BERT

And other language models

Boris Zubarev

@bobazooba

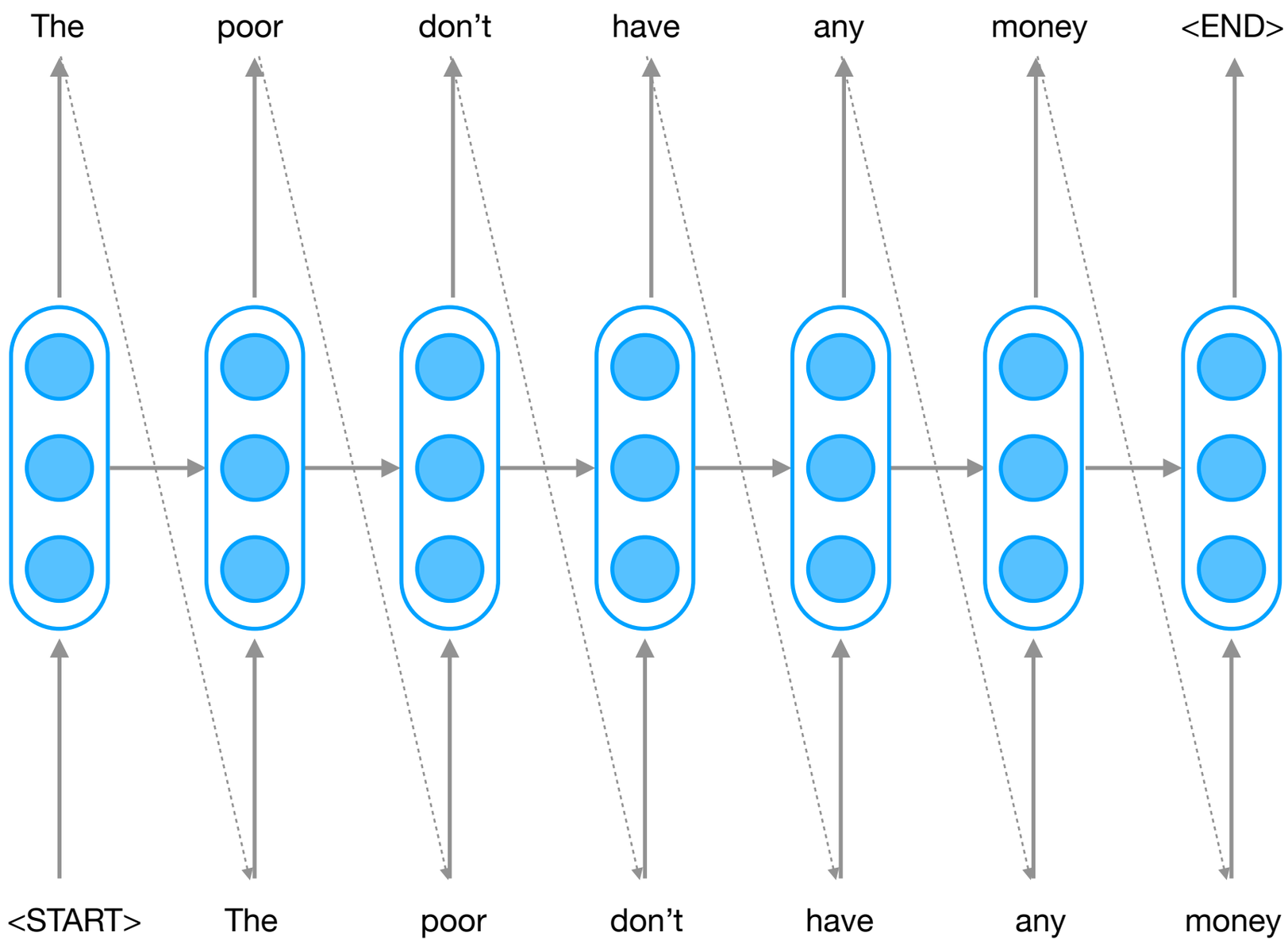
BERT

And other Sesame Street characters

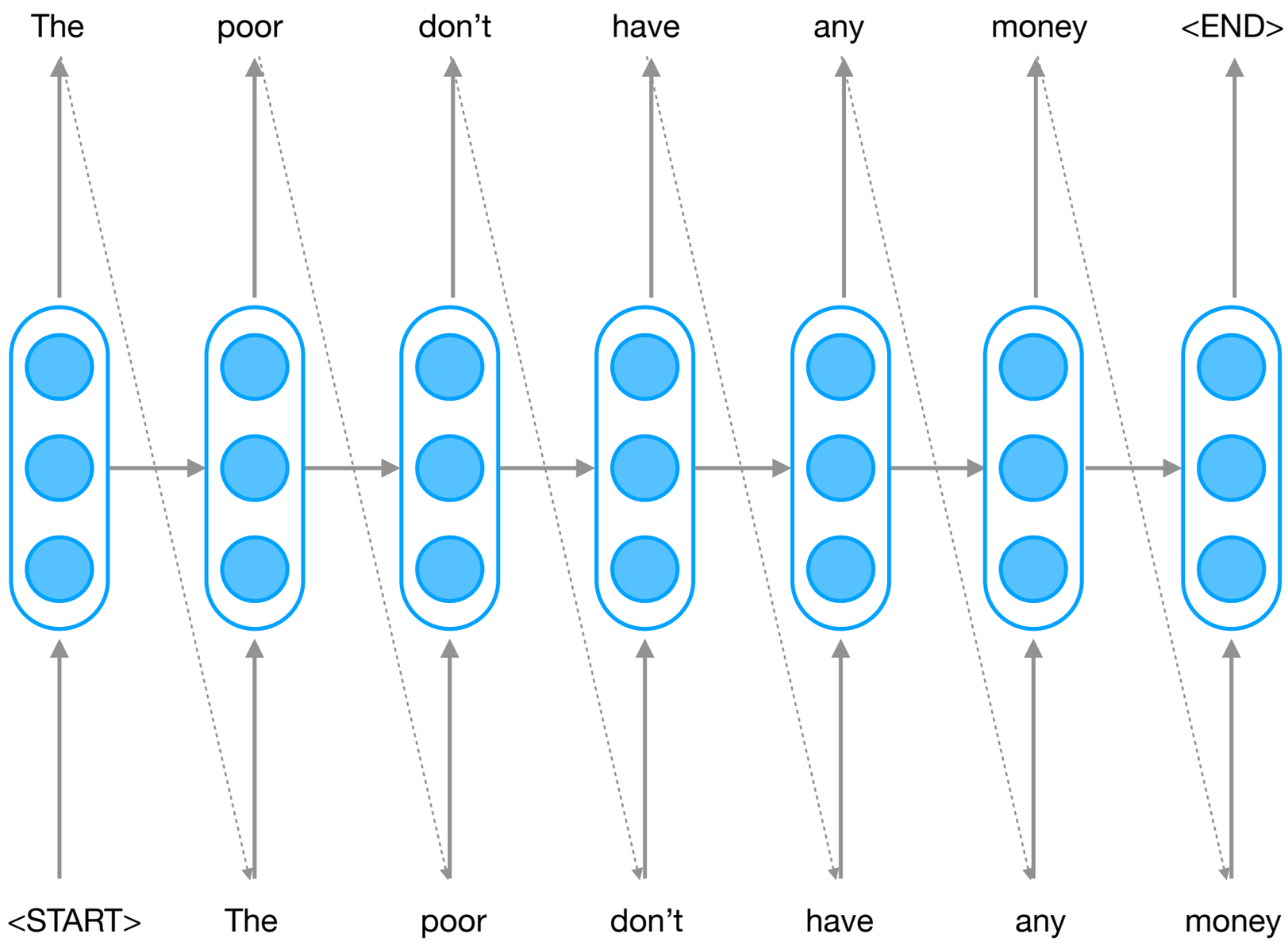
Boris Zubarev

