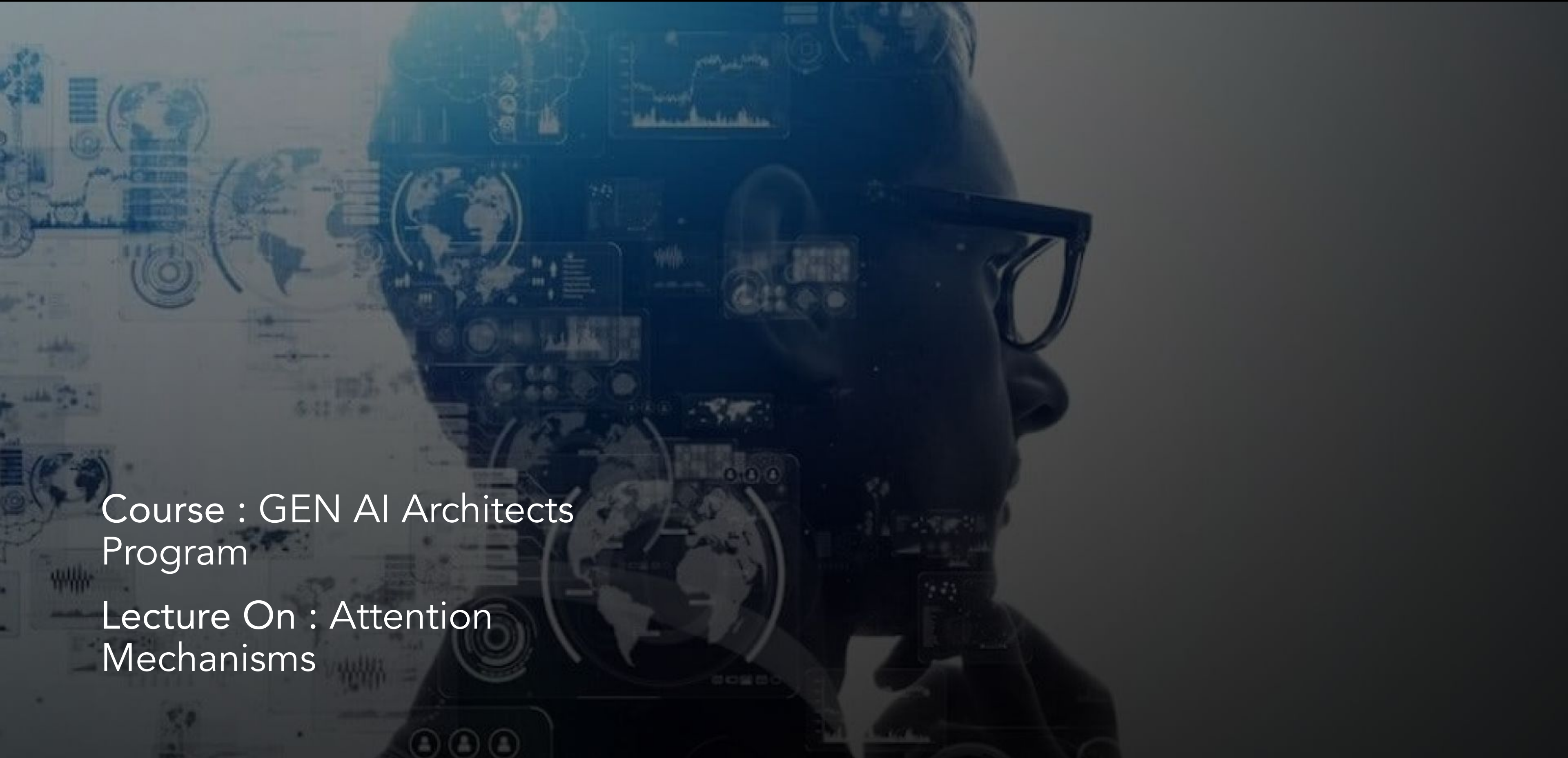
upGrad ENTERPRISE

GEN AI Architects Program - Hexaware

10th August, 2024



Course : GEN AI Architects Program

Lecture On : Attention Mechanisms

In Previous Sessions, we covered....

- Machine Learning- Supervised Techniques - Regression, classification
- Machine Learning- Unsupervised Techniques - Clustering & Dimensionality Reduction
- Basics of Deep Neural Networks
- Deep Learning with Tensorflow & Keras
- Advanced Programming for LLM Development
- Basics of NLP

upGrad ENTERPRISE

Today’s Agenda

01 Seq2Seq Modelling

02 Neural Machine Translation

03 Attention Models

Sequence modelling - Recap

What comes next?

- Why RNN?
- LSTM – Forget, Input and Output gates
- GRU – 2 gates

Why RNN?

- To deal with variable length sequence
- To maintain sequence order
- Keep track of long term dependencies
- To share parameters across the sequence

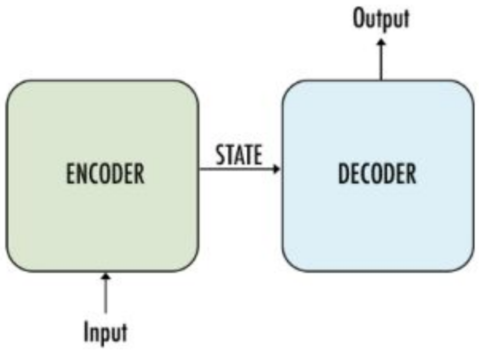
Language Translation

The objective is to convert a German sentence to its English counterpart using a Neural Machine Translation (NMT) system.

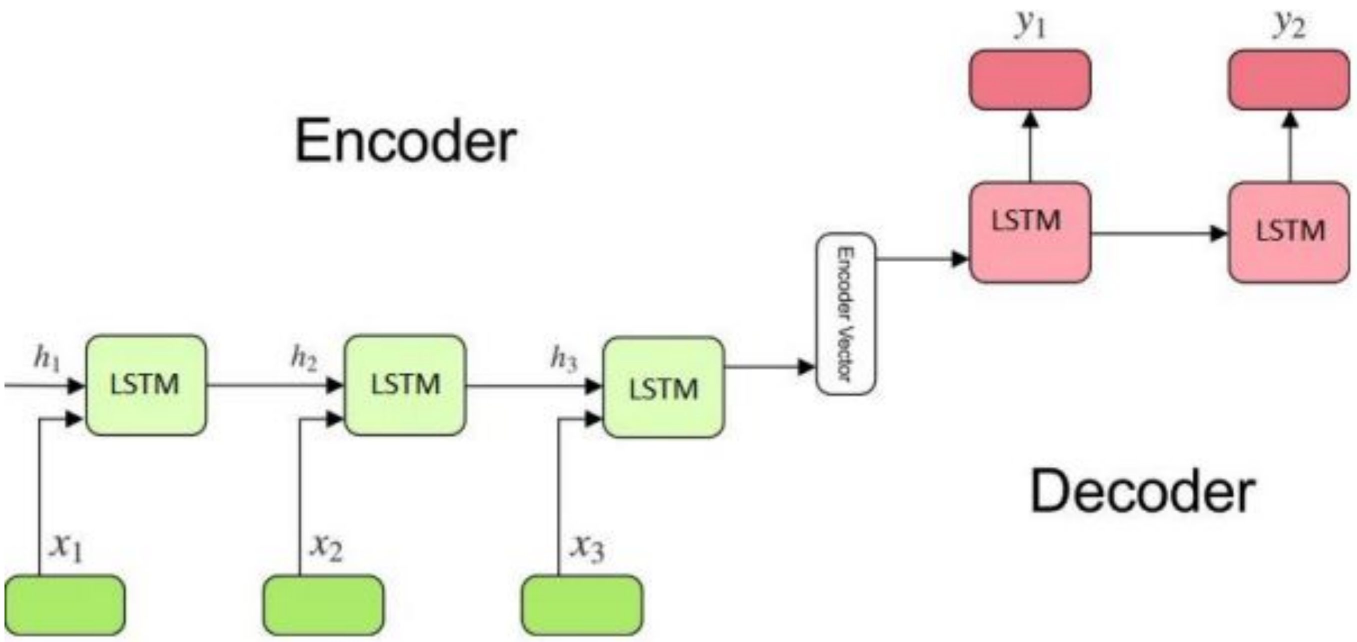
$(\text{Es regnet draußen})_{\text{German}} \rightarrow (\text{It's raining outside})_{\text{English}}$

