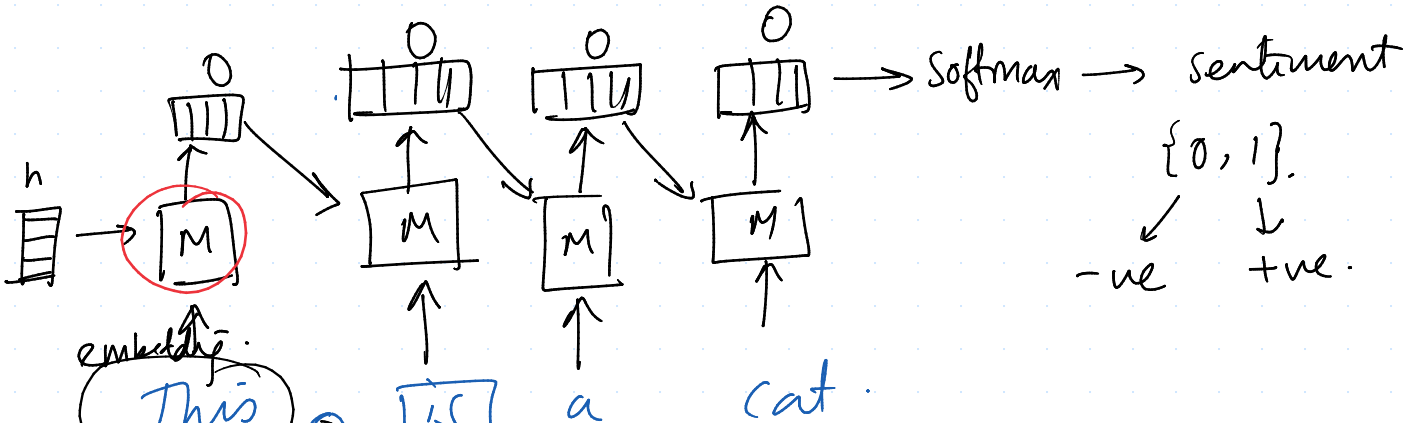
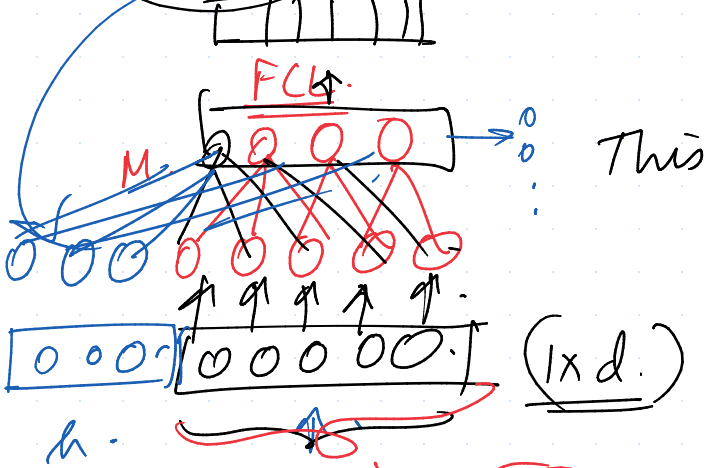
Lecture-11 Transformers Continued.





tokenizer embedding

This → 3 →

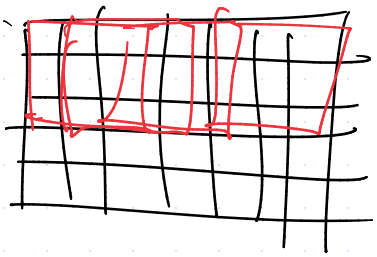
n x d → vector.

n [ ] embedding.

d.

Transformer ×.