@bobazooba

Language Model



OpenAI GPT

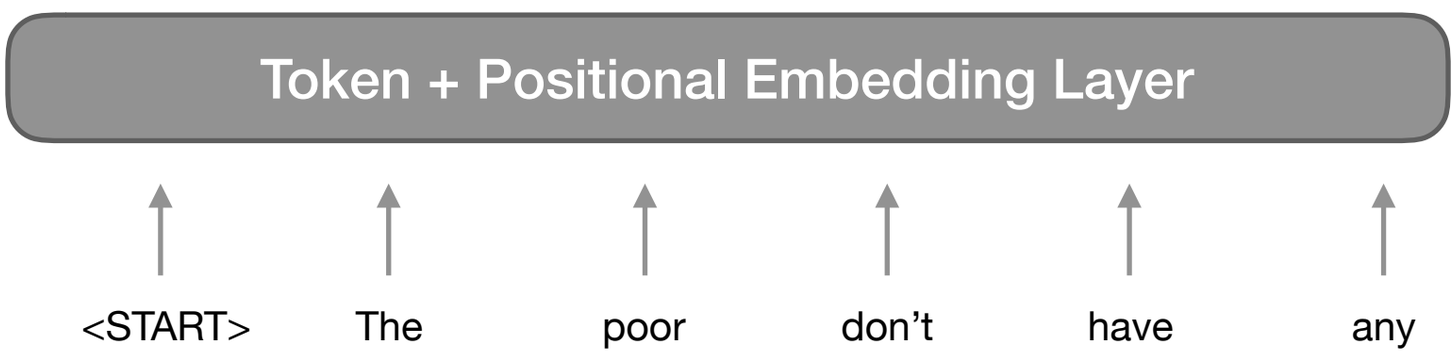

The
poor
don’t
have
any
money
<END>
Transformer Encoder With Masking
Embedded Tokens
<START>
The
poor
don’t
have
any
money

