

NLP

word2vec

RNN

Transformers

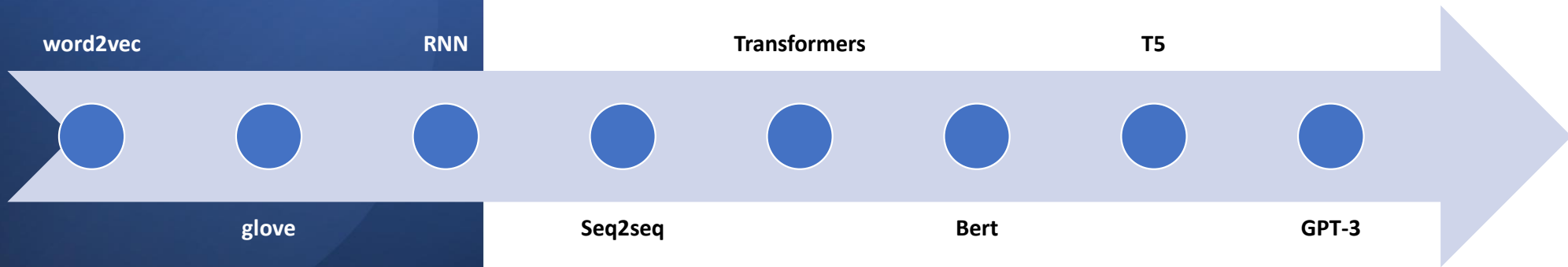
T5

glove

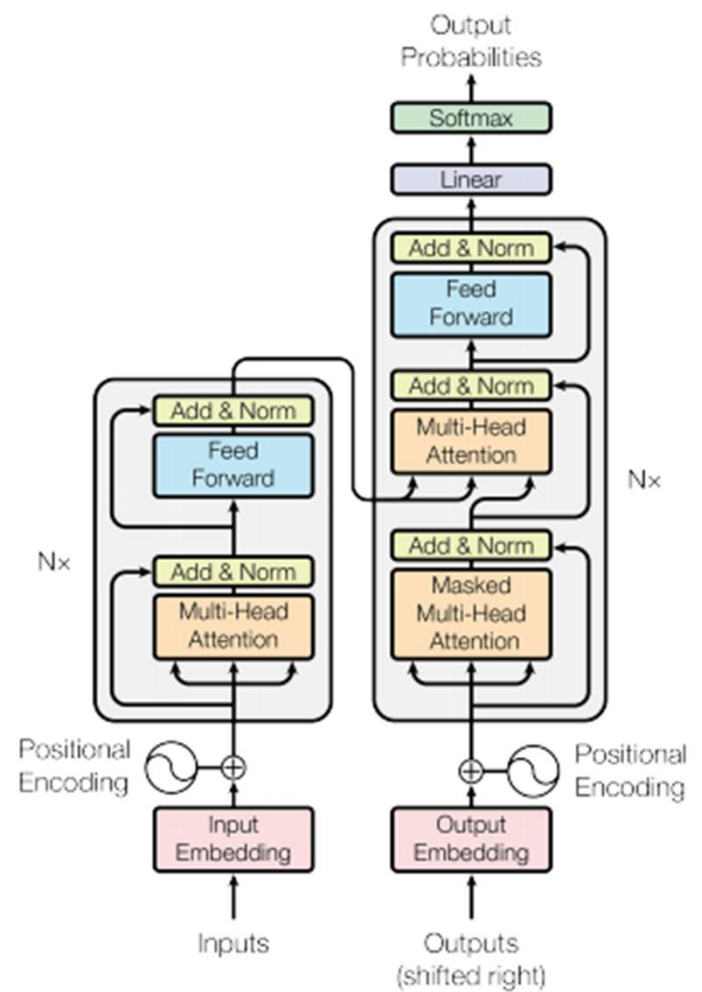
Seq2seq

Bert

GPT-3



Transformer





Transformers

Modelo Revolucionário

Utilizado em GPT-3, T5 e BERT

Não usa recorrência como RNN