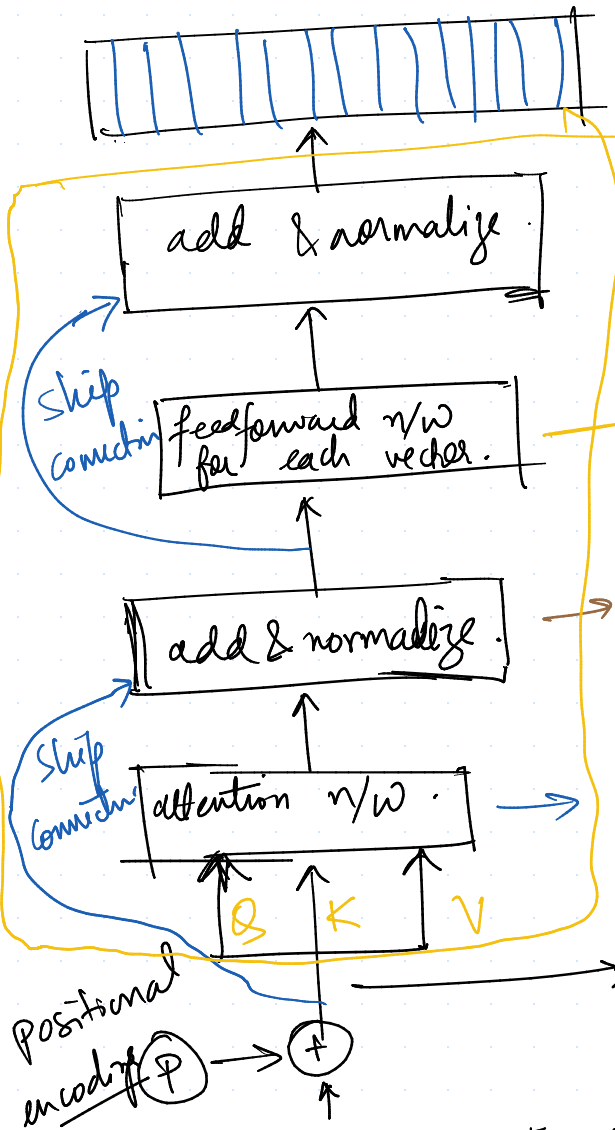
2D.



Maxpool ; BN ; - - -

(2D)

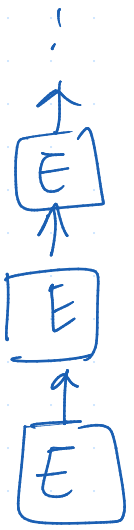
Attention - Based Sequence Encoder.



o/p sequence.

Normalize.

Encoder (E)



deep sequence encoder.

→ to not loose the original word embeddings.

→ context of the surrounding words are taken into account (context)

→ now they know about the relative positions of the words.

(vectors in $\mathbb{R}^D$).

→ meaning of the words.

↳ doesn't take into account the contextual information of the words.

i/p sequence.

(words in English).

Feedforward Network.

Provide regularization or structure to the n/w.

Restrict the o/p of the n/w to be contained to the subspace of the vectors with the n/w. (maths concept)

tanh → If we use → o/p (-1 & 1)

$$v = \frac{1}{N} \sum_{n=1}^{N} \tilde{c}_n$$

$$p(\text{+ve sentiment}) = \frac{\exp(v \cdot e)}{1 + \exp(v \cdot e)}$$

(0 - 1)

Sigmoid $= \frac{1}{1 + \exp^{-(v \cdot e)}}$

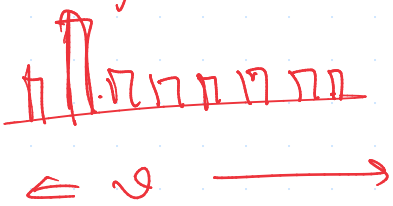
Add some activation like sigmoid / softmax for classification after the sequence encoder.

Predict the next word.

$$p(w_{N+1}=k) = \frac{\exp(v \cdot e_k)}{\sum_{j=1}^{|v|} \exp(v \cdot e_j)}$$

softmax.

argmax.



Predict the sequence of the next word.

English: The | orange | cat | is | Nacho

↓

Encode.

Translation task.