OpenAI GPT

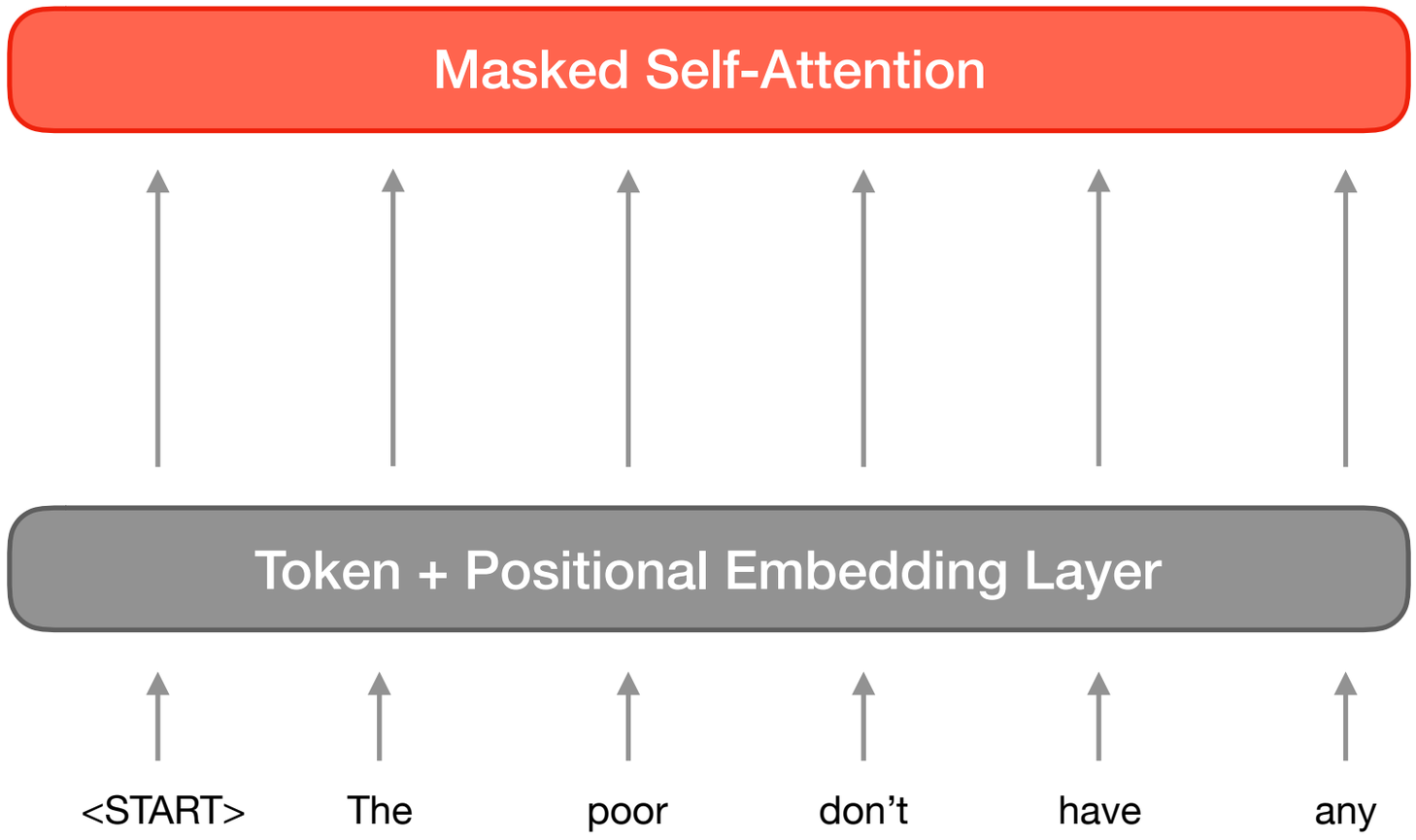
OpenAI GPT

<START> The poor don’t have any

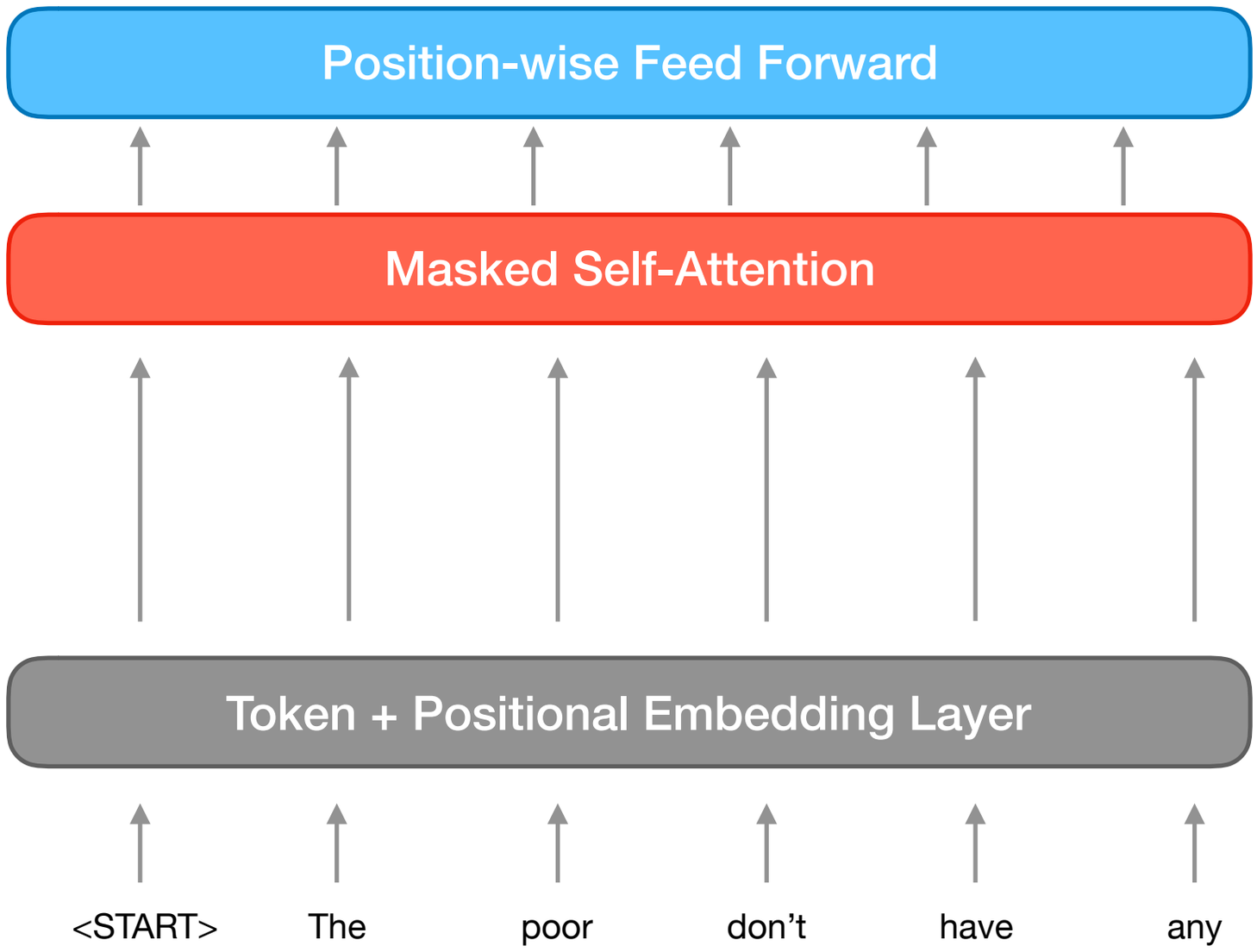
