Decoder-based models

Decoder-based models are trained to generate coherent and contextually relevant text by autoregressively predicting the next word or token based on the preceding ones.

Some of the most common decoder-based models are GPT-2 and GPT-3 proposed by OpenAI, which are adept at understanding the context provided and generating output that corresponds to the input they receive.

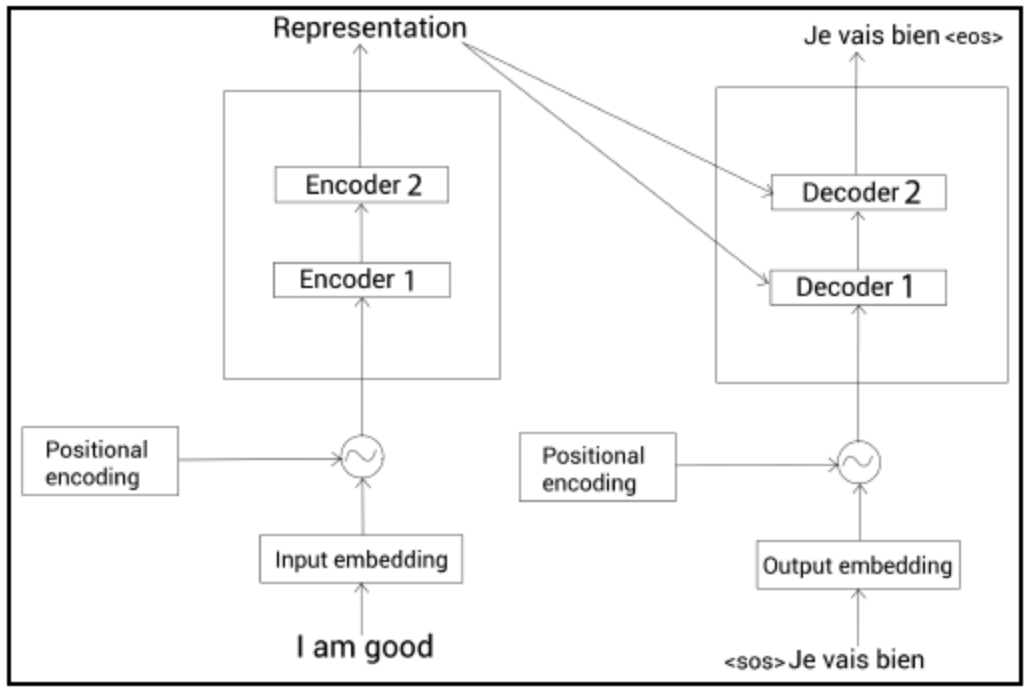


Figure 1.44 – Encoder and decoder of the transformer

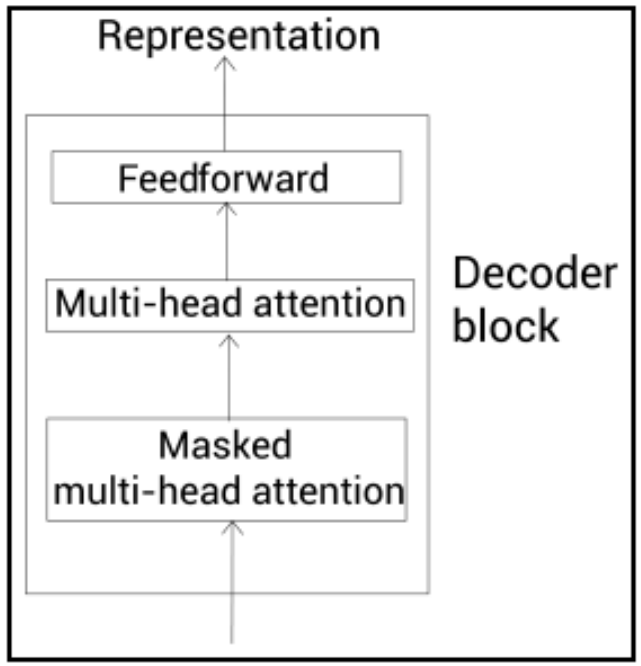


Figure 1.42 – A decoder block

X =					
	<sos>	7.9	3.5 ...	16.1	x_1
	Je	8.1	4.4 ...	83.1	x_2
	vais	17	0.54 ...	6.12	x_3
	bien	11.12	11.12...	22.1	x_4

