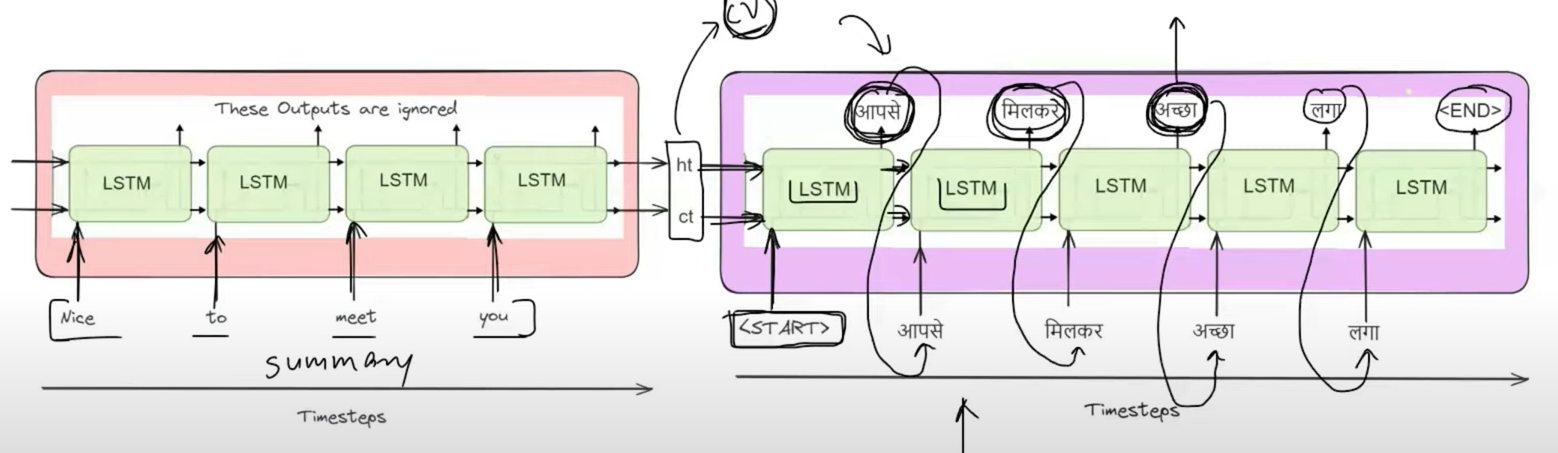
## horizontal line

Masked Multihead and Cross Attention

12.06.2025

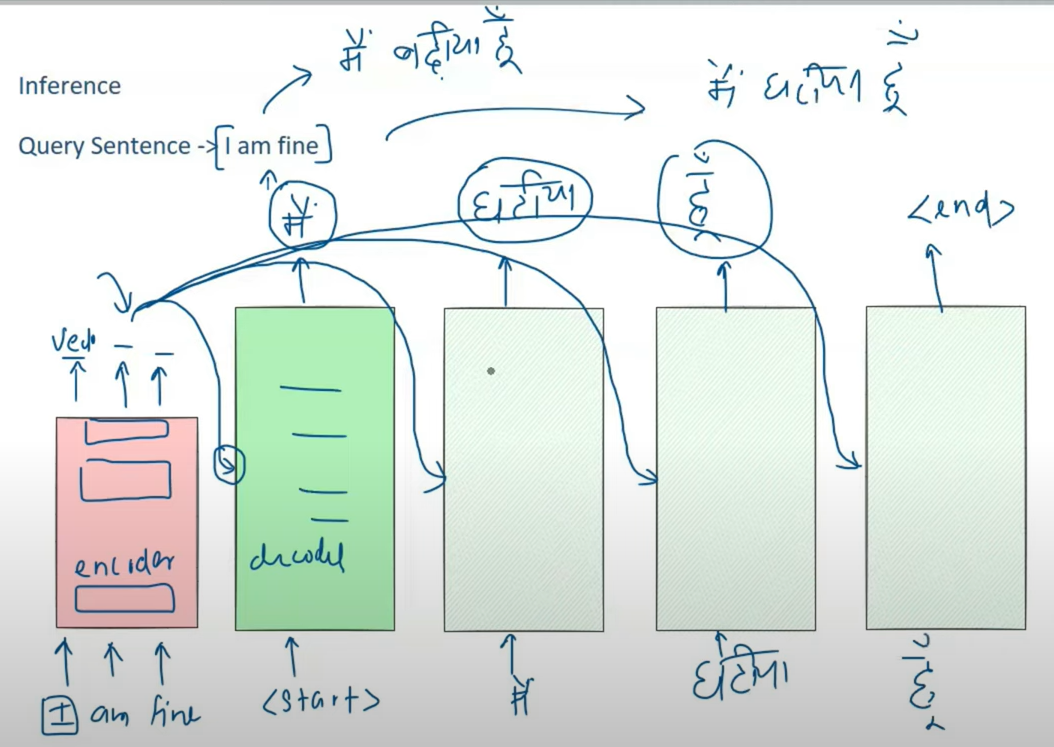
# Masked Multi Head Attention

The Transformer decoder is autoregressive at inference time and