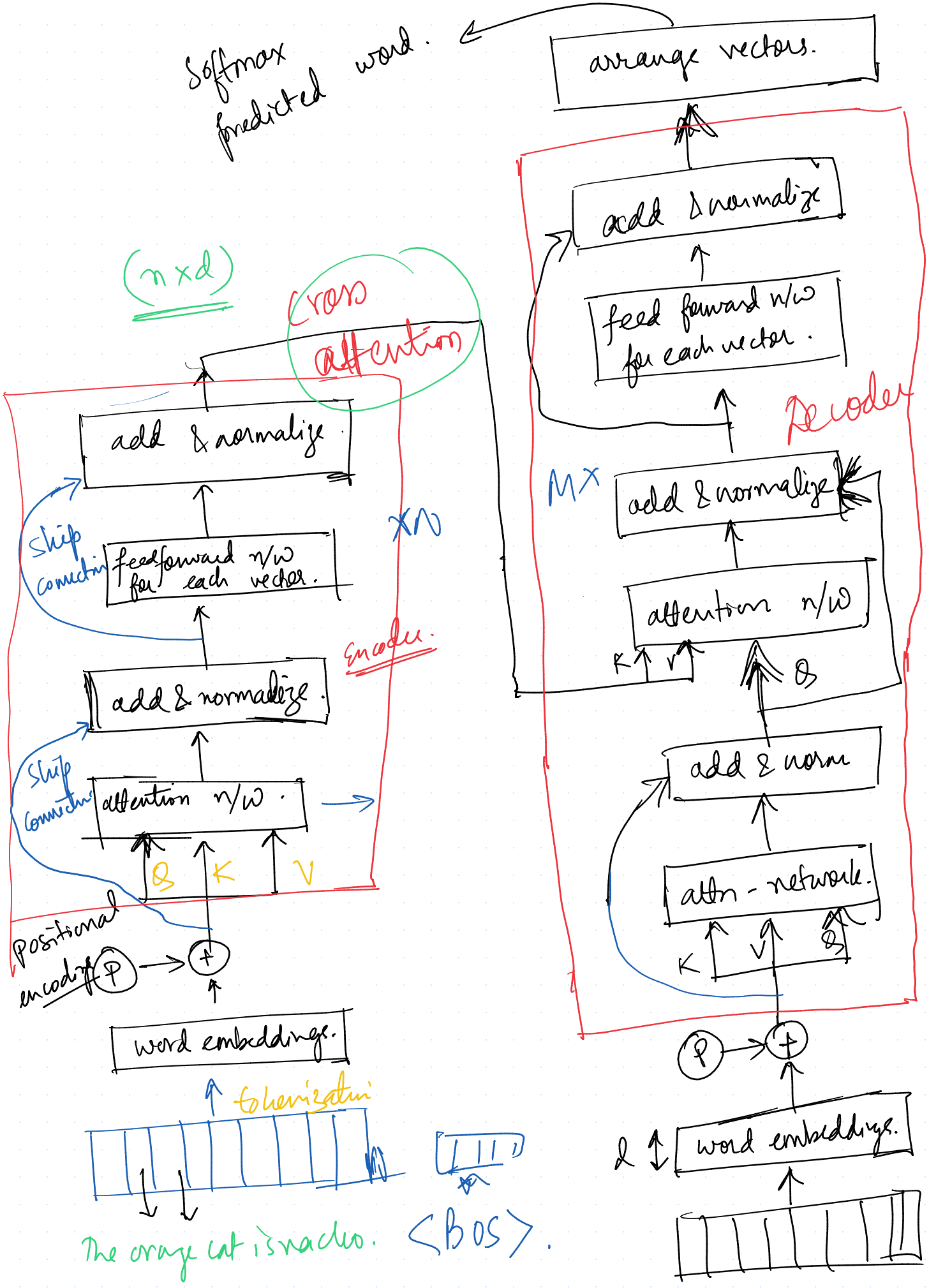
↓

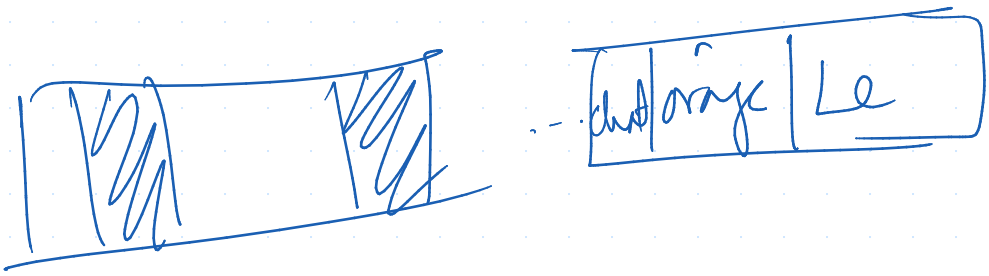
French: Le | chat | orange | est | Nacho.