$$\frac{Q_i K_i^T}{\sqrt{d_k}} =$$

	<sos>	Je	vais	bien
<sos>	9.125	7.5	1.25	5.625
Je	5.0	12.37	3.12	8.75
vais	7.25	5.0	10.37	1.25
bien	1.5	1.37	1.87	10.0

$$\frac{Q_i K_i^T}{\sqrt{d_k}} =$$

	<sos>	Je	vais	bien
<sos>	9.125	$-\infty$	$-\infty$	$-\infty$
Je	5.0	12.37	3.12	8.75
vais	7.25	5.0	10.37	1.25
bien	1.5	1.37	1.87	10.0

To predict the word next to the word <*sos*>, our model should not attend all the words to the right of <*sos*>

So, mask all the words to the right of <*sos*>

$$\frac{Q_i K_i^T}{\sqrt{d_k}} =$$

	<sos>	Je	vais	bien
<sos>	9.125	$-\infty$	$-\infty$	$-\infty$
Je	5.0	12.37	$-\infty$	$-\infty$
vais	7.25	5.0	10.37	1.25
bien	1.5	1.37	1.87	10.0

Figure 1.50 – Masking all the words to the right of Je with $-\infty$

$$\frac{Q_i K_i^T}{\sqrt{d_k}} =$$

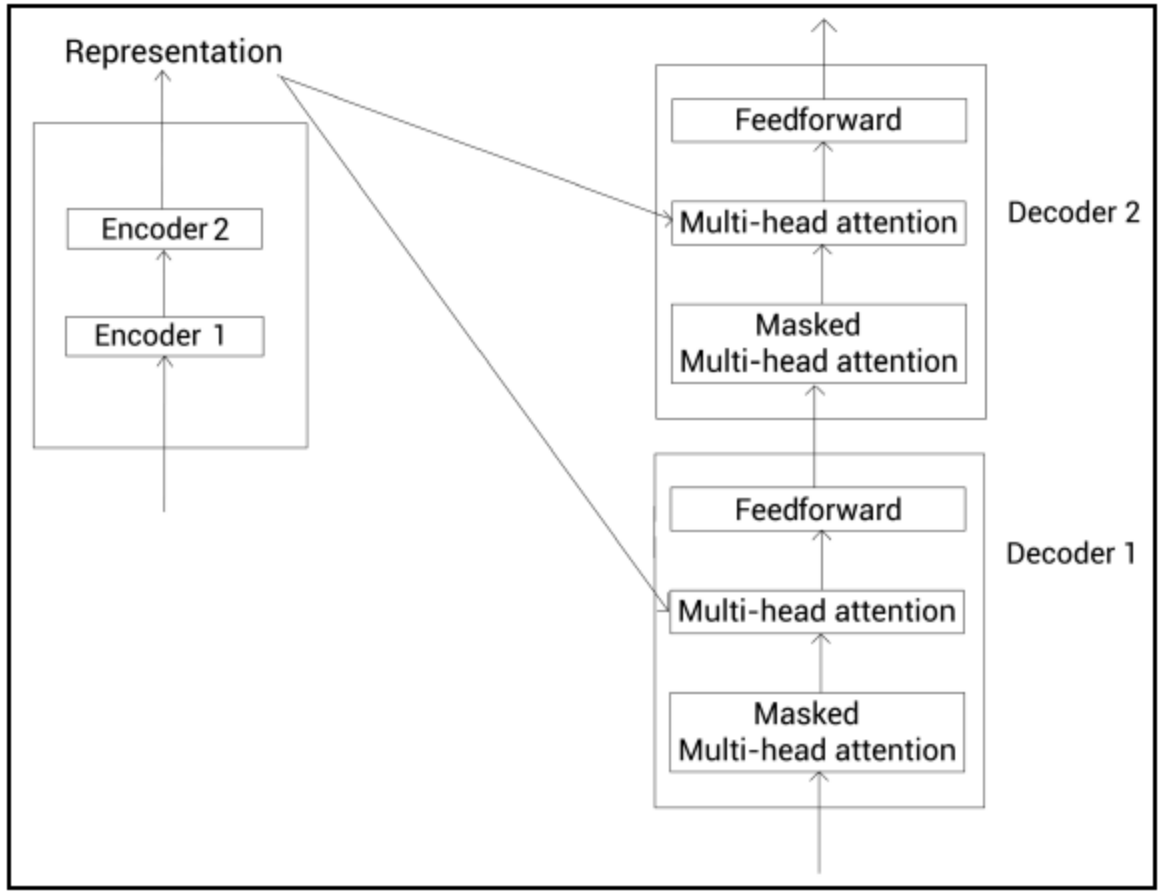
	<sos>	Je	vais	bien
<sos>	9.125	$-\infty$	$-\infty$	$-\infty$
Je	5.0	12.37	$-\infty$	$-\infty$
vais	7.25	5.0	10.37	$-\infty$
bien	1.5	1.37	1.87	10.0

