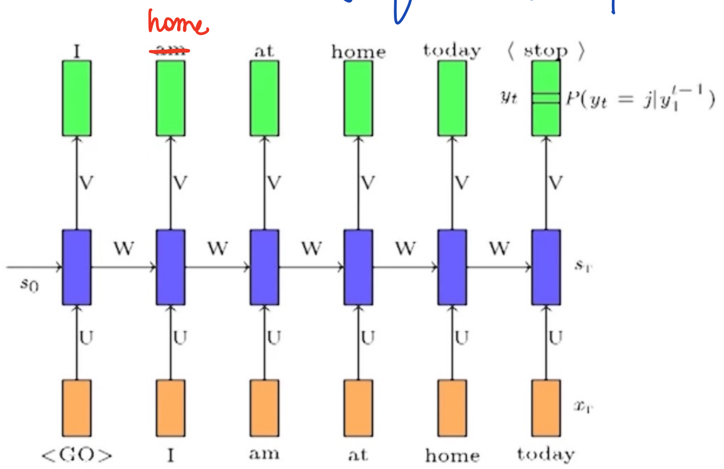
Let's revisit the task of autocompletion.



vocab = {I, at, today, home, am}

j=0	0·1	I
j=1	0·1	at
j=2	0·2	today
j=3	0·1	home
j=4	0·5	am

At any time step t, we want to compute

$$\underset{j \in V}{\arg\max} \; P\left(y_t = j \,\middle|\, y_{t-1}, y_{t-2}, \dots, y_1\right)$$

word

Vocabulary

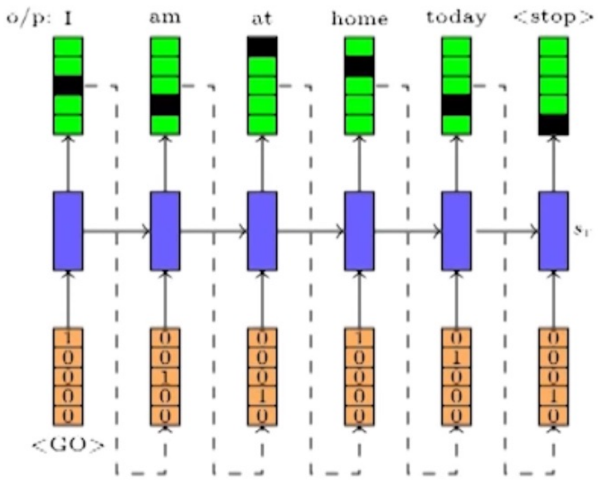
eg. $$P\left(y_4 = \text{home} \,\middle|\, \text{at}, \text{am}, \text{I}\right)$$

Using RNN,

$$P\left(y_t = j \,\middle|\, y_{t-1}, y_{t-2}, \dots, y_1\right) = \left(\text{softmax}\left(Vs_t + c\right)\right)_j$$

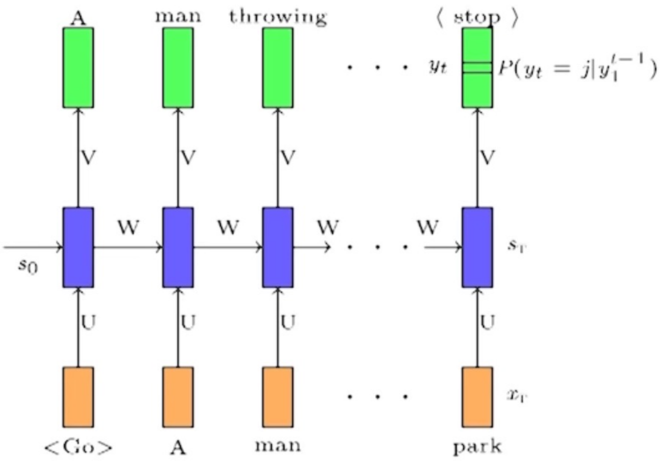
s_t captures all the information until time step t.

At test time,



} one hot encoding of the words.