Model Components

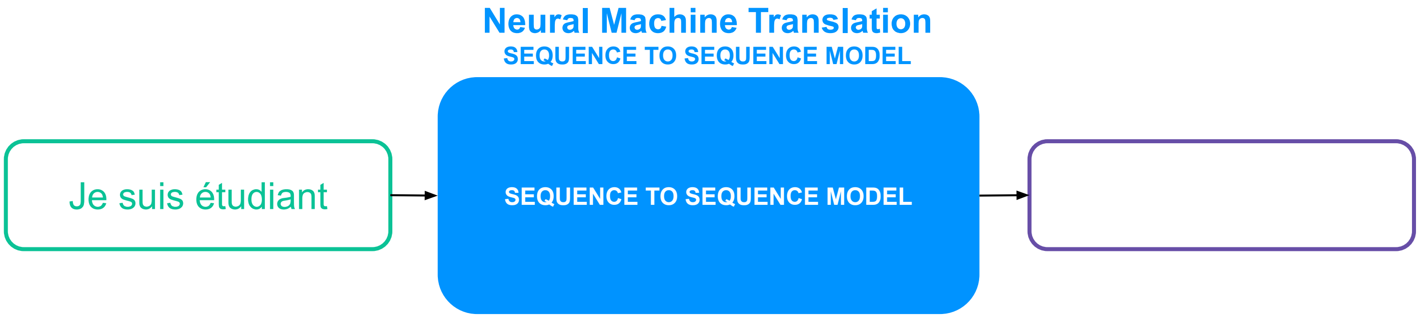
- A typical seq2seq model has 2 major components
  - a) an encoder
  - b) a decoder
- Both these parts are essentially two different RNN/ LSTM models combined into one giant network: Encoder Decoder



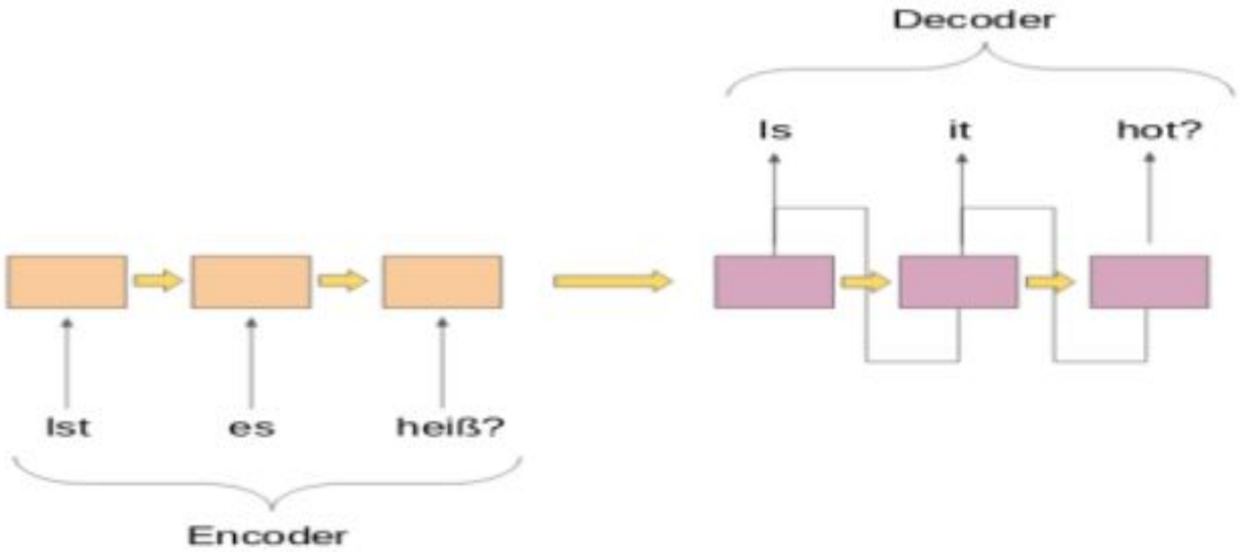
Encoder-Decoder



Encoder-Decoder



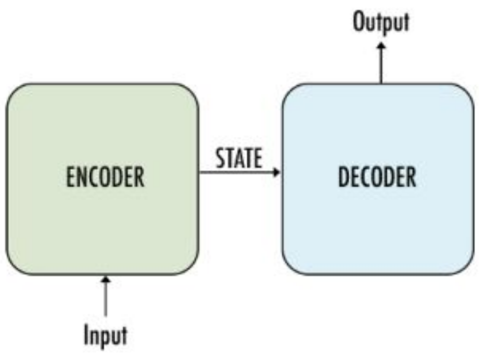
Encoder-Decoder



upGrad ENTERPRISE

NMT

- No need to know any rule about human language.
- Achieved better results than 20 years of work with Statistical Machine translation.