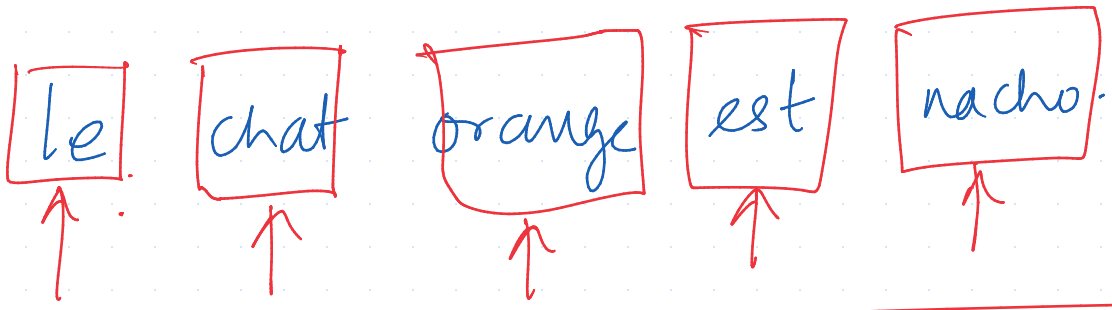
or use

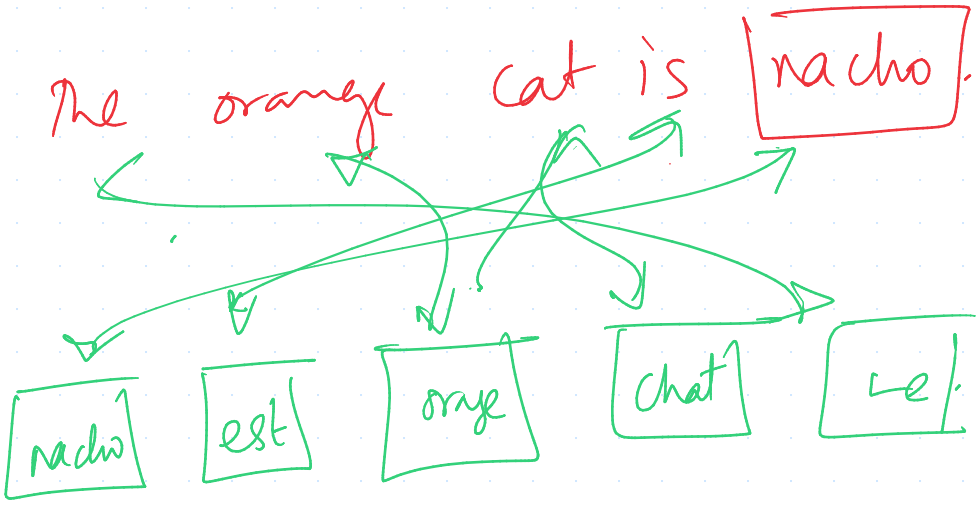
source text: $w_1, w_2, \ldots, w_n$.

Summarize → Transformers.