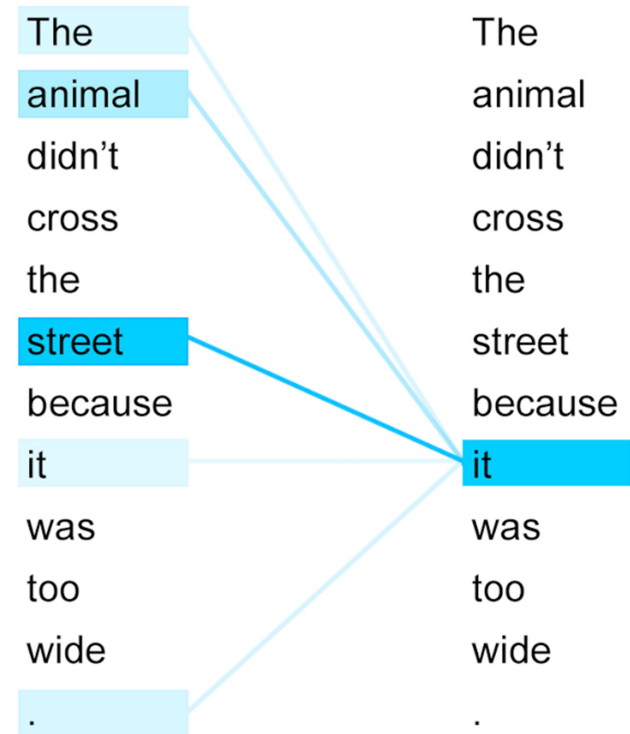
Basedo no mecanismo de atenção
“Attention”

Arquitetura ENDEC

Attention / Self Attention

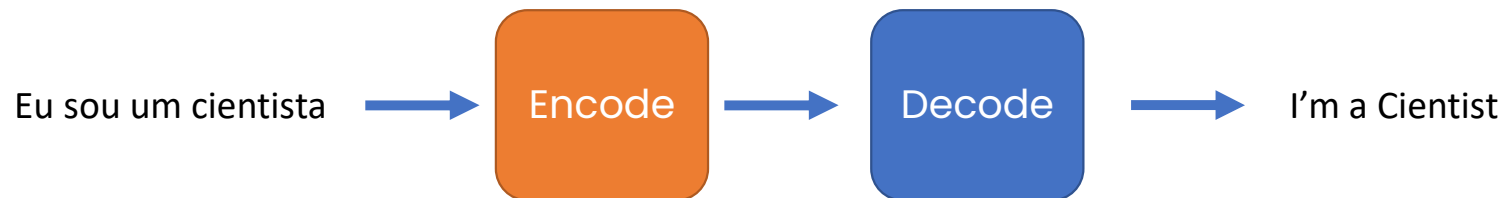
- Attention aprende a relação contextual entre palavras
- Word2vec não!
- Precisa ser treinado
- Existem modelos pré-treinados