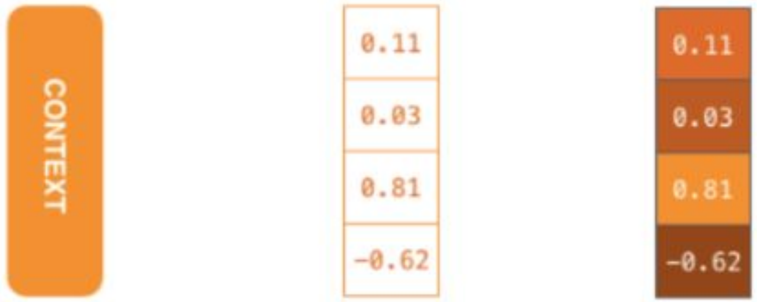


Sequence to Sequence (Seq2Seq) Model

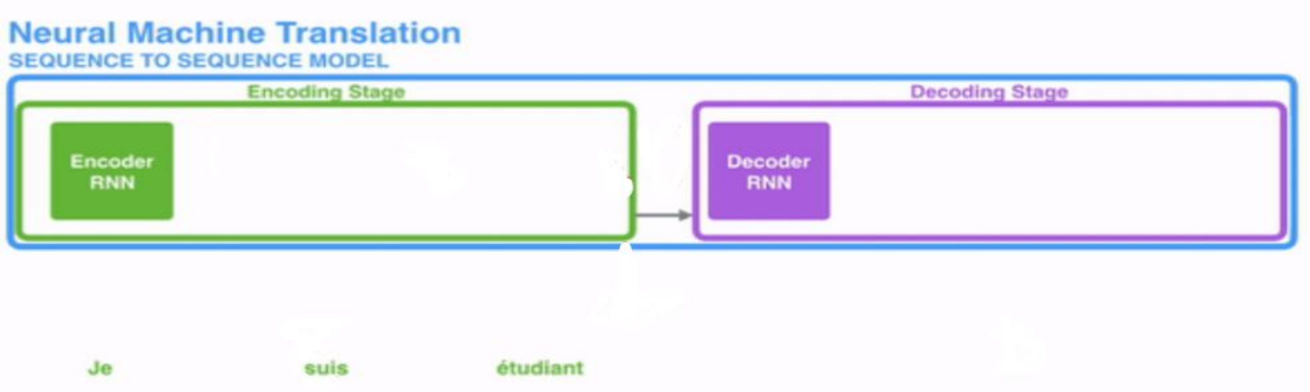
Input Sequence	Encoder	Output Encoding	Decoder	Output Sentence
"Let's" "go" "to" "Delhi"	Recurrent Neural Network	0.9512 0.0377 0.4927 0.6106	Recurrent Neural Network	"आओ" "हम" "दिल्ली" "जाएं"

Context Vector

- The context is a vector in the case of machine translation which basically represents the context information in a given sentence
- We can set the size of the context vector when we set up your model. It is basically the number of hidden units in the encoder RNN.
- These visualizations show a vector of size 4, but in real world applications the context vector would be of a size like 256, 512, or 1024.

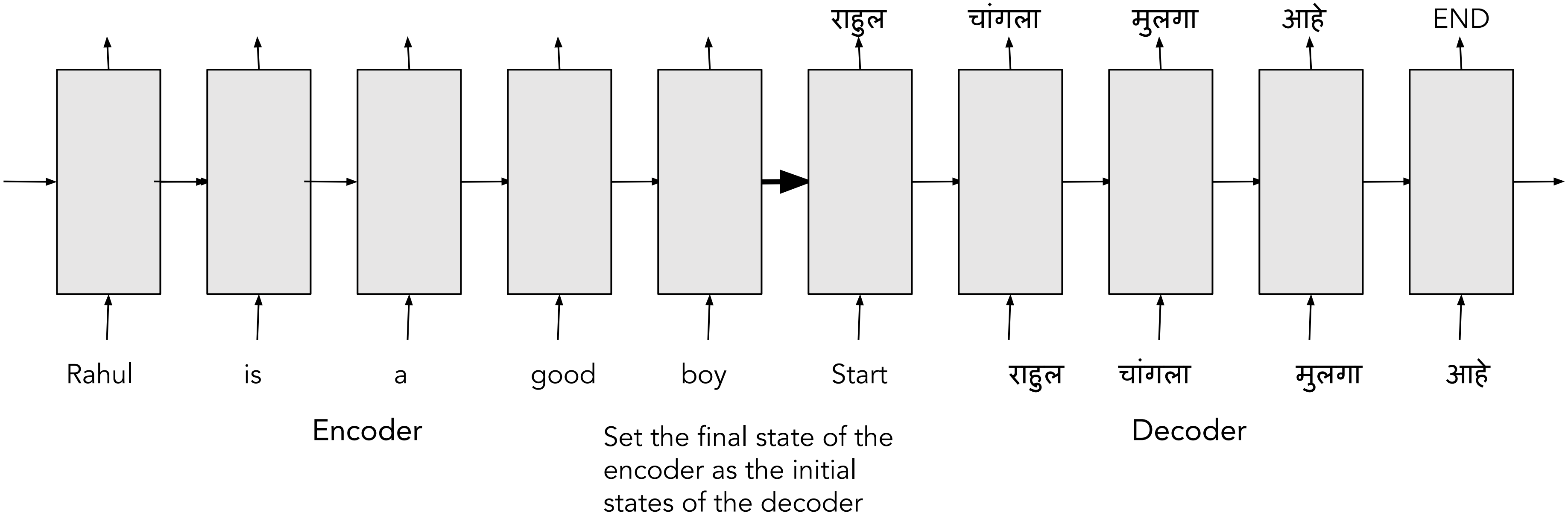


How it works?



- Encoder or decoder is that RNN/LSTM processing its inputs and generating an output for that time step.
- Since the encoder and decoder are both same units, each time step one of the units does some processing, updates its hidden state based on its inputs and previous inputs it has seen
- The decoder also maintains a hidden states that it passes from one time step to the next

Seq2Seq Model



The encoder is forced to compress the entire input sentence into a single(context) vector only

Seq2Seq Model

- Instead of discarding these intermediate states of the encoder, the attention utilizes them in order to construct the context vector for the decoder at different time steps Discard the encoder outputs

- The states and outputs at each time step,become the states and input respectively for the next time step

Attention Models

What's wrong with seq2seq models?

- The seq2seq models is normally composed of an encoder-decoder architecture, where the encoder processes the input sequence and encodes /compresses/summarizes the information into a context vector (also called as the "thought vector") of a fixed length.
- A critical and apparent disadvantage of this fixed-length context vector design is the incapability of the system to remember longer sequences.

Concept of Attention

- Bicycle example adding handle for better modelling
- When you predict “राहुल”, it's obvious that this name is the result of the word “Rahul” present in the input English sentence regardless of the rest of the sentence.
- We say that while predicting “राहुल”, we pay more attention to the word “Rahul” in the input sentence.
- Similarly while predicting the word “चांगला”, we pay more attention to the word “good” in the input sentence and so on.
- Hence the name “ATTENTION”.

The central idea behind Attention

- As human beings we are quickly able to understand these mappings between different parts of the input sequence and corresponding parts of the output sequence.

- It’s not that straightforward for neural networks to automatically detect these mappings.

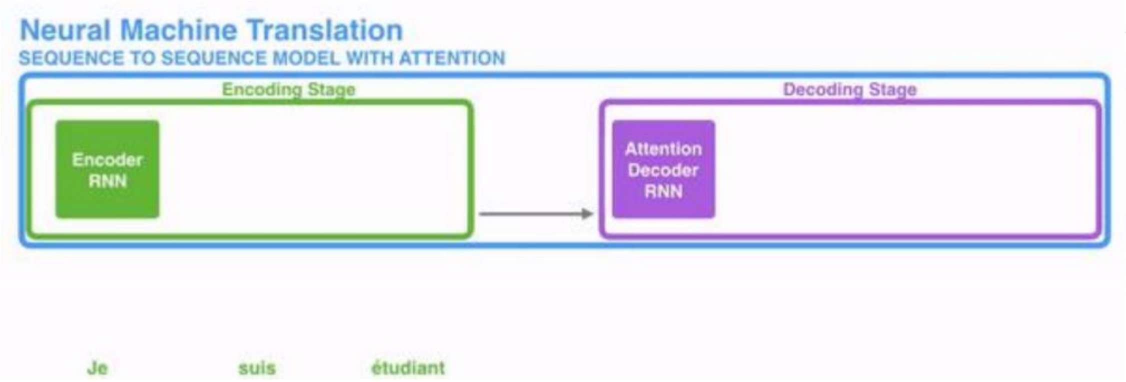
The central idea behind Attention

- Thus the Attention mechanism is developed to “learn” these mappings through Gradient Descent and Back-propagation.

