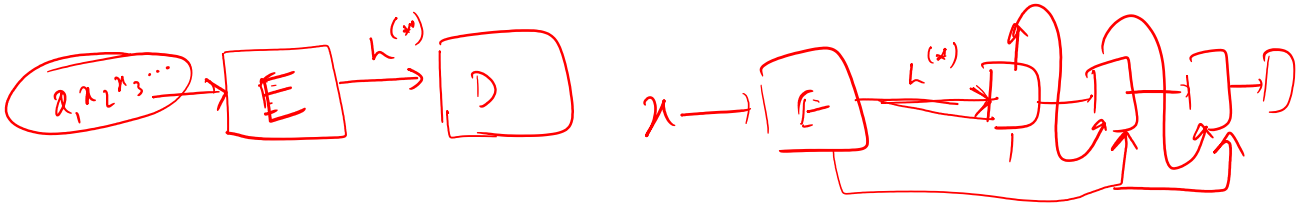
t_1 : [1 0 0 0 0]

t_2 : [0 0 0 0 1]

i/p : Main ghar ja raha hoon

- Humans try to produce each word in the output by focusing only on certain words in the input
- Essentially at each time step we come up with a distribution on the input words

Attention Model



o/p : I am going home

t_1 : [1 0 0 0 0]

t_2 : [0 0 0 0 1]

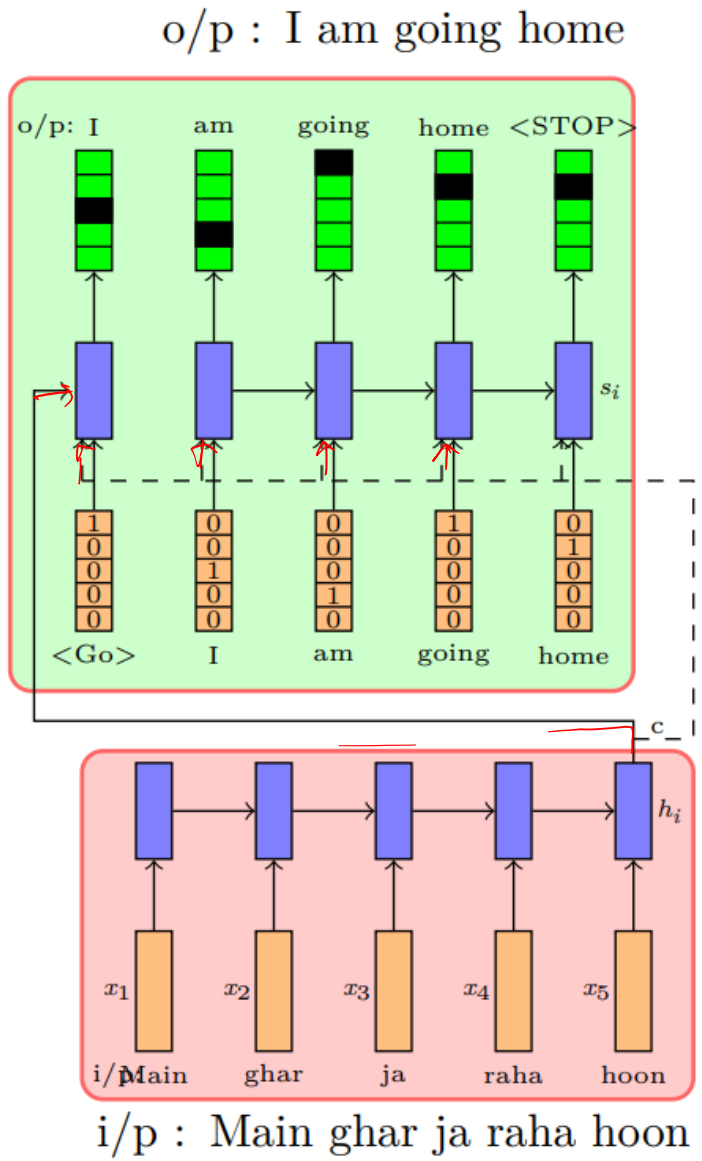
t_3 : [0 0 0.5 0.5 0]

i/p : Main ghar ja raha hoon

- Humans try to produce each word in the output by focusing only on certain words in the input
- Essentially at each time step we come up with a distribution on the input words
- This distribution tells us how much attention to pay to each input words at each time step
- Ideally, at each time-step we should feed only this relevant information (i.e. encodings of relevant words) to the decoder