non-autoregressive at training. In deep learning , autoregressive models that generate new data points based on previously generated points.



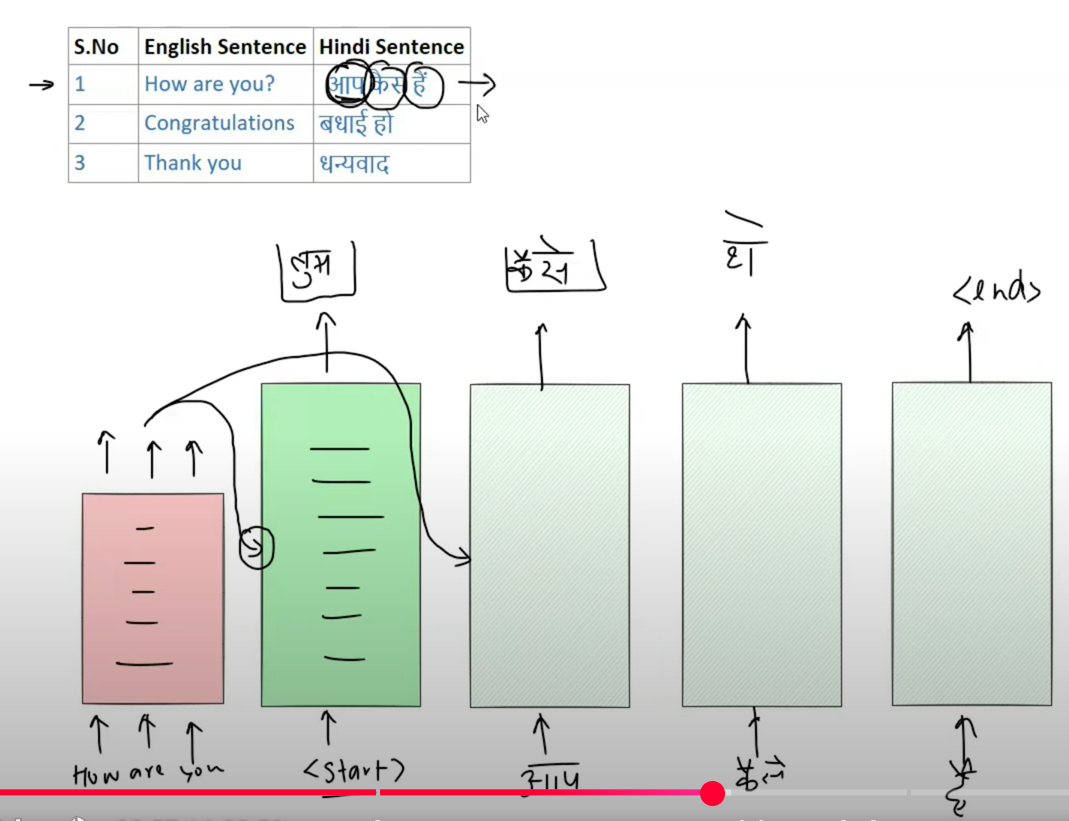
Next time stamp output generated by using previous generated o/p.