- The central idea behind Attention is not to throw away those intermediate encoder states but to utilize all the states to construct the context vectors required by the decoder to generate the output sequence.

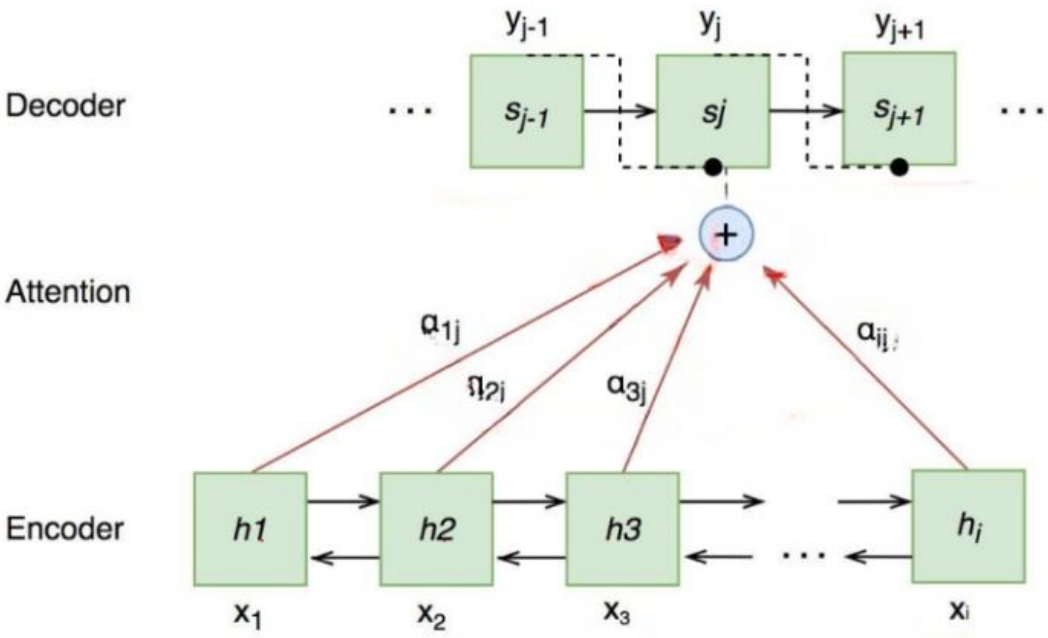
upGrad ENTERPRISE

NMT with attention mechanism



- An attention model differs from a classic sequence-to-sequence model in two main ways:
    - The encoder passes a lot more data to the decoder.
    - Instead of passing the last hidden state of the encoding stage, the encoder passes all the hidden states to the decoder

Attention based encoder-decoder



upGrad ENTERPRISE

Attention Mechanism

Attention decoder does an extra step before producing its output. In order to focus on the parts of the input that are relevant to this decoding time step, the decoder does the following:

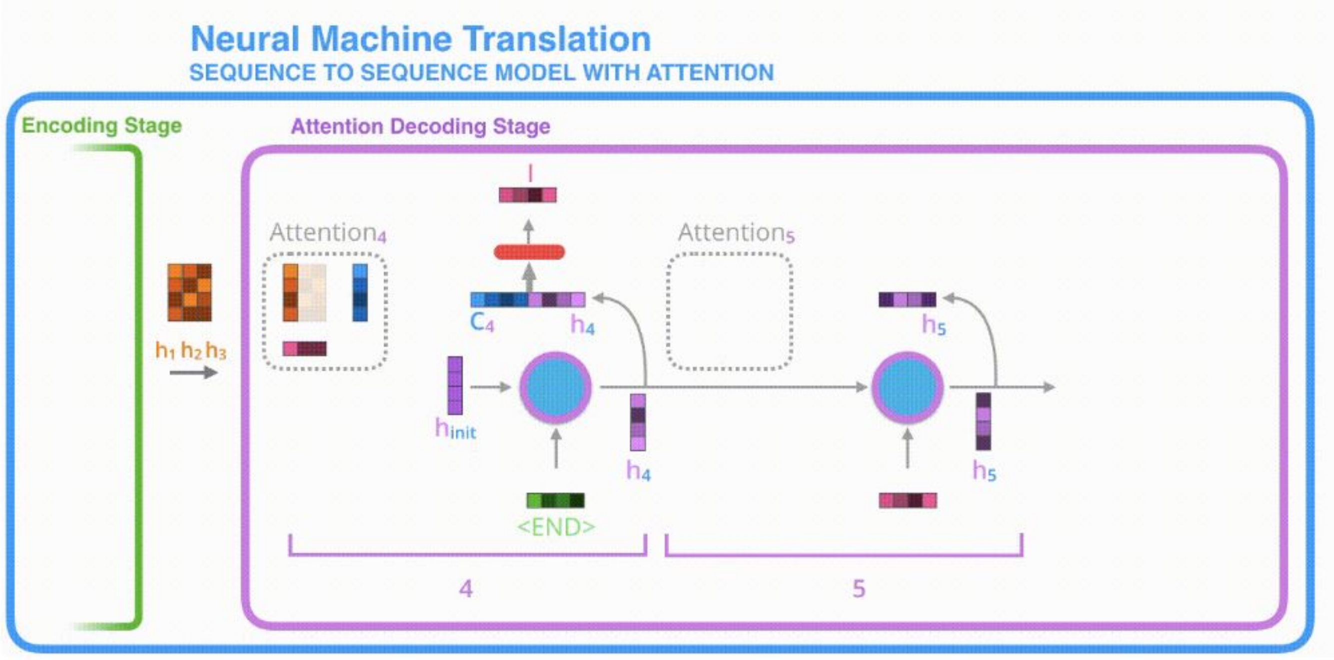
1. Look at the set of encoder hidden states it received – each encoder hidden states is **most associated with a certain word** in the