Figure 1.51 – Masking all the words in the right of vais with $-\infty$

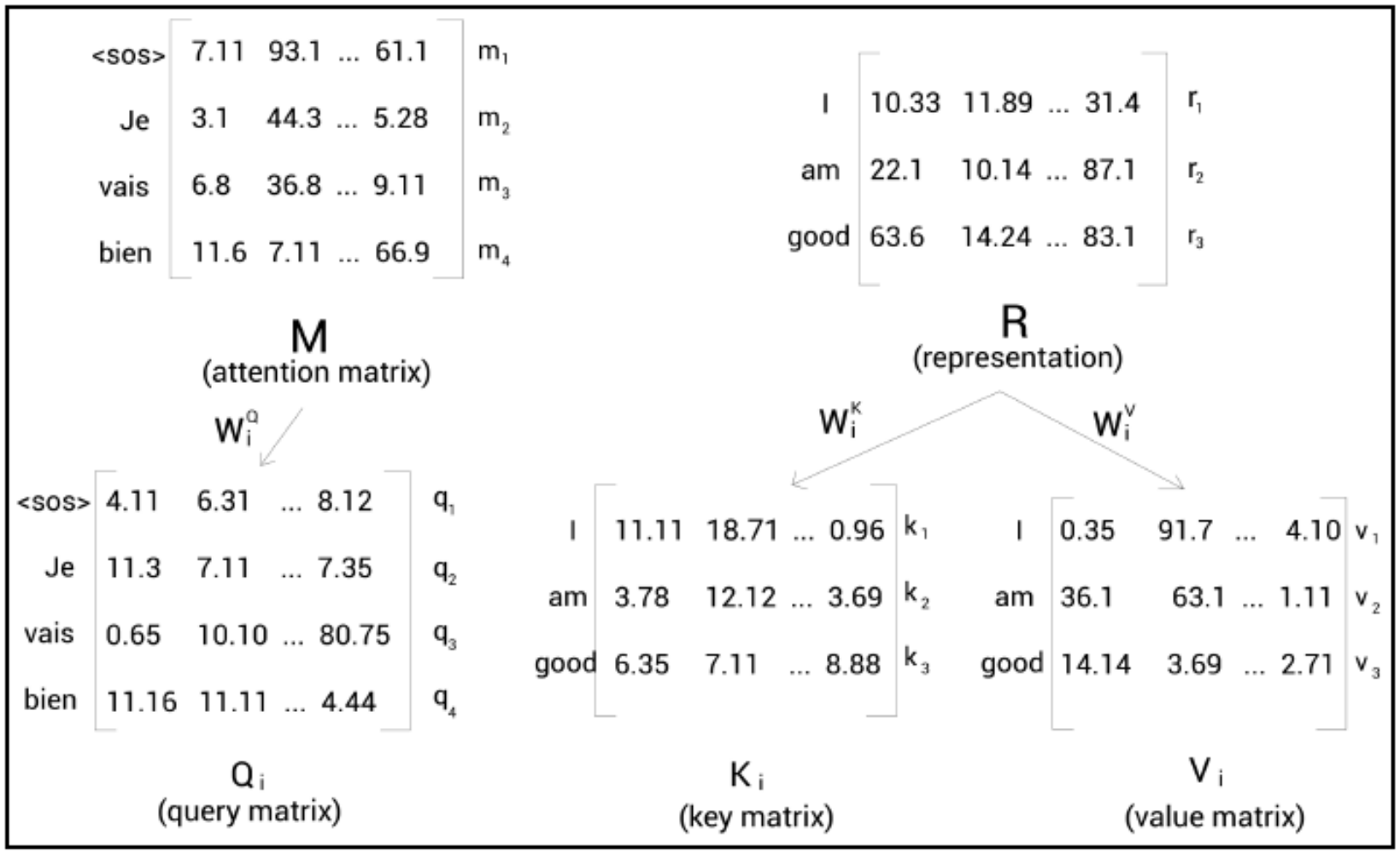
- Now, we can apply the softmax function to the preceding matrix and multiply the result by the value matrix, , and obtain the final attention matrix, .
- Similarly, we can compute *h* number of attention matrices, concatenate them, and multiply the result by a new weight matrix, , and create the final attention matrix, , as shown:

$$M = \text{Concatenate}(Z_1, Z_2, \ldots Z_i, \ldots Z_h)W_0$$

Multi-head attention(encoder-decoder attention)



- The query matrix, Q_i, is created by multiplying the attention matrix, M, by the weight matrix, W_i^Q.
- The key and value matrices are created by multiplying the encoder representation, R, by the weight matrices, W_i^K and W_i^V, respectively. This is shown in the following figure:



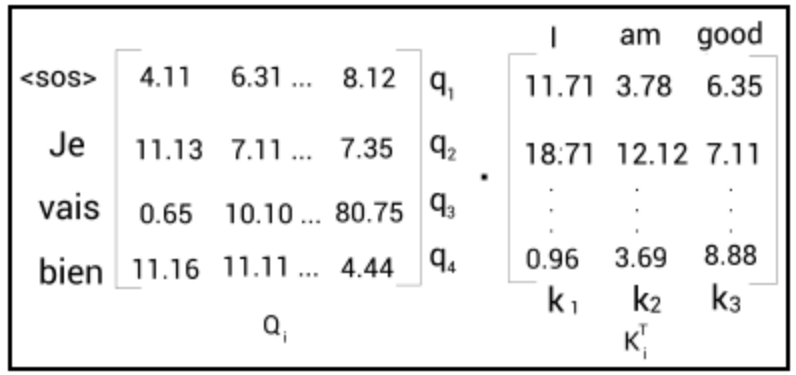
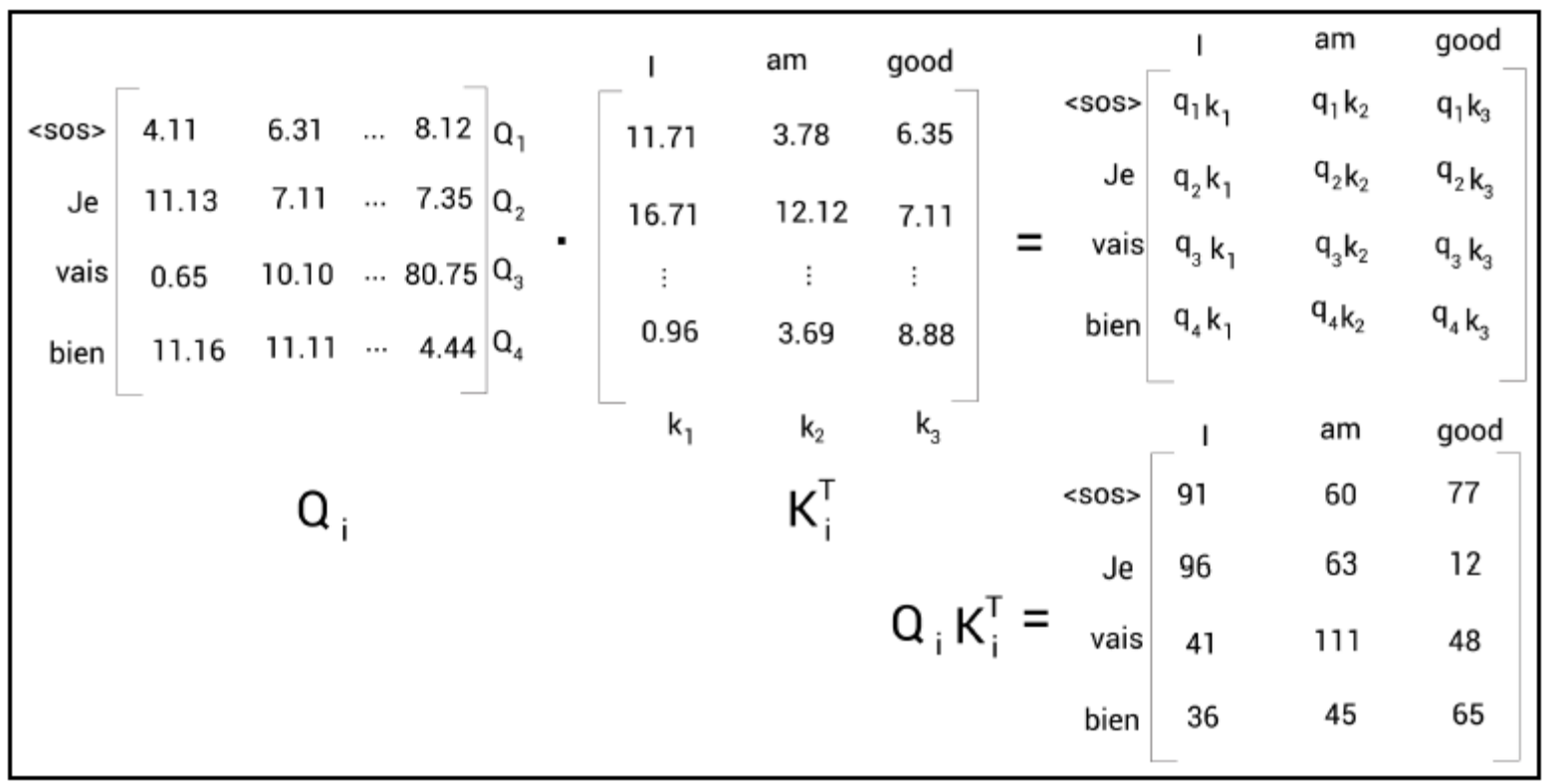