## Let’s see at the time of Inference :

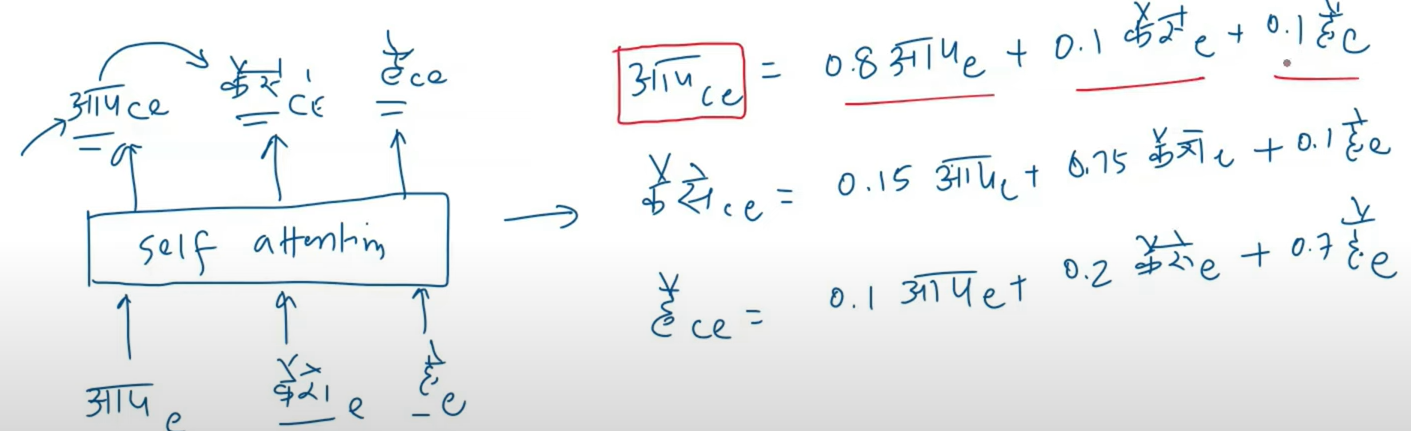
You need to give what is generated in the previous timestamp output either it is correct or in correct.

But at the time of training we always give the correct data that is available in the dataset so that model would be correctly trained.

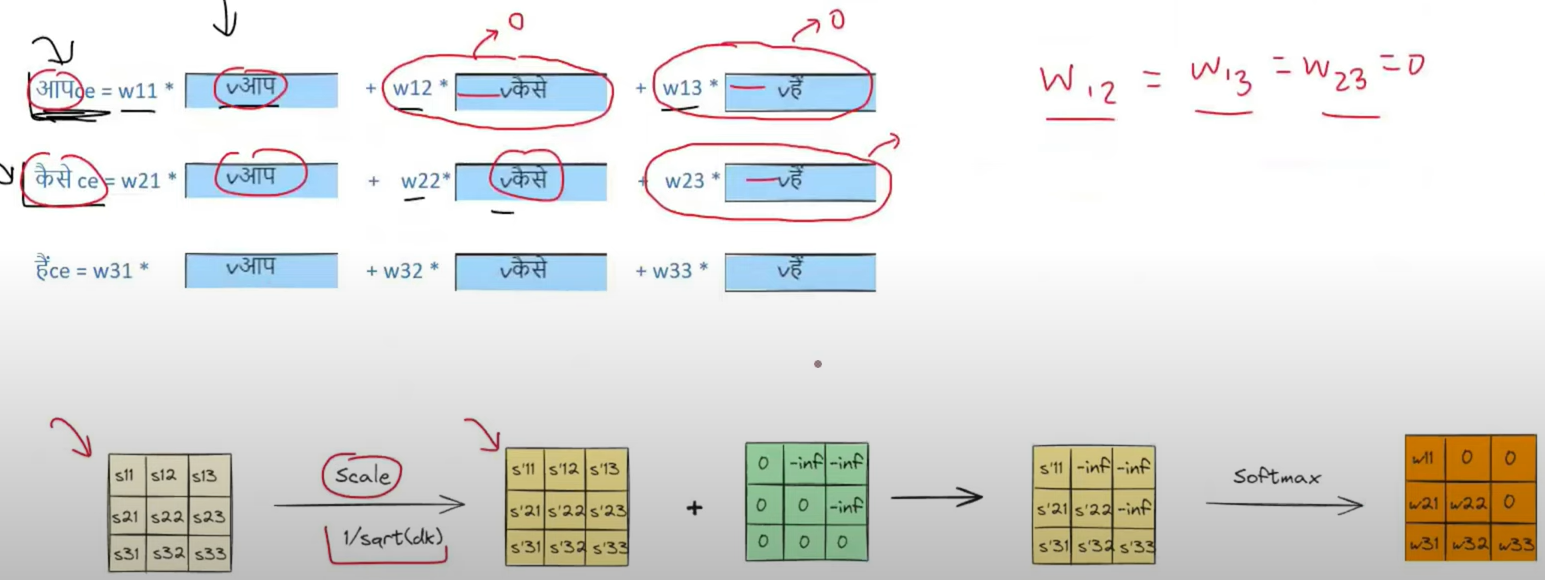
## Now at the time of Training :