input sentence

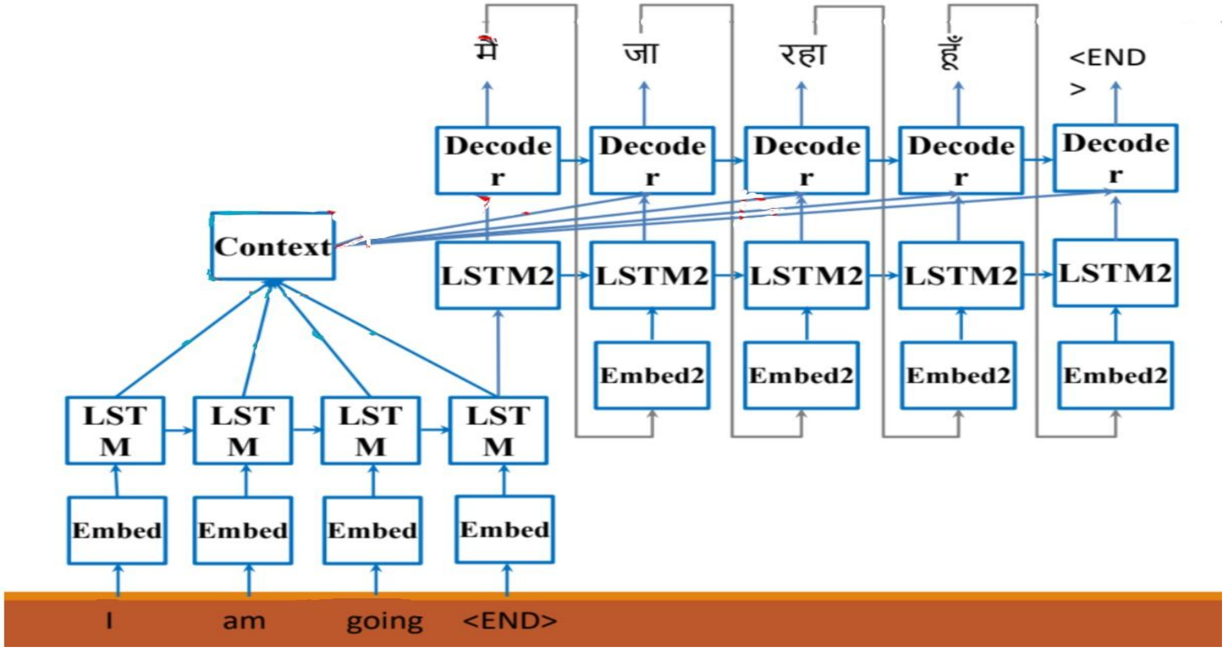
2. Give each **hidden states a score**

3. Multiply each hidden states by its softmaxed score, thus **amplifying hidden states** with high scores, and drowning out hidden states with low scores