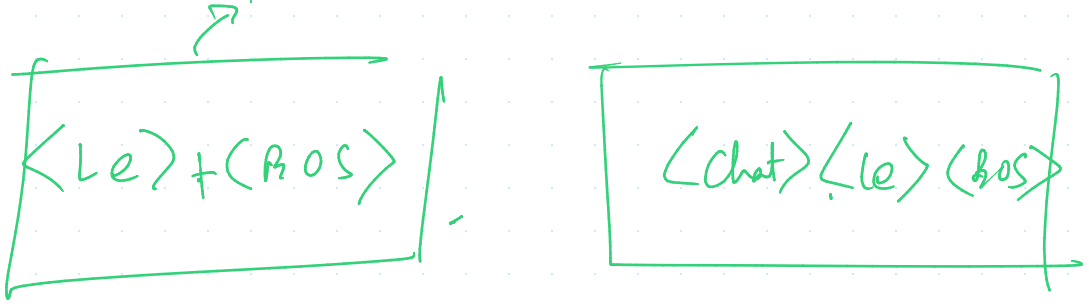

Softmax predicted word.
arrange vectors.
add & normalize
feed forward n/w for each vector.
Decoder
M×
add & normalize
attention n/w
K
V
Q
add & norm
attn - network
K
V
Q
word embeddings
word embeddings
Cross attention
(n × d)
add & normalize
skip connection
feedforward n/w for each vector.
×N
Encoder
add & normalize
skip connection
attention n/w
Q
K
V
positional encoding
word embeddings
tokenization
The orange cat is reading.
<BOS>.