This means training is not auto regressive and doesn’t depend on previous outputs.

Data is available so this training could be done parallely but the main problem in that is how we can provide a data that is didn’t received now. This is unfair because that cannot be done at the time of prediction as well.

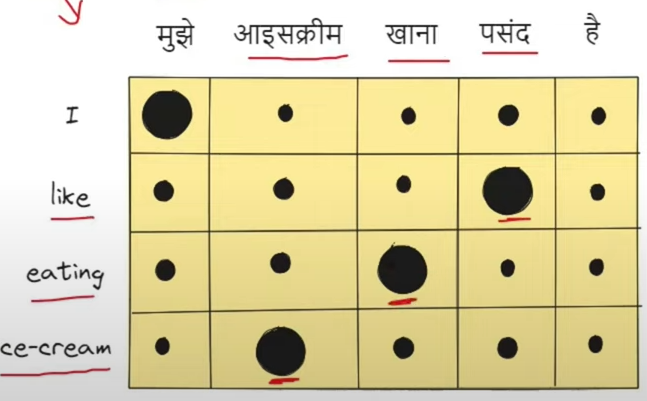
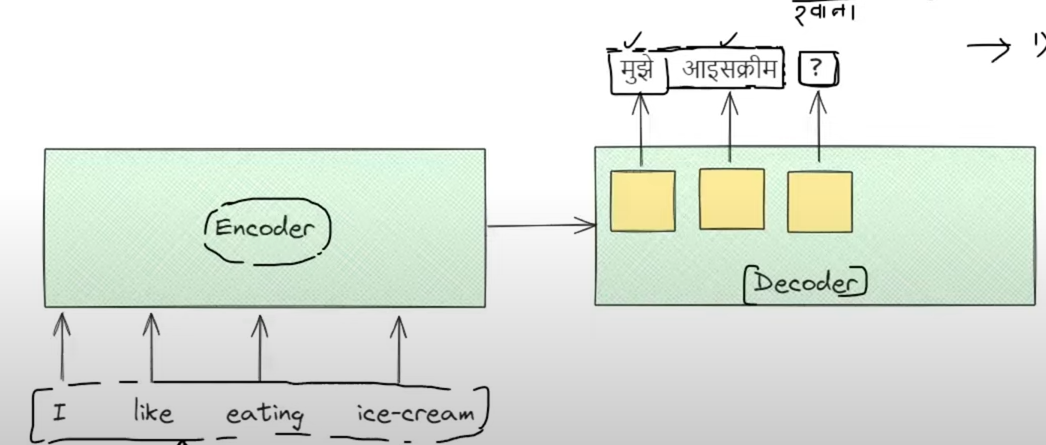
“कैसेe” is also calculated simultaneously with “आपe” that how can it be used when it is not calculated till now? This is Data Leakage.