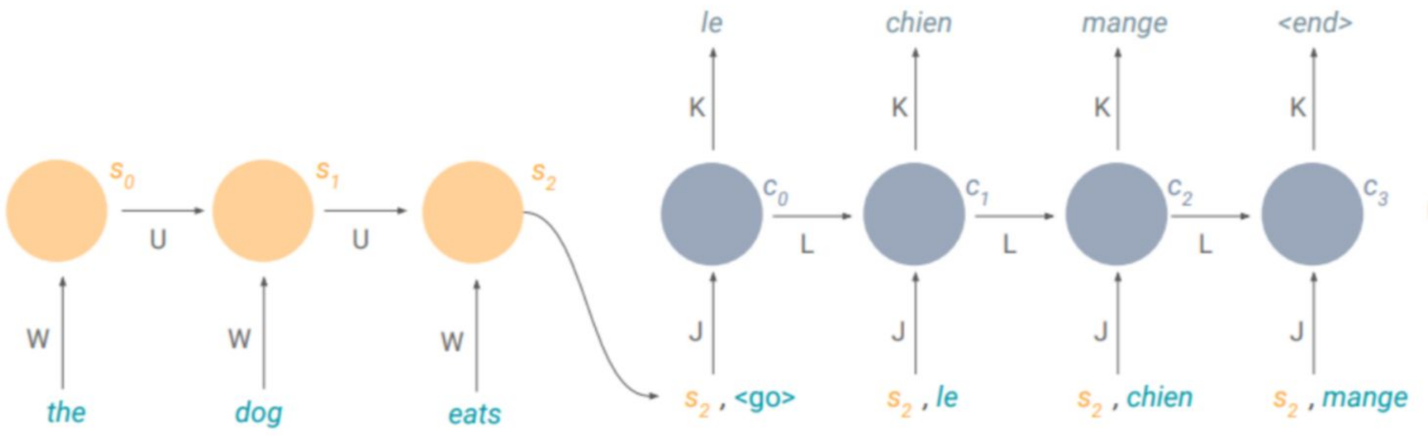
Attention Mechanism



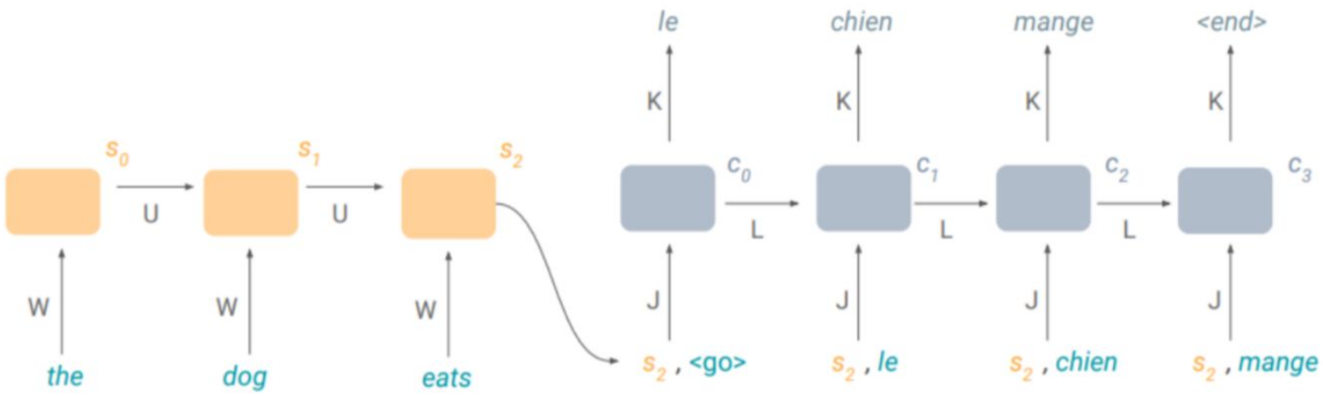
NMT with Attention - Total Architecture



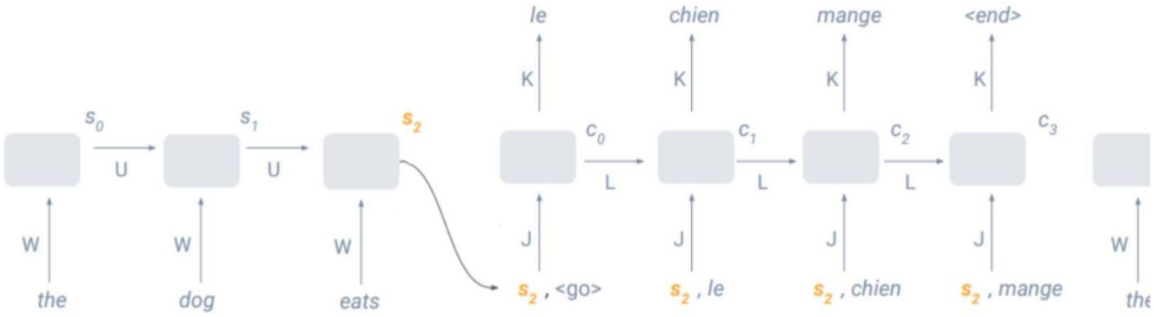
Machine Translation - Encoder, Decoder



Machine Translation - with LSTM cells

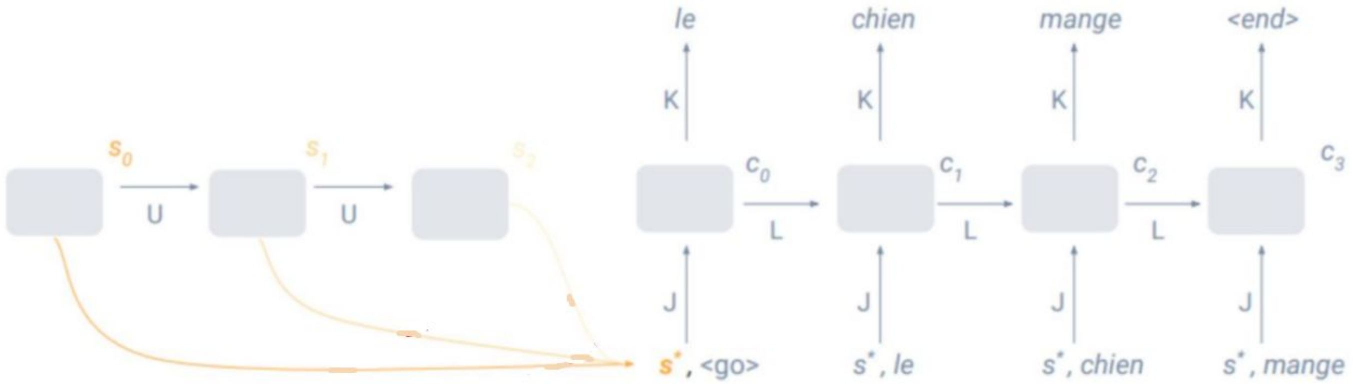


Single encoding is limiting



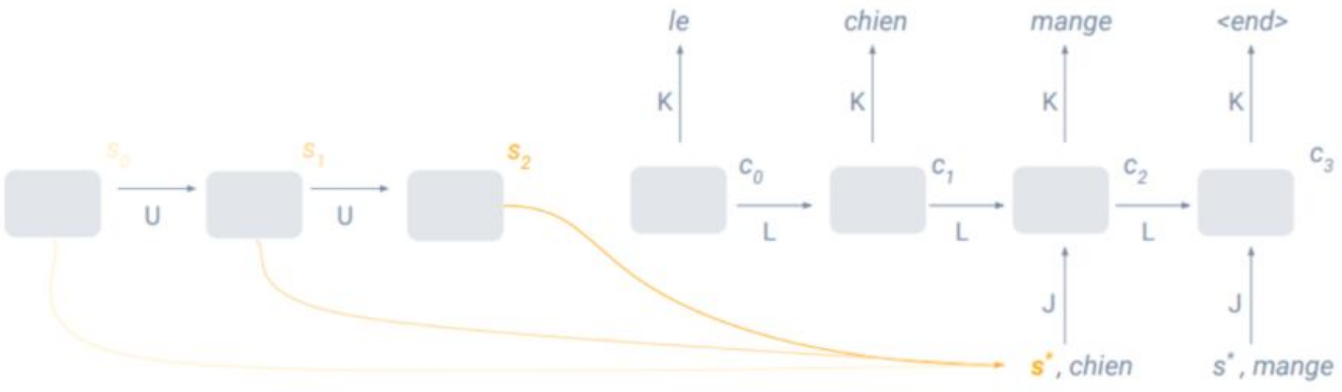
all the decoder knows about the input sentence is in one fixed length vector, s_2

Attend over all encoder states

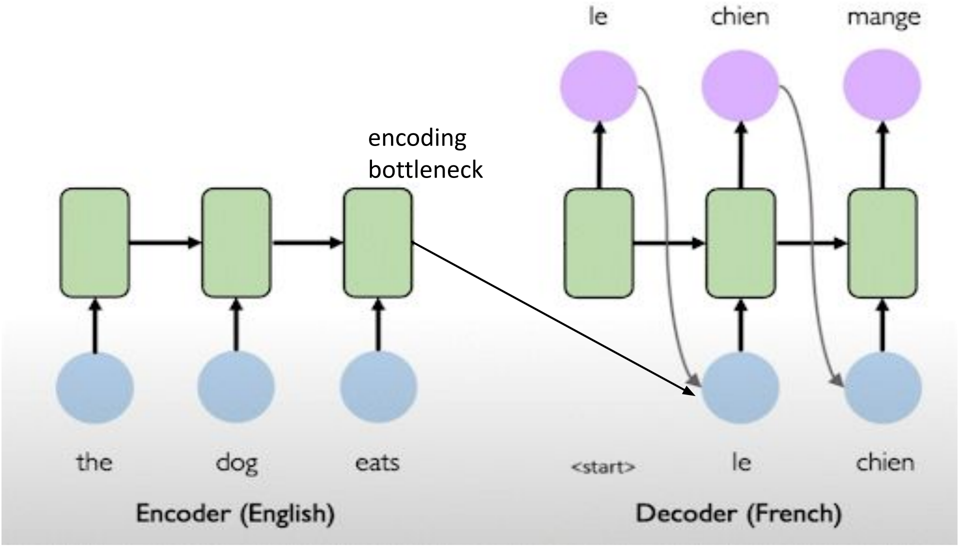


upGrad ENTERPRISE

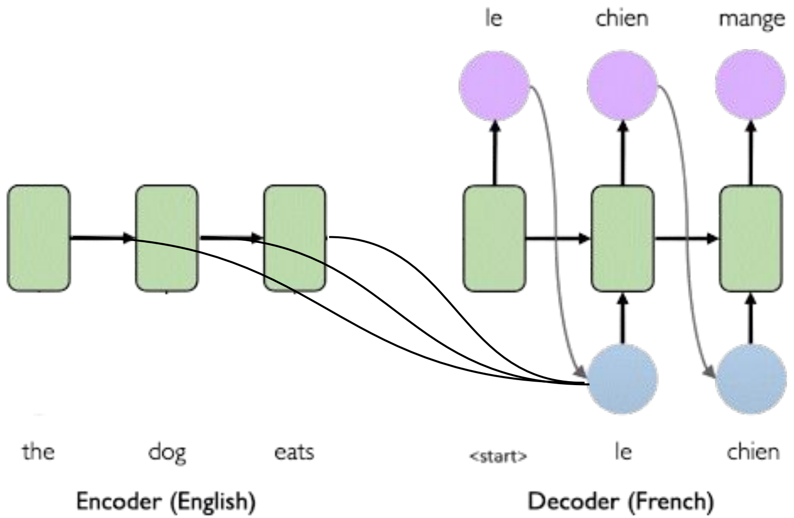
Attend over all encoder states



Attention Summary



Attention Summary



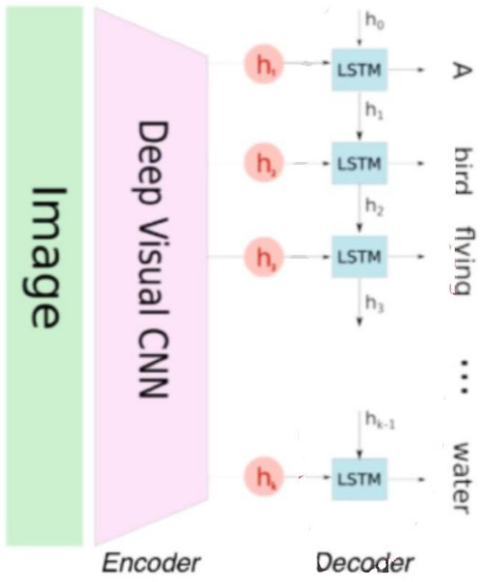
Evaluation Metrics

- BLEU – Bilingual Evaluation Understudy score
- This is for generated sentence to reference sentence
- Automatic evaluation of Machine translation
- NLTK: sentence_blue()
- Corpus_blue()

upGrad ENTERPRISE

Is attention limited to NMT type of applications?