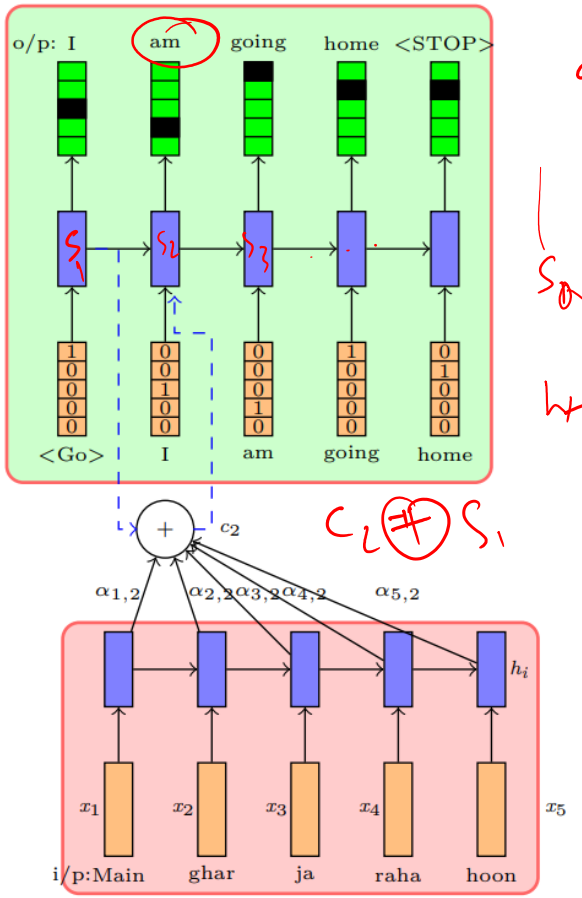
Attention Model

- Let us revisit the decoder that we have seen so far
- We either feed in the encoder information only once(at s_0)
- Or we feed the same encoder information at each time step



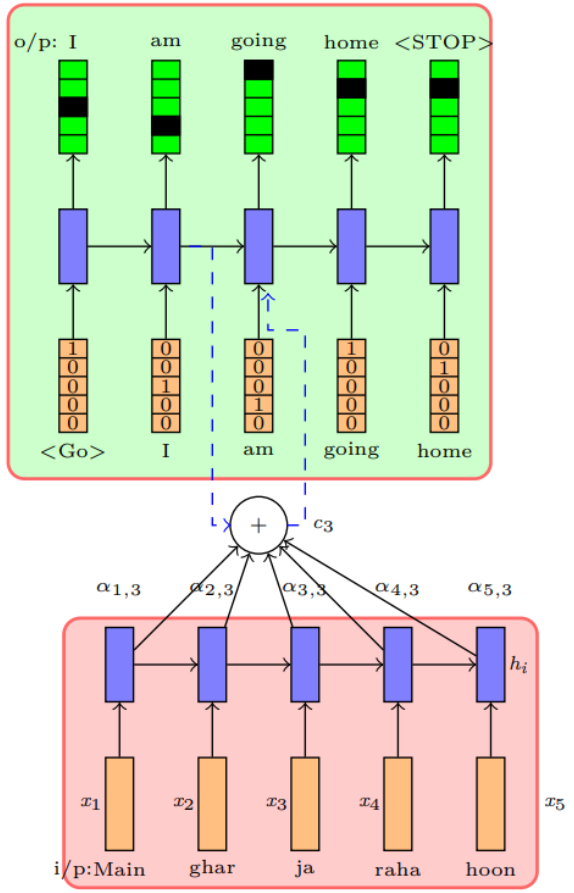
Attention Model

- We could just take a weighted average of the corresponding word representations and feed it to the decoder