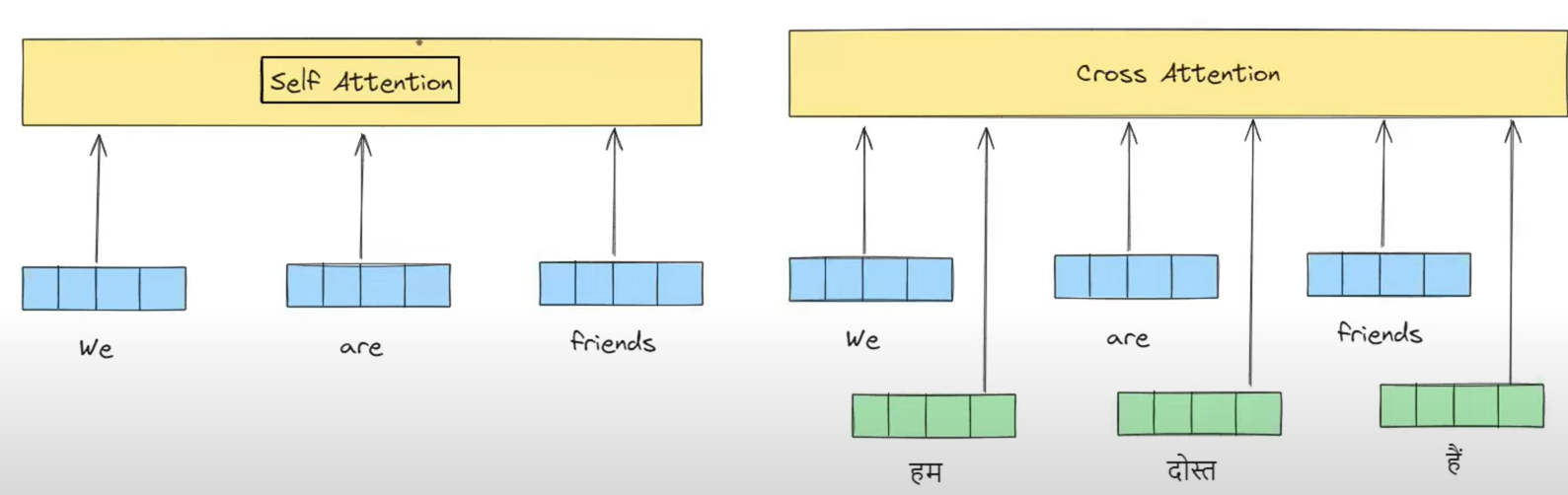
# The Solution :

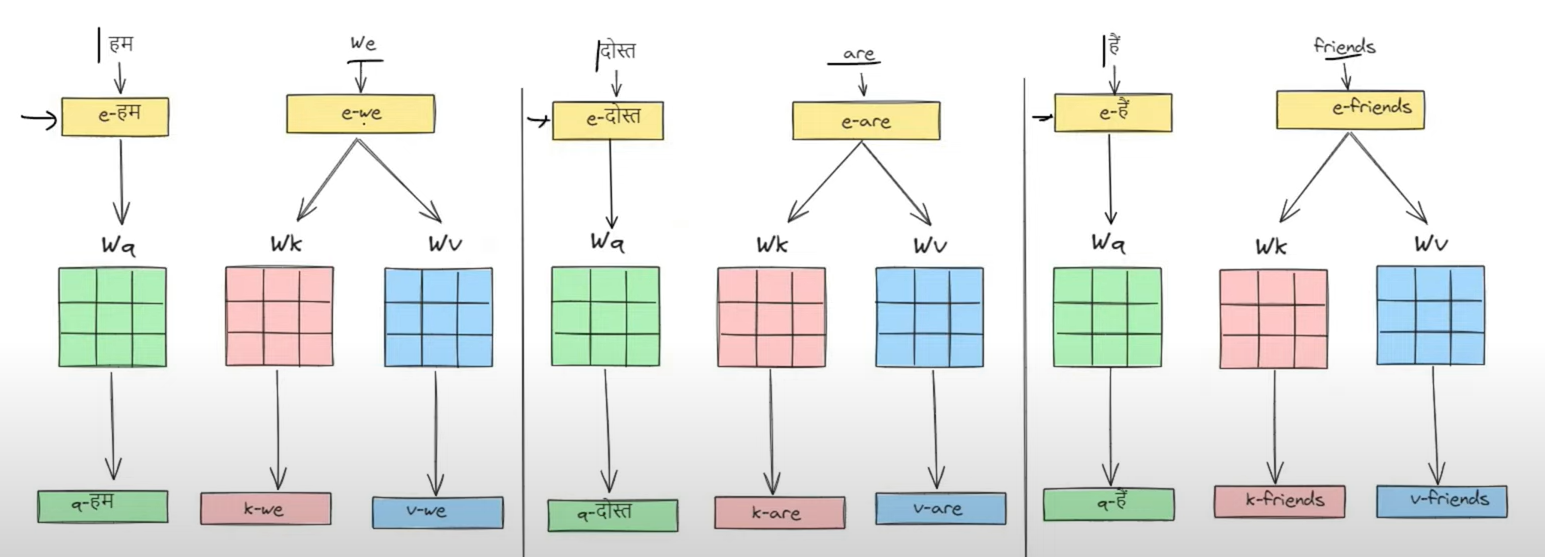
If we neglect the terms where future outputs are used. 