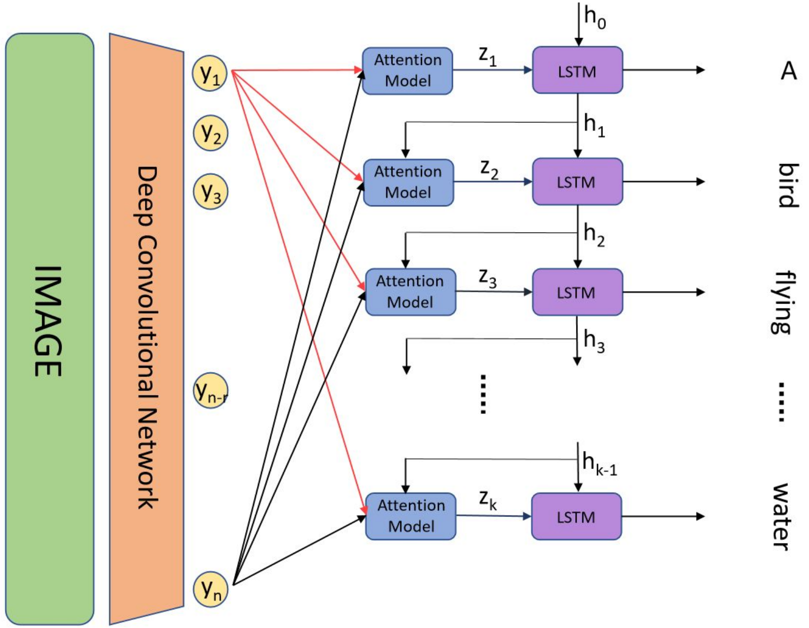
Attention to give title to the image



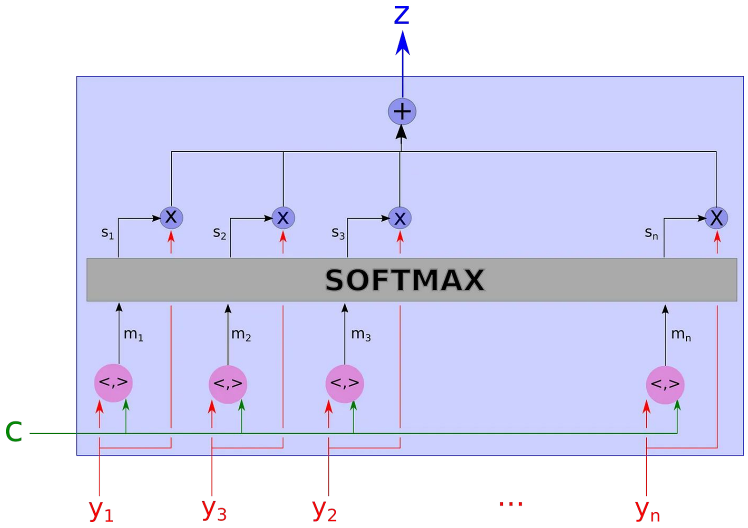
A Girl throws a Frisbee in the park.

upGrad ENTERPRISE

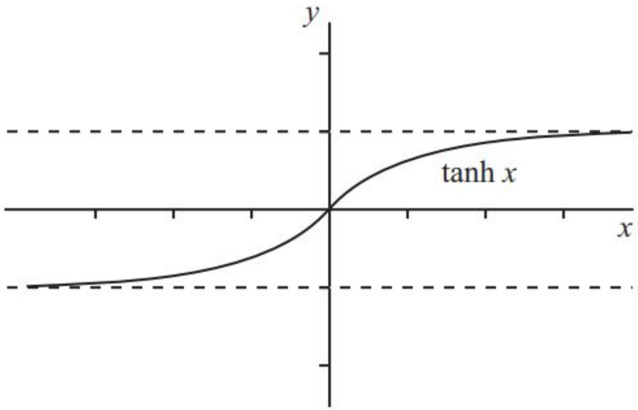
Attention to give title to the image



Attention to give title to the image



$$m_i = \tanh(y_i W_{yi} + C)$$



$$m_i = cy_i$$

Attention to give title to the image

Types of Attention

1. Soft Attention: different parts, different subregions

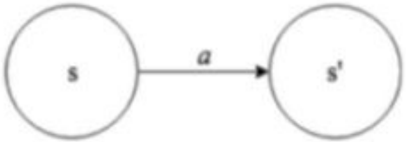
2. Hard Attention: only ONE subregion

Attention

1. Soft Attention: different parts, different subregions

$$z = \sum_n s_n y$$

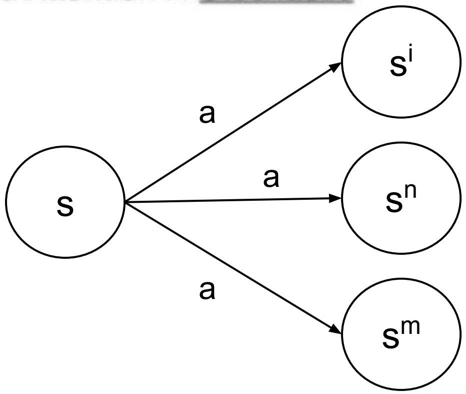
Soft Attention is *Deterministic*



Attention

2. Hard Attention: only ONE subregion

Hard Attention is *Stochastic*



Combining CNN with RNN

Encoding Pictures into words?

