

Recap: ENC-DEC, Att. Mechanism.

① RNN prev-words
7-16

② LSTM 70-100

③ GRU 70-100

④ ENC-DEC 1000 words
& Att. Mech

Drawback of Attention Mechanism:

eg: h_{t1} h_{t2} h_{t3} h_{t4} h_{t5} (Sequential processing)
Rahul is a good boy
 X_1 X_2 X_3 X_4 X_5

i.e. X_2 can't be sent before X_1
 X_3 ——— || ——— X_2
 X_4 ——— 4 ——— X_3
 X_5 ——— 4 ——— X_4


Today we hv super powerful GPU

★ Transformers overcome this drawback.



You
1 chunk / 1 min
 $\therefore 30 \text{ chunks} \rightarrow 30 \text{ min}$

Ravan (Parallel Processing)


 $\therefore 30 \text{ chunks} \rightarrow 3 \text{ min}$

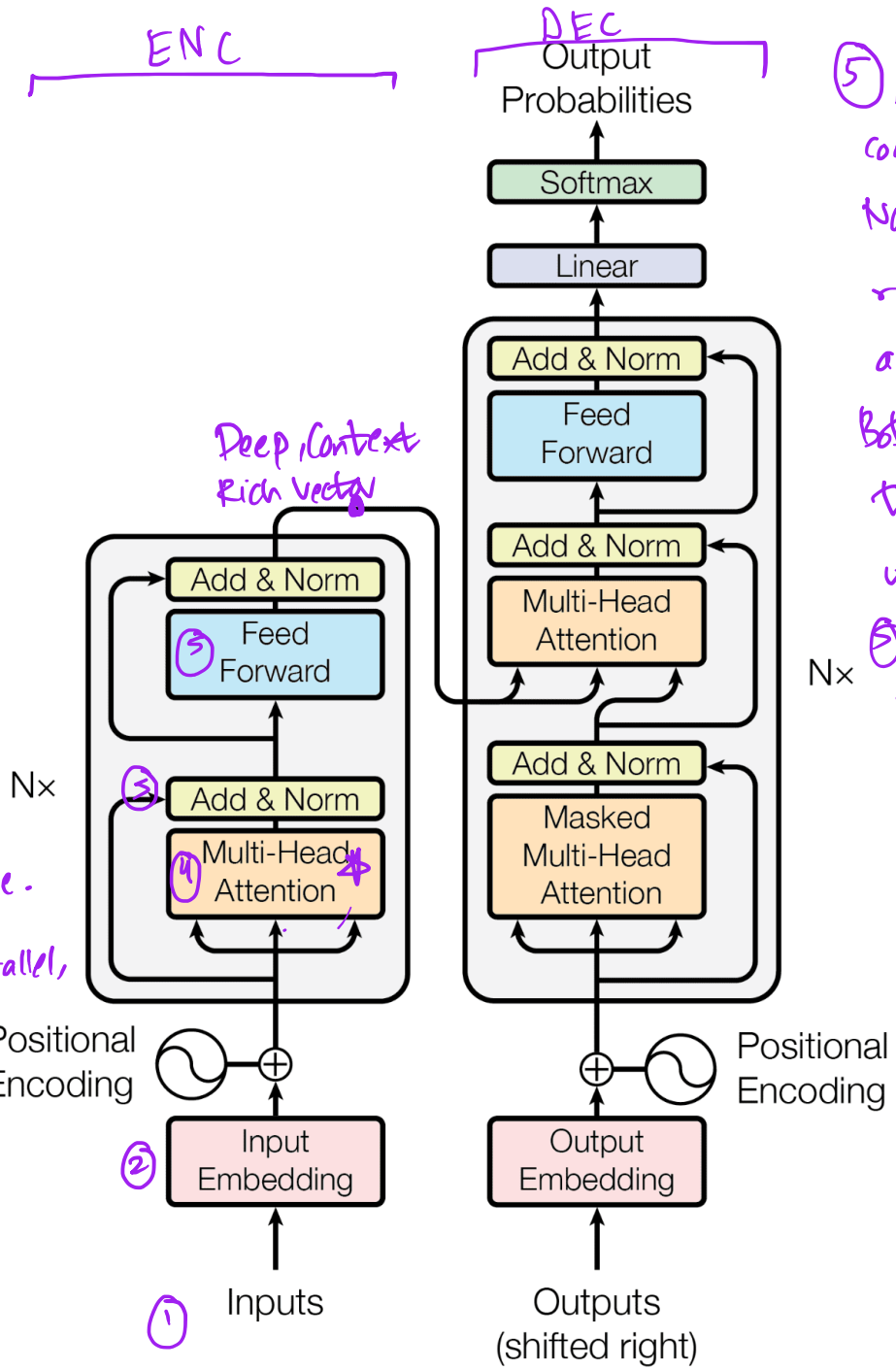
ENC - HIN

Rahul¹ is² a³ good⁴ boy⁵

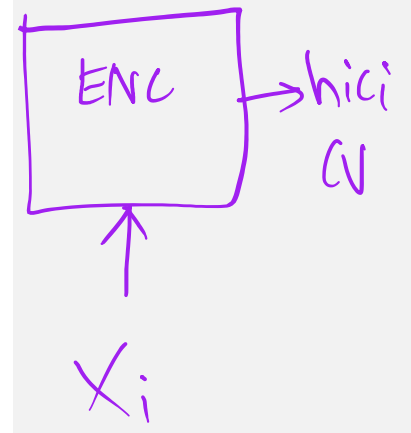
2. ENC 3. DEC 4. HIN 5. HIN

- 1 ['Rahul', 'is', 'a', 'good', 'boy']
- 2 Rahul → [0.2, 0.2, -1.1, ..., 0.5]
- 3 good → [0.6, 0.0, ..., -0.2]
- ...