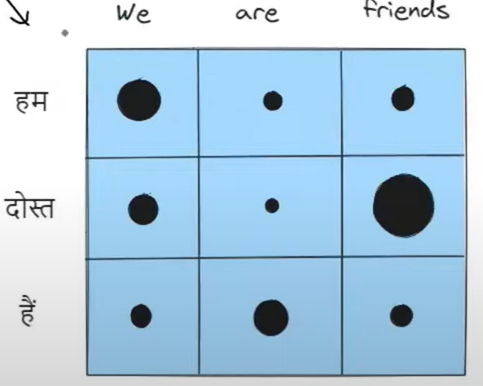
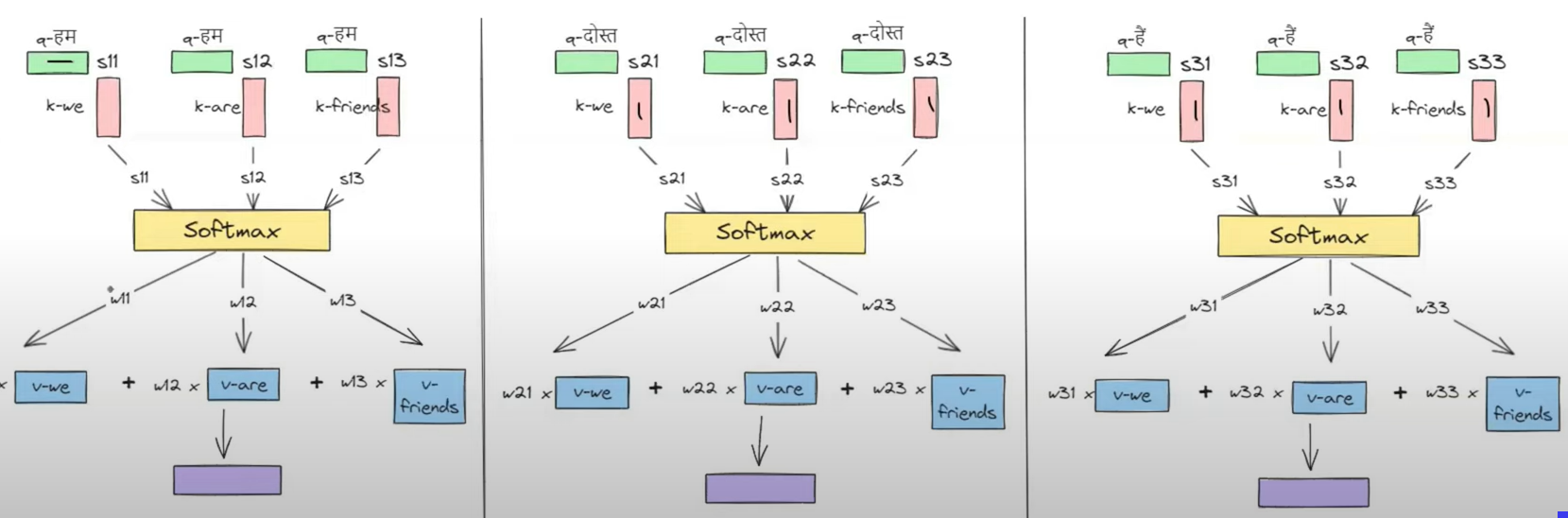
# Cross Attention

A mechanism that focus on different parts of an input sequence while generating an output sequence. We need to find similarity between the 2 sequences to predict the next output.