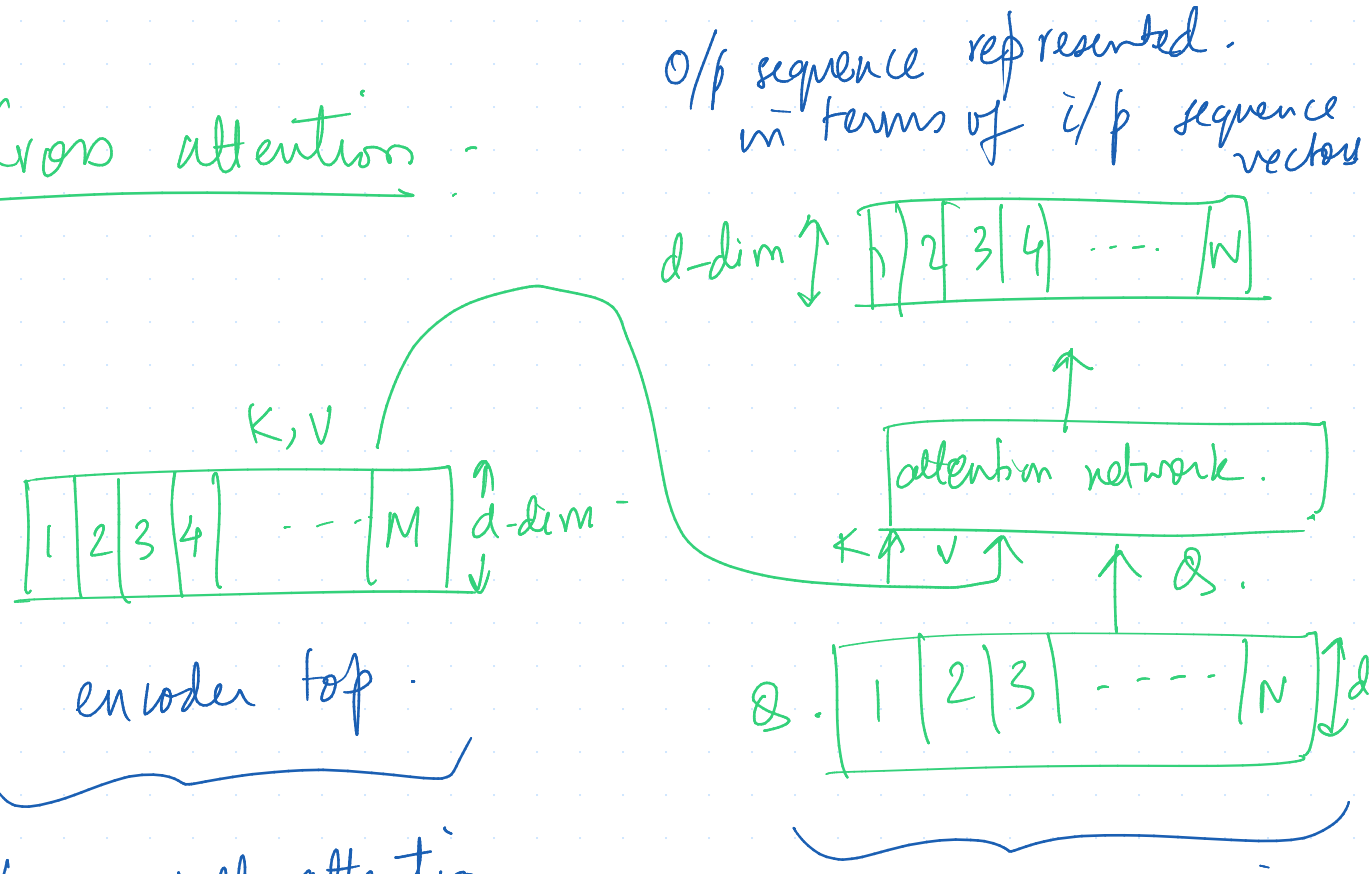




i/p to the bottom right is shifting as we predict new words.



Cross attention :-



o/p of self attention n/w applied to i/p sequence.

o/p to the decoder

o/p to self-attention network applied to sequence decoder thus far

N-words have been decoded thus far → Query (Q)

Cross attention :-

K, V → coming from the i/p N-words that are encoded at the top of the encoding n/w.

Encoding the i/p sequence → left → Encoder.

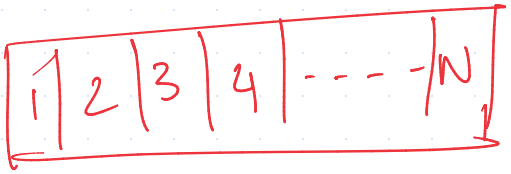
decoding the o/p sequence → right → decoder.

- self attention → b/w the N-words.
- Cross attention → b/w the o/p of the encoding & the o/p of the decoding.

Multihead attention.

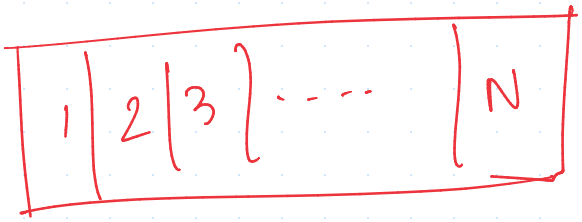
Output (o/p) = Attention (K, V, Q).

Q



d-dim.

V

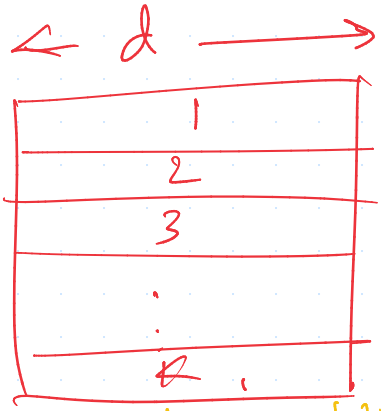


d-dim.

K

d-dim.

Project onto k-vectors.