- 4 Each word attends to every other word & learns how important they are to each other.
 - 5 Self Attention - means a word can look at other words in the same sentence.
 - 6 Multi-head - it does this multiple times in parallel, with each head focussing on different relationships.
- W1 - Grammar W2 - meaning W3 - Distance
- eg: boy - attend to good, is → Rahul



- 5 Add - Adds the original input (residual connection) to the output of attention.
 - Norm - Applies Layer Normalization, which rescales the values to keep them within a stable range.
- Both help with Training Stability & ensure that the network doesn't drift into unstable values.
- 5 FFN - Simple N.N. (1 HL, AF, 10P)
- It makes each word vector refined - smarter, more contextual.



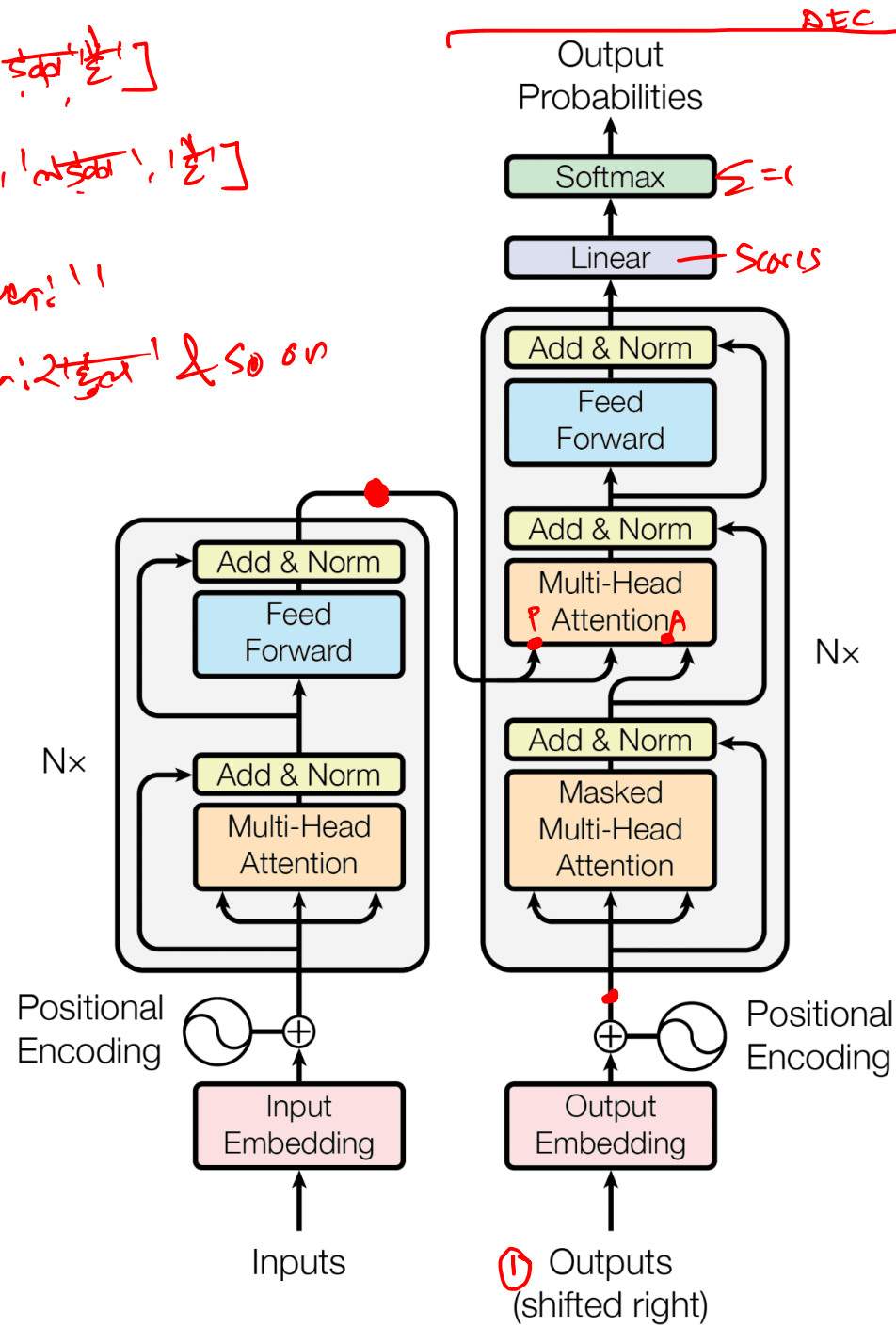
① Target: ['राहुल', '50', 'अच्छा', 'नहीं', 'है']

Shifted input: ['', 'राहुल', '50', 'अच्छा', 'नहीं', 'है']

Decoder

So model learns: Predict राहुल Given: ''

————— '' ——— 50 Given: राहुल & so on



The grass

The grass is

The grass is green.