<https://ai.googleblog.com/2017/08/transformer-novel-neural-network.html>

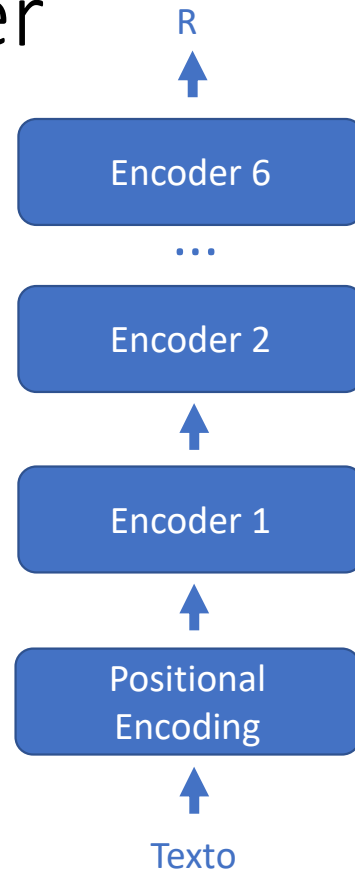


ENDEC: Encode / Decode

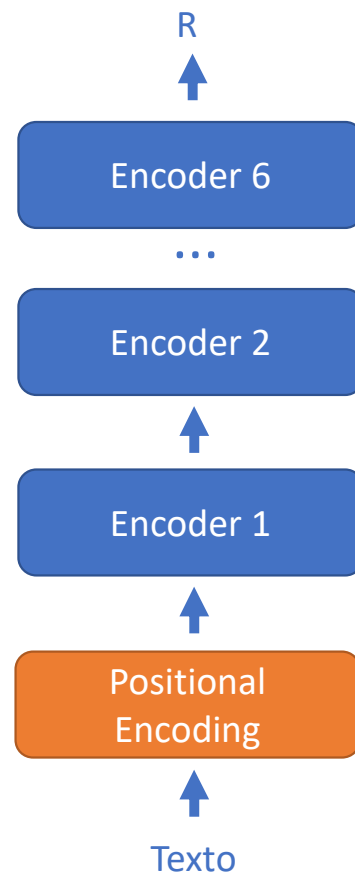
- Encoder : Colocar a mensagem em um formato que o computador compreenda
- Decoder: Colocar a mensagem em de volta ao formato original*



Transformer: Encoder



Positional Encoding



Positional Encoding

- NLP depende de ordem
- É natural processar em sequência
- A maioria das técnicas processam dados que dependem de Recorrência (RNN)
- Porém um Transformer submete texto (Embedding) em paralelo
 - Performance e melhoras no aprendizado

Positional Encoding

- Matriz com mesmas dimensões do embedding

The	0,42	0,70	0,04	...
Law	0,70	0,62	0,80	...
Will	0,17	0,04	0,41	...

Embedding Original

+

The	0,61	0,52	0,89	...
Law	0,17	0,29	0,16	...
Will	0,62	0,39	0,08	...

PE

=

The	1,04	1,22	0,93	...
Law	0,87	0,91	0,96	...
Will	0,79	0,43	0,49	...