Image Captioning



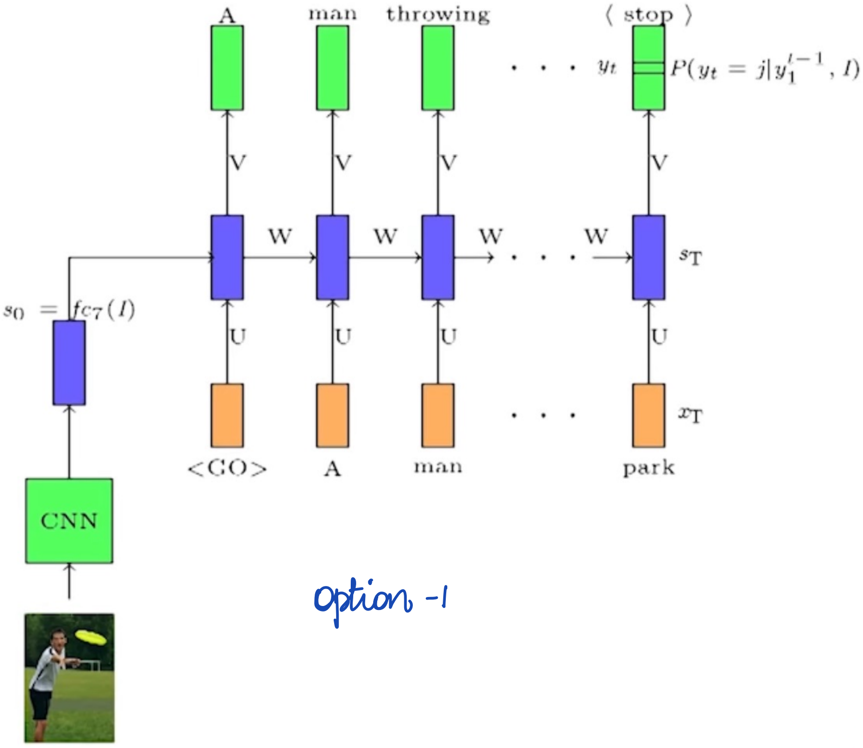
9:05 AM

A man throwing a frisbee in a park

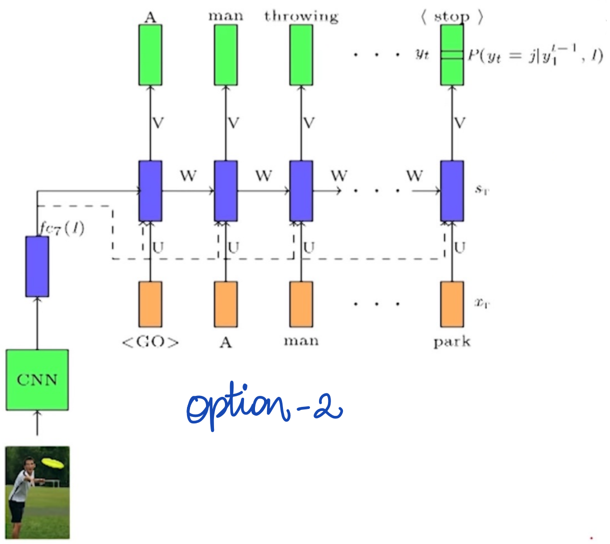
We want to generate a sentence given an image.

We are interested in $P(y_t = j \mid y_{t-1}, \dots, y_1, \overset{\text{Image.}}{I})$

Since CNN architectures are good for images, we use it to learn important features from the image and then pass it on to the RNN model.

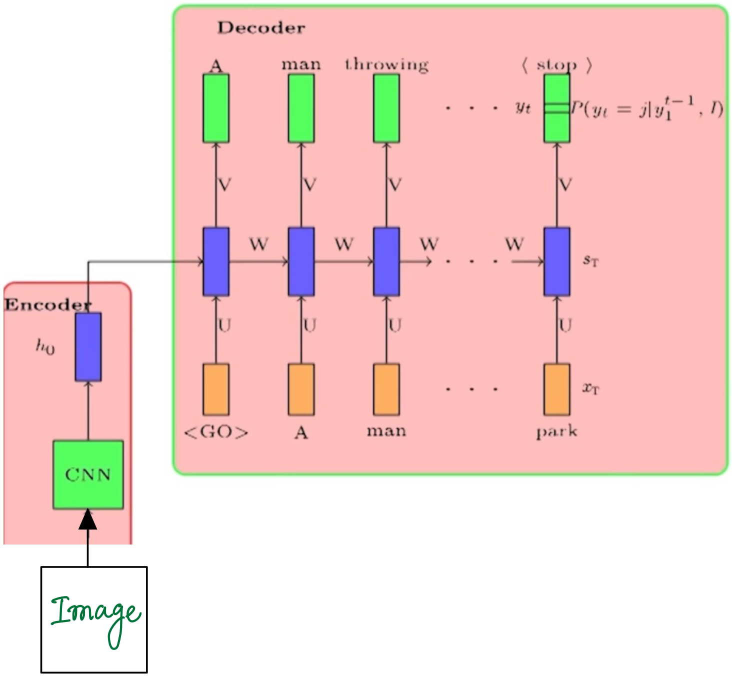


Option -1



option-2

Encoder - Decoder



In image captioning,

A CNN is used to "encode" the image

- Learn good feature representation of the input.