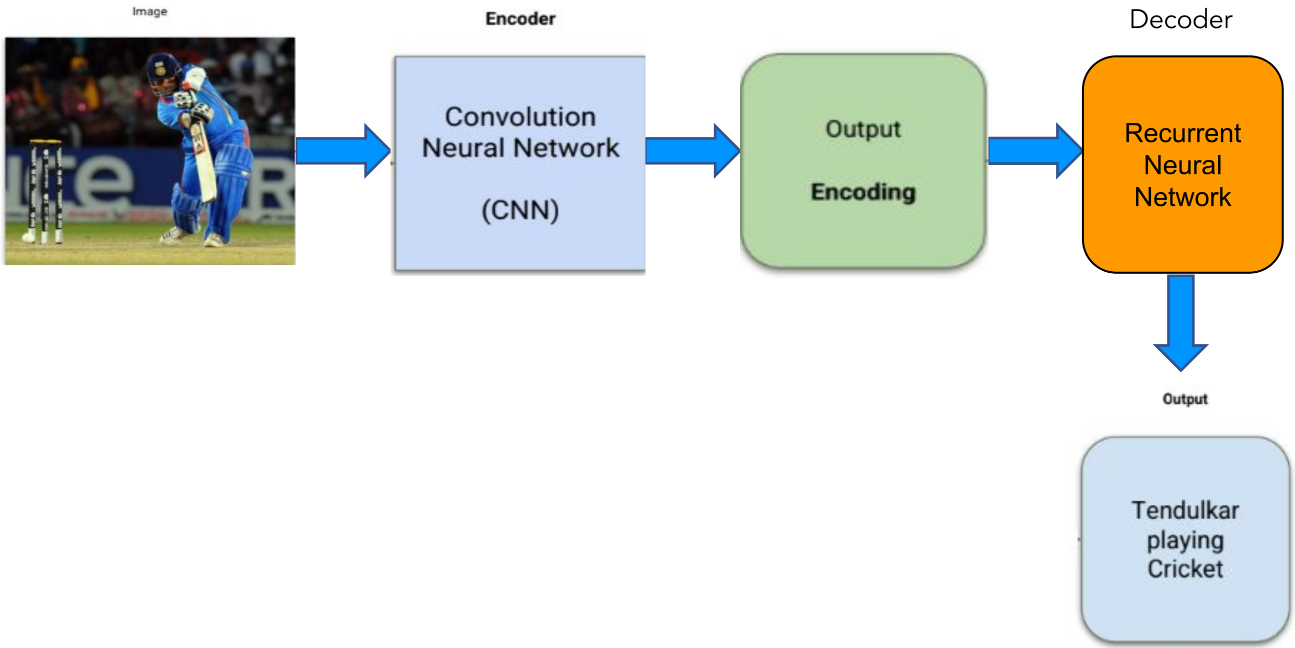
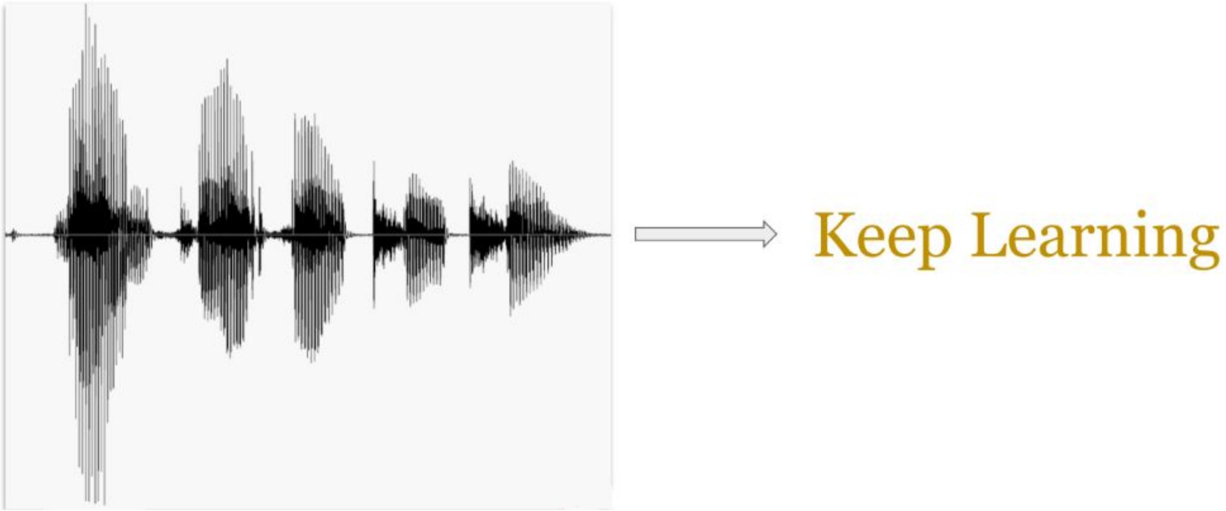


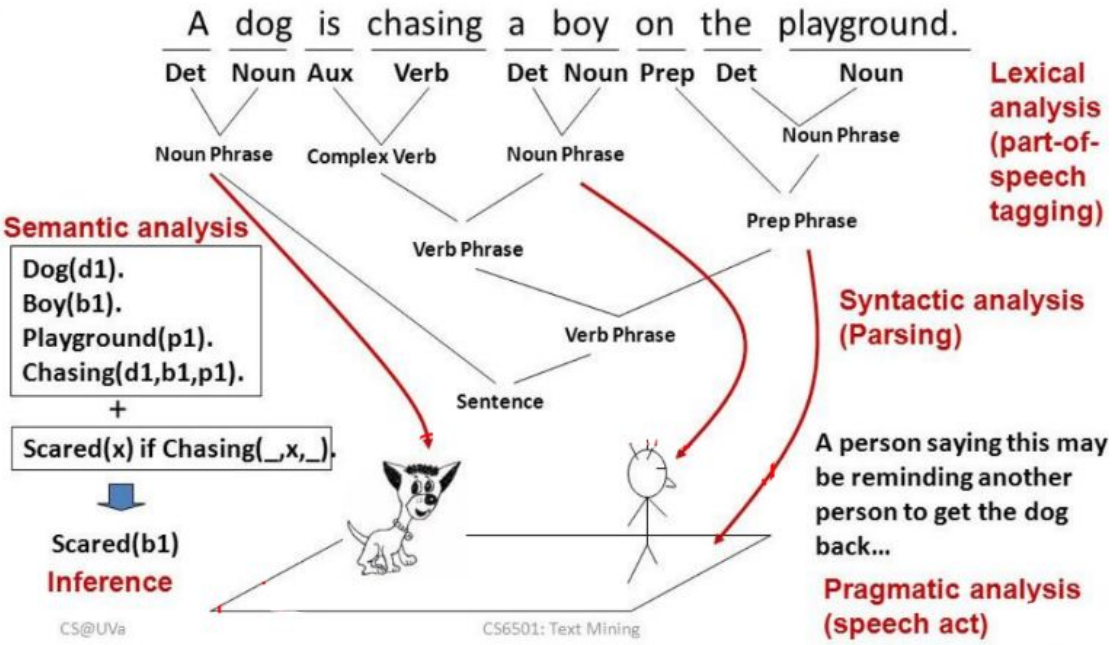
Image search Engine :)

Tendulkar playing cricket

Speech Recognition



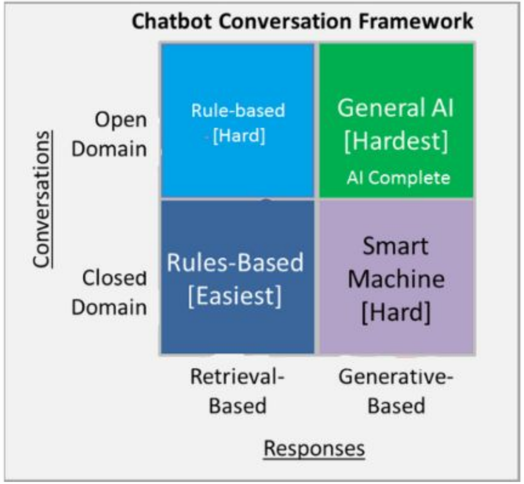
NLP pipeline example



Type of chatbots

Use Cases:

- Uber to book a taxi
- KLM to deliver flight information
- CNN to keep you up-to-date with news content
- Pizza Hut to help you order a pizza



Quick Recap

Encoder Decoder:

Attempt to encode whole
input sequence into
a single output
Only Last vector
Not limited to NLP

Attention Models:

Encoder Decoder with a context
Expose all embeddings
First token wrong is a big
problem

Thank you

References

- http://introtodeeplearning.com/2017/Sequence%20Modeling.pdf
- https://towardsdatascience.com/illustrated-guide-to-lstms-and-gru-s-a-step-by-step-explanation-44e9eb85bf21
- http://jalammar.github.io/illustrated-bert/
- https://arxiv.org/abs/1810.04805
- https://jalammar.github.io/visualizing-neural-machine-translation-mechanics-of-seq2seq-models-with-attention/
- https://jalammar.github.io/illustrated-transformer/
- https://machinelearningmastery.com/teacher-forcing-for-recurrent-neural-networks/
- https://medium.com/analytics-vidhya/a-must-read-nlp-tutorial-on-neural-machine-translation-the-technique-powering-google-translate-c5c8d97d7587
- http://colah.github.io/posts/2015-08-Understanding-LSTMs/
- https://medium.com/datathings/the-magic-of-lstm-neural-networks-6775e8b540cd
- https://pub.towardsai.net/chatgpt-how-does-it-work-internally-e0b3e23601a1