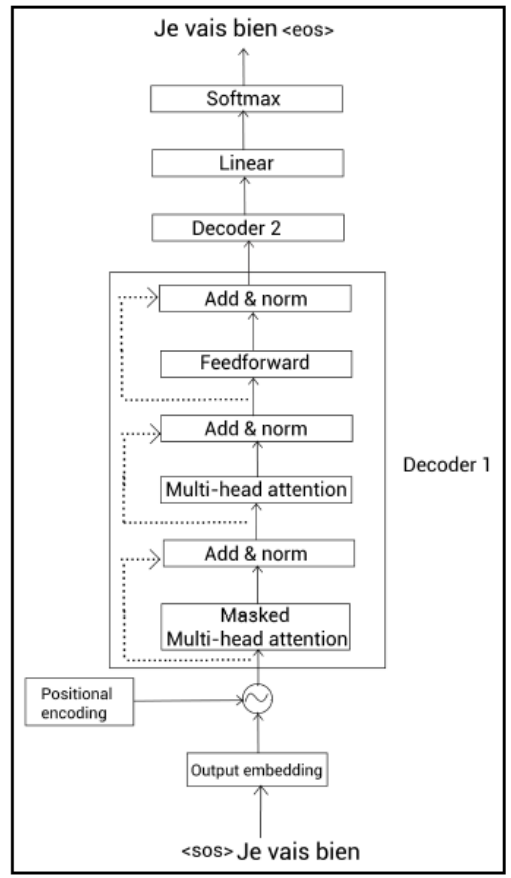


Figure 1.54 – Query and key matrices




Je vais bien <eos>
Softmax
Linear
Decoder 2
Add & norm
Feedforward
Add & norm
Multi-head attention
Add & norm
Masked
Multi-head attention
Decoder 1
Positional
encoding
Output embedding
<sos> Je vais bien

Masked multi-head attention

- instead of feeding the input directly to the decoder, we convert it into an
- embedding (output embedding matrix) and add positional encoding, and then feed it to the
- decoder.