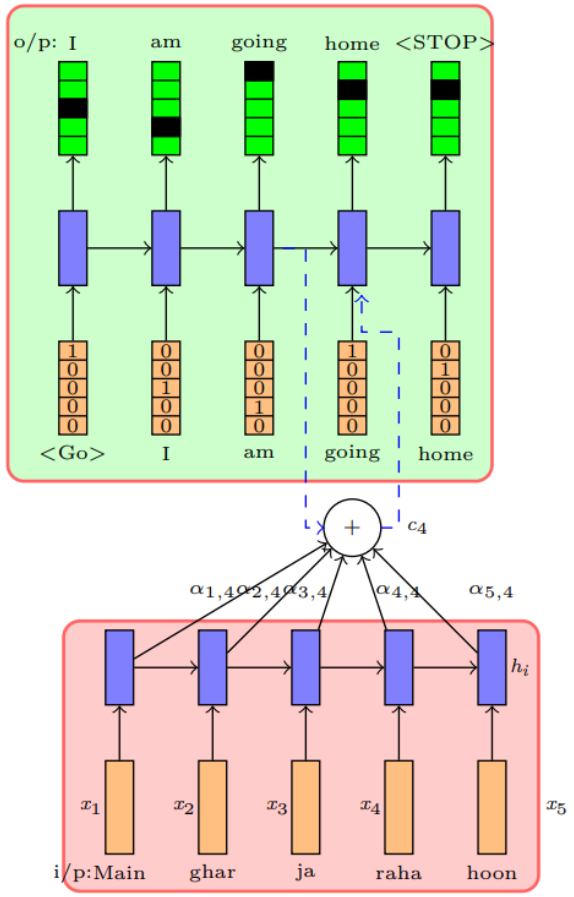
Attention Model

- We could just take a weighted average of the corresponding word representations and feed it to the decoder
- For example at timestep 3, we can just take a weighted average of the representations of ‘ja’ and ‘raha’



Attention Model

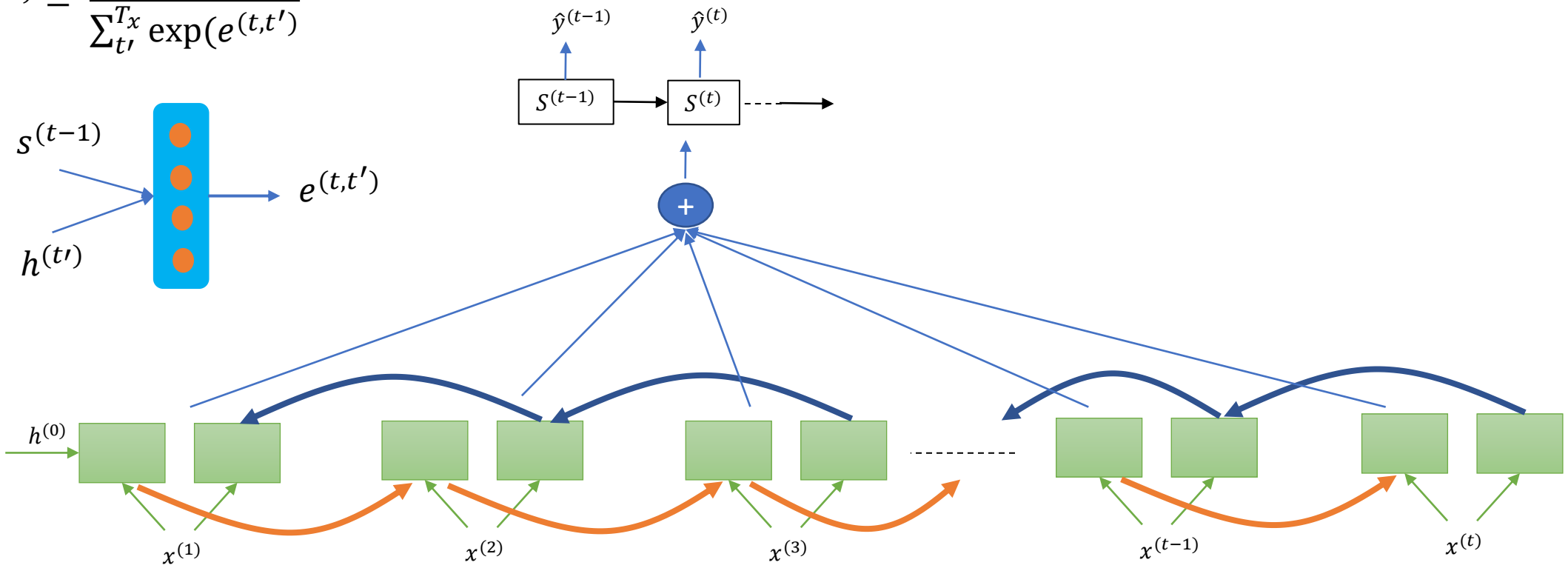
- We could just take a weighted average of the corresponding word representations and feed it to the decoder
- For example at timestep 3, we can just take a weighted average of the representations of ‘ja’ and ‘raha’
- Intuitively this should work better because we are not overloading the decoder with irrelevant information (about words that do not matter at this time step)