Input in Cross Attention : 

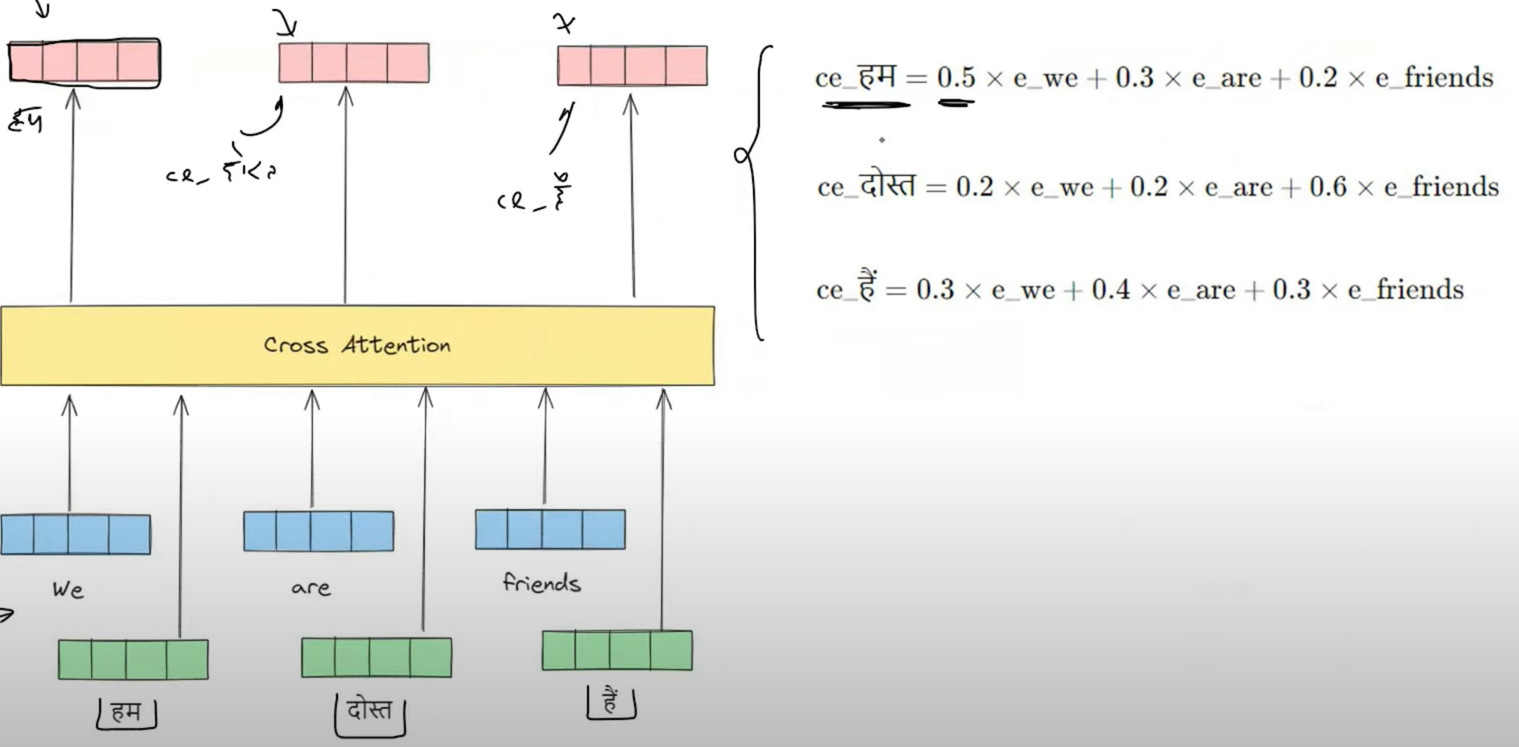
Processing in Cross Attention : 



We take Query vector from output sequence while

Key and Value vector from input sequence.

Output in Cross Attention :

The no. of contextual embeddings we get in output is equal to the no. of words in output sequence.

Cross Attention is very similar to Bahdanau / Luang Attention.

Use Cases is when you need to find similarity between 2 different sequences.

E.g. Image to text , text to image , speech to text.