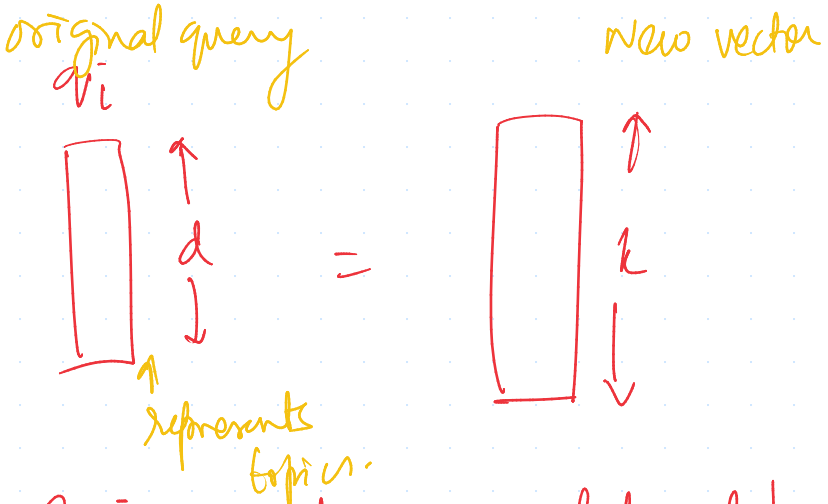


represents meta-topics.

M_i^Q.

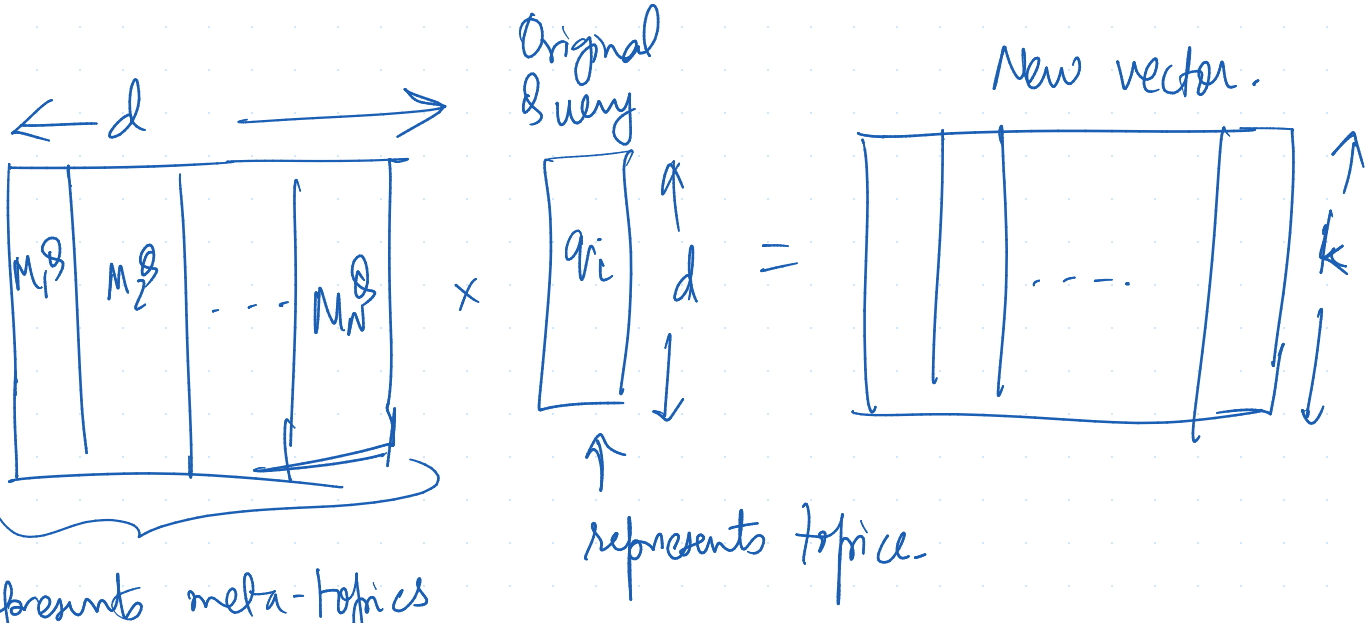
projection matrix M^Q associated with query vectors.

Query Matrix.