Embedding Final

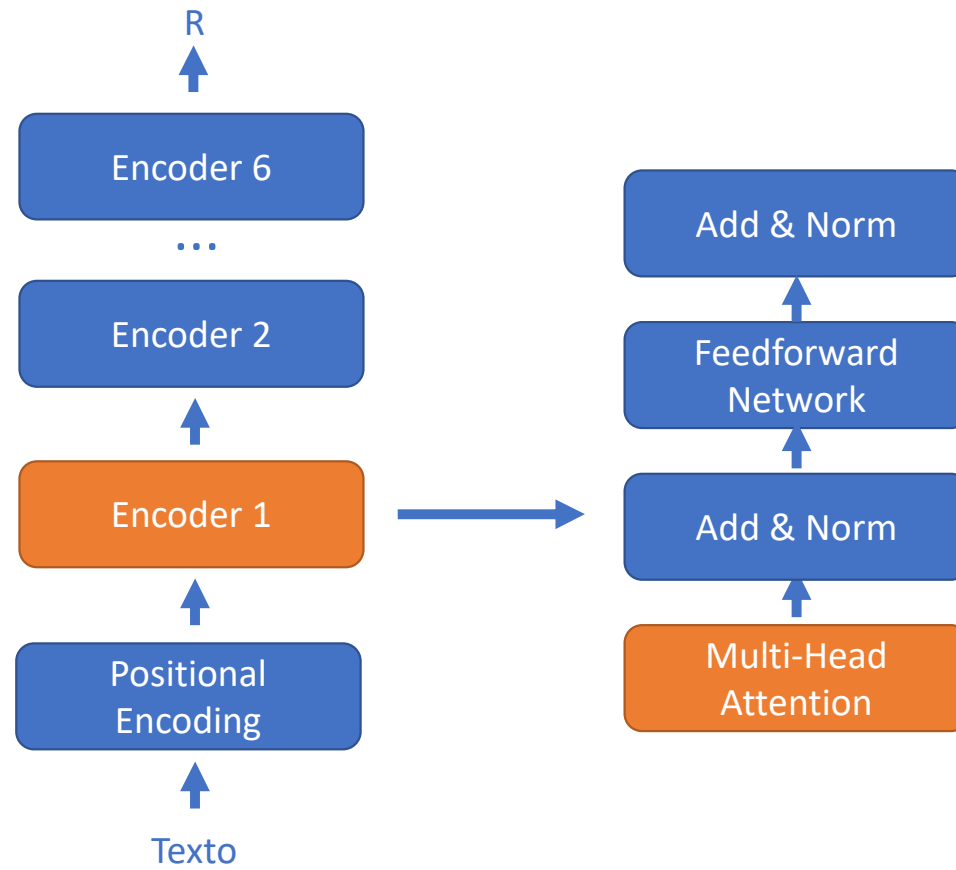
Positional Encoding

- Com a matriz PE é calculada?
 - Usa da função senoidal
 - pos é posição da palavra na sentença
 - i a posição no embedding
 - Seno quando é par e coseno quando é impar

$$PE_{(POS,2i)} = \sin\left(\frac{pos}{10000^{2i/d_{model}}}\right)$$

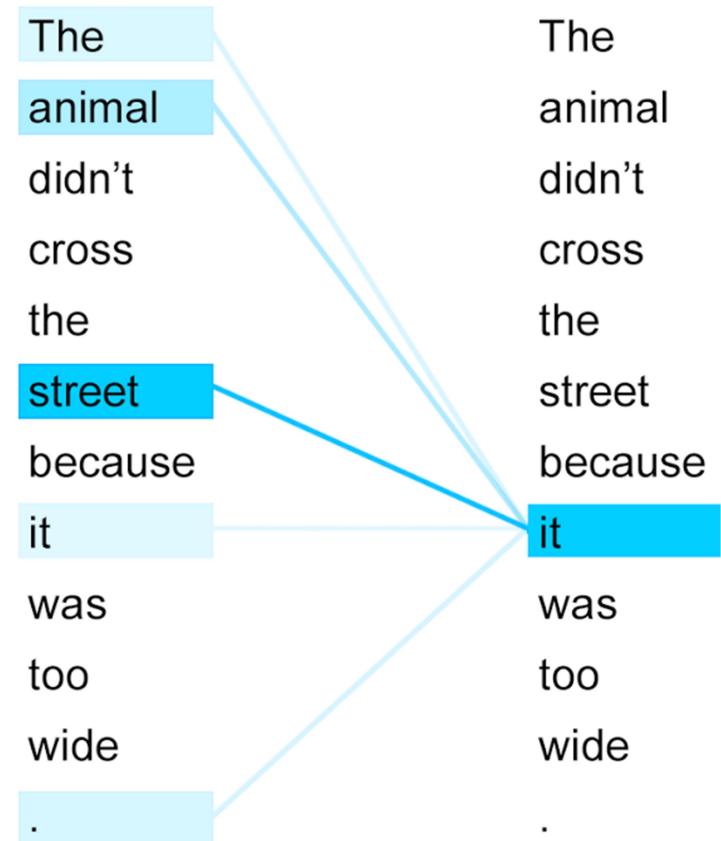
$$PE_{(POS,2i)} = \cos\left(\frac{pos}{10000^{2i/d_{model}}}\right)$$

Transformer



Self-attention

- A função é descobrir a relação entre as palavras
- *It* se refere a *animal* ou *street*?
- Como saber a qual se refere?
- O transformer calcula a representação e a relação de cada palavra
- Assim ele compreende que *it* se refere *street* e não a *animal*



Self-attention

- Self-attention é o resultado do processamento
- Mostra que *it* tem um peso maior para *street*
- Como ele descobre a relação?
 - 5 etapas!