An RNN is used to "decode" a sentence from this encoding.

Task: Image Captioning

Data: $\{ x_i = \text{image} , y_i = \text{caption} \}_{i=1}^{N}$

Model :

Encoder

$$s_0 = CNN(x_i)$$

Decoder

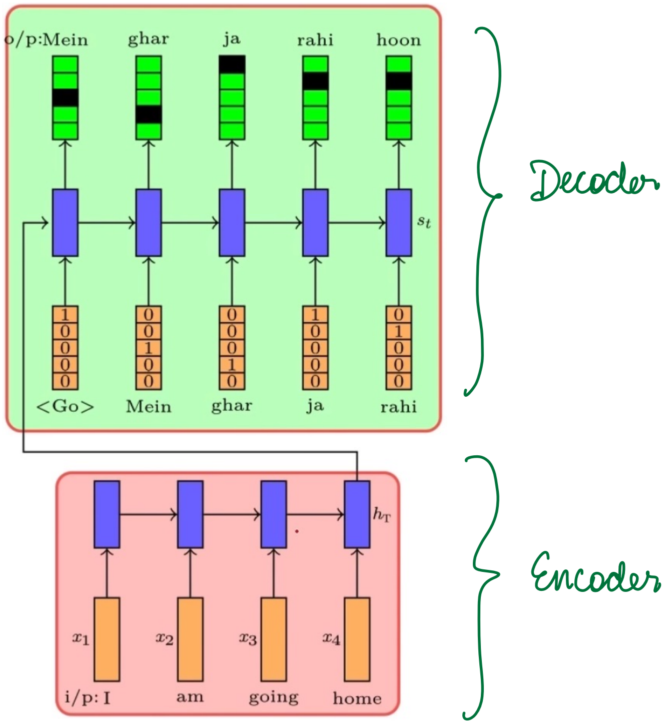
$$s_t = RNN(s_{t-1} , y_{t-1})$$

Parameters : U, V, W, b, c and all weights and biases of CNN.

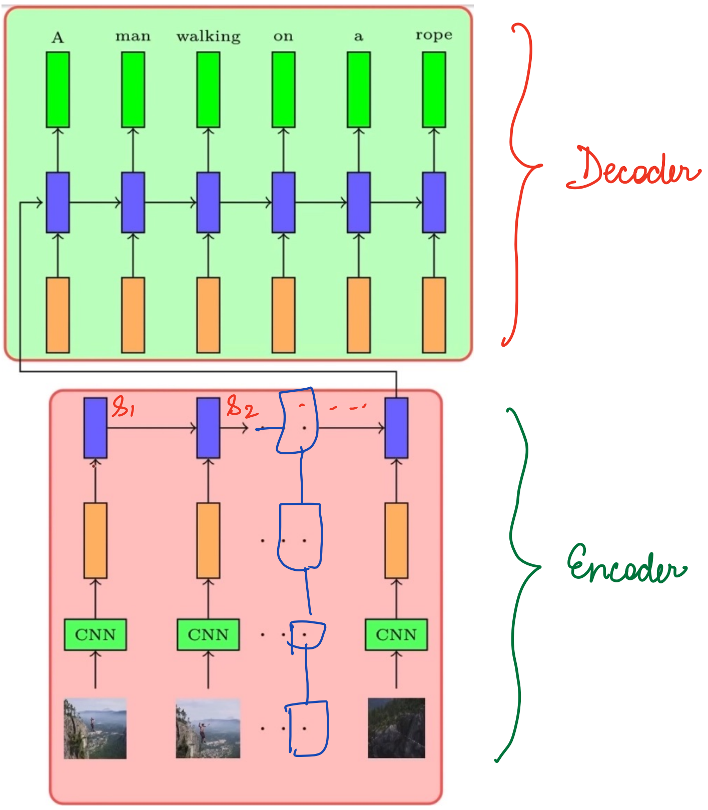
Training all parameters of encoder and decoder together.

– End to end model

Machine Translation



Video Captioning


A
man
walking
on
a
rope
Decoder
s_1
s_2
CNN
CNN
CNN
Encoder