Attention Model

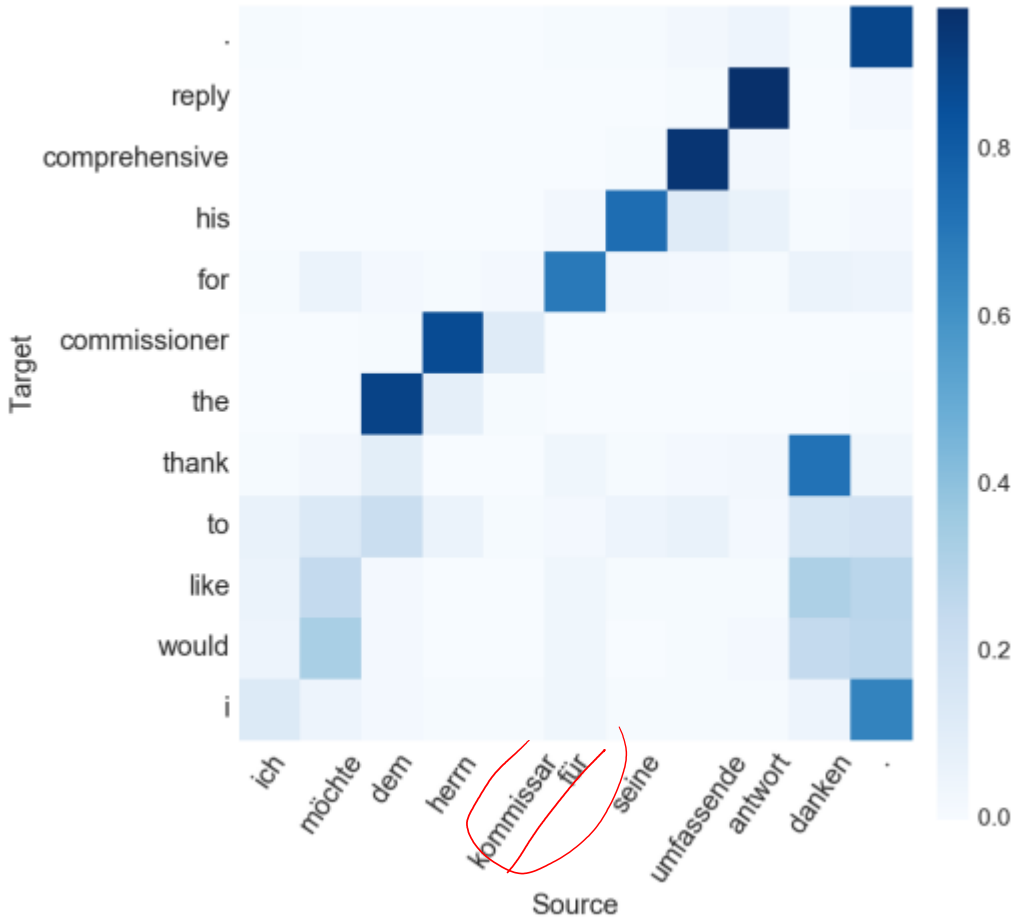
$\alpha^{(t,t')} = amount\ of\ attention\ y^{(t)}\ should\ pay\ to\ h^{(t')}$

$$\alpha^{(t,t')} = \frac{\exp(e^{(t,t')}}{\sum_{t'}^{T_x} \exp(e^{(t,t')}}$$



Attention Visualization

Machine Translation



Sentiment Classification