1. Matrizes Q, K e V

- The Animal Didn't (...)
- Produz o embedding da sentença gerando matriz X
- Dimensões de X é o da sentença * tamanho do embedding [3 x 300]
- Três novas matrizes ponderadas são geradas
 - Q: query matrix
 - K: key matrix
 - V: Value matrix
- Pesos são ajustados durante o treinamento

animal	0,42	0,70	0,04	...
...				
street	0,70	0,62	0,80	...
...				
it	0,17	0,04	0,41	...

X (word embedding)



animal	0,44	0,47	0,97	...
...				
street	0,29	0,71	0,61	...
...				
it	0,40	0,87	0,41	...

W_q



animal	0,17	0,07	0,41	...
...				
street	0,61	0,08	0,79	...
...				
it	0,60	0,57	0,39	...

Query



animal	0,02	0,26	0,19	...
...				
street	0,07	0,06	0,44	...
...				
it	0,69	0,08	0,39	...

W_k



animal	0,77	0,64	0,80	...
...				
street	1,08	1,17	1,93	...
...				
it	0,93	0,29	0,73	...

Key



animal	0,17	0,93	0,02	...
...				
street	0,23	0,80	0,13	...
...				
it	0,37	0,51	0,80	...

W_v



animal	0,33	0,95	0,84	...
...				
street	0,36	0,64	0,36	...
...				
it	0,18	0,31	0,55	...

Value



2º: Score: Produto escalar entre Query e Key

- Mostra o grau de similaridade entre palavras

animal	0,17	0,07	0,41	...
...				
street	0,61	0,08	0,79	...
...				
it	0,60	0,57	0,39	...

Query

x

animal	0,77	0,64	0,80	...
...				
street	1,08	1,17	1,93	...
...				
it	0,93	0,29	0,73	...

Key

=

animal	0,33	0,95	0,84	...
...				
street	0,36	0,64	0,36	...
...				
it	0,18	0,31	0,55	...

Value

3º: Dividir QK pela raiz quadrada das dimensões de Key

- Objetivo: obter bons gradientes

$$\frac{QK^T}{\sqrt{D_X}}$$

4º Produção da Score Matrix

- Normalização entre 0 e 1 com softmax
- Objetivo é gerar um score de relação entre as palavras

$$\text{softmax} \left(\frac{QK^T}{\sqrt{D_K}} \right)$$

animal	0,17	0,07	0,41	...
...				
street	0,61	0,08	0,79	...
...				
it	0,60	0,57	0,39	...

Score

5º: Calcular a matriz Z (matriz de atenção)

- Cálculo é Score Matrix * V
- Cada palavra terá um score: Z_1, Z_2, Z_3
- Mostra o peso da relação entre as palavras

	animal	street	it	
animal	0,83	0,07	0,41	...
...				
street	0,61	0,08	0,79	...
...				
it	0,60	0,57	0,39	...

Score

x

	animal	street	it	
animal	0,33	0,95	0,84	...
...				
street	0,36	0,64	0,36	...
...				
it	0,18	0,31	0,55	...