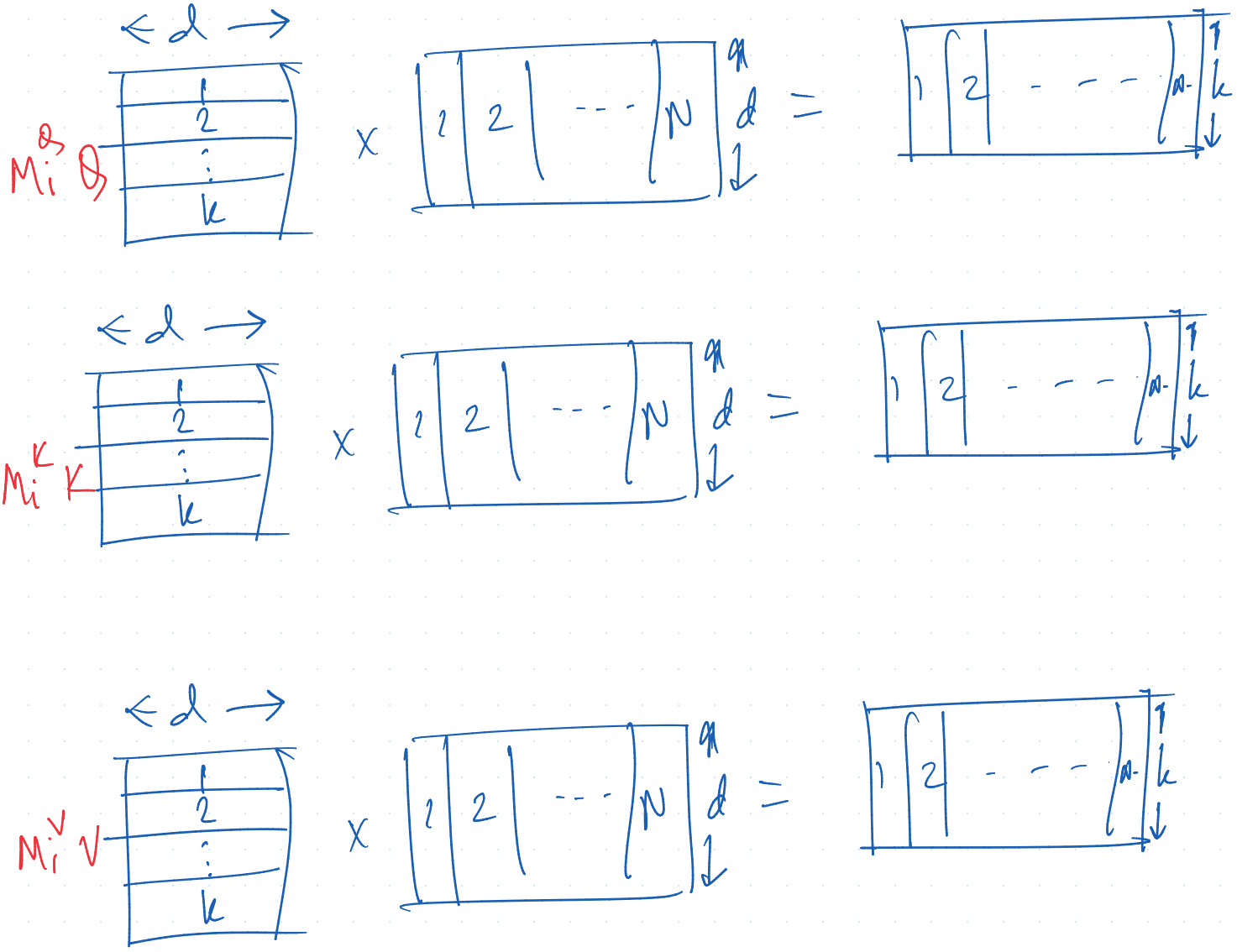


linear subspace defined by M^Q.

New projected vector denoted $M^Q q_i$ applied to all queries as $M^Q Q$.



Represents meta-topics M_i^Q.



Attention head i

$$\text{output}_i = \text{Attention}(M_i^K K, M_i^V V, M_i^Q Q)$$

$$\text{Multi-Head op} = \text{Concat}(\text{o/p}_1, \text{o/p}_2, \cdots, \text{o/p}_k)\, W^O$$

Multi-head attention :-

- Allows attention to focus on different aspects of the language/words.
- h-projections allow the model to attend based on different (projections) of word vectors.