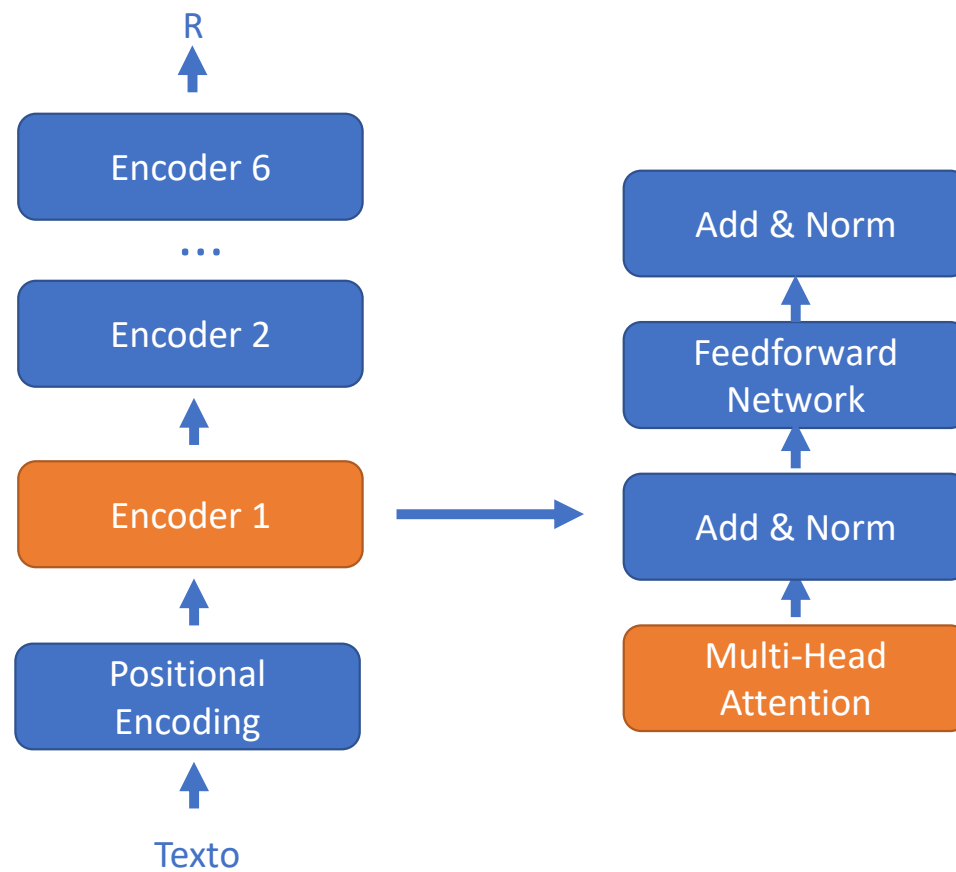
Value

Attention

$$\text{softmax} \left(\frac{QK^T}{\sqrt{D_K}} \right) * V$$

- A matriz Z está pronta para ser enviada através da próxima camada Feedforward Network (porém antes será normalizada)

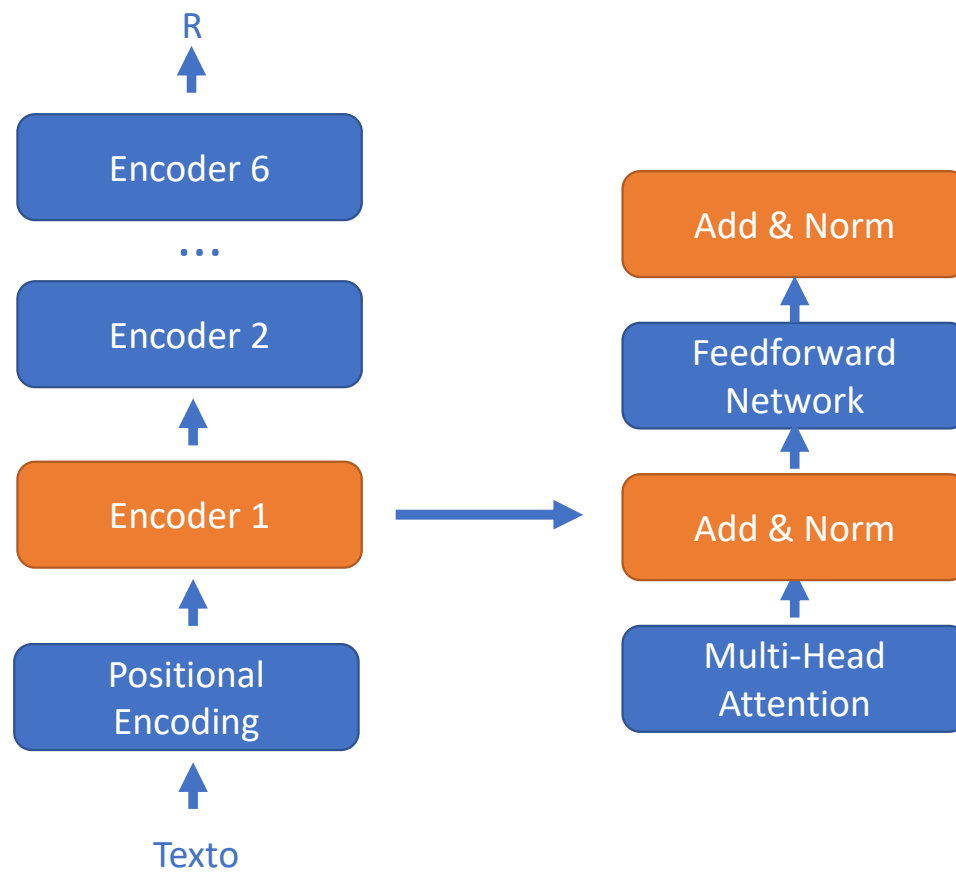
Encoder



Multi Attention

- Em vez de um processo Attention, múltiplos processos são processados e o resultado é somando
- Aumenta a precisão do sistema
- Todas as matrizes são concatenadas e multiplicadas por uma matriz ponderada, pois a RNA espera apenas uma matriz

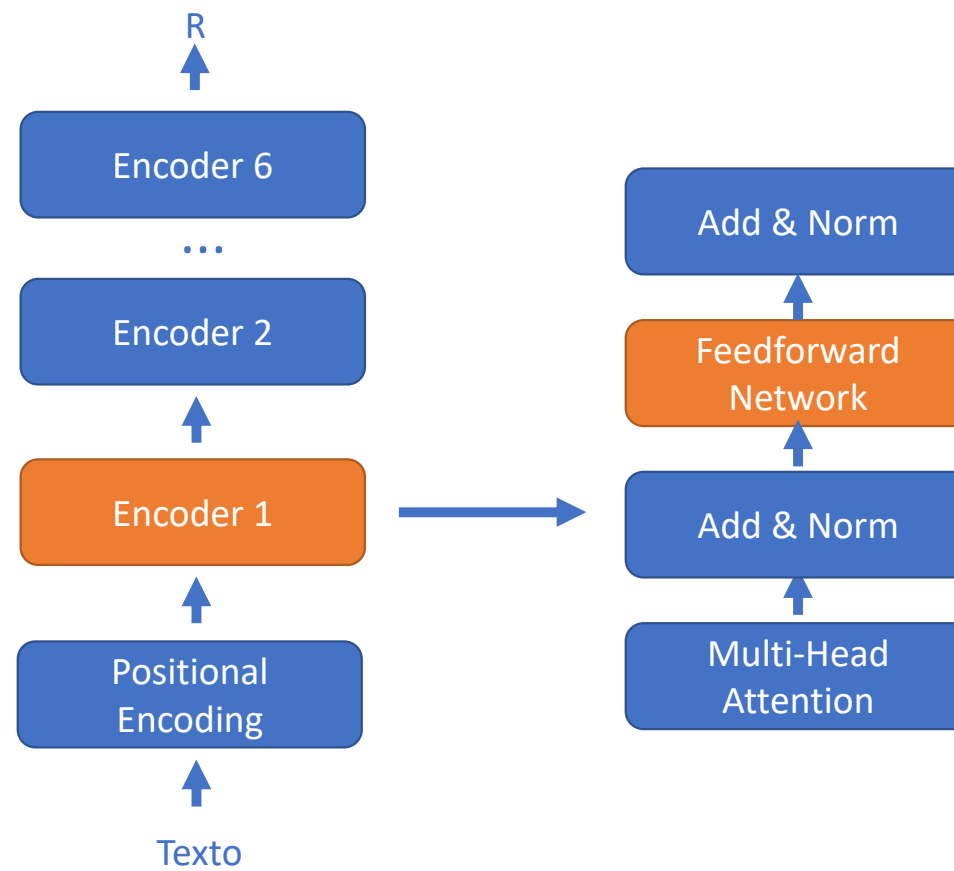
Encoder



Add & Norm

- Camada de Normalização
- Objetivo é conectar e normalizar
- Conecta a entrada do multi head attention com sua saída
- Conectada a entrada da feedforward com sua saída

Encoder



Feedforward network

- Duas camadas densas com Relu