OpenAI GPT


Token + Positional Embedding Layer
<START>
The
poor
don’t
have
any

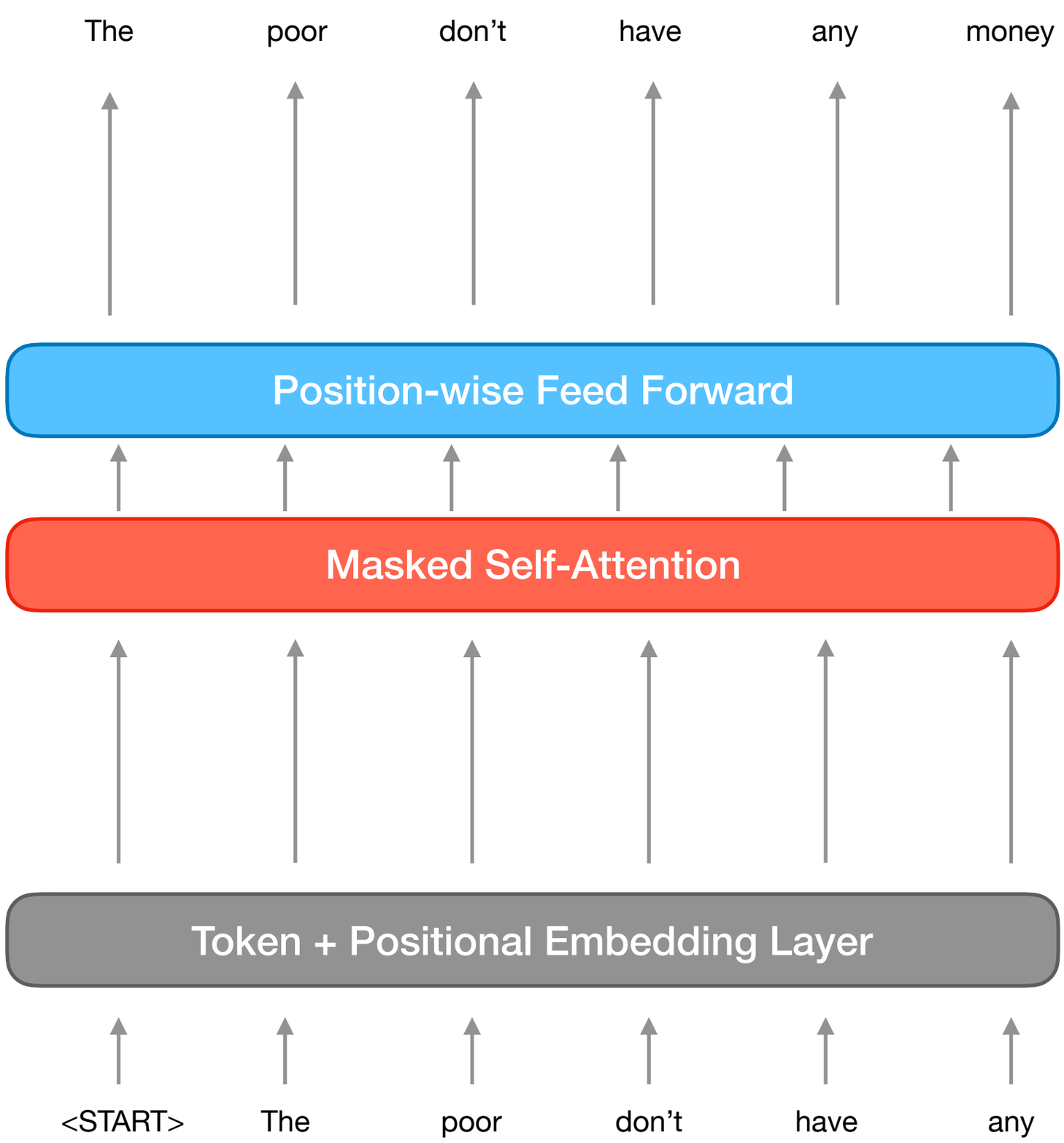
OpenAI GPT


Masked Self-Attention
Token + Positional Embedding Layer
<START>
The
poor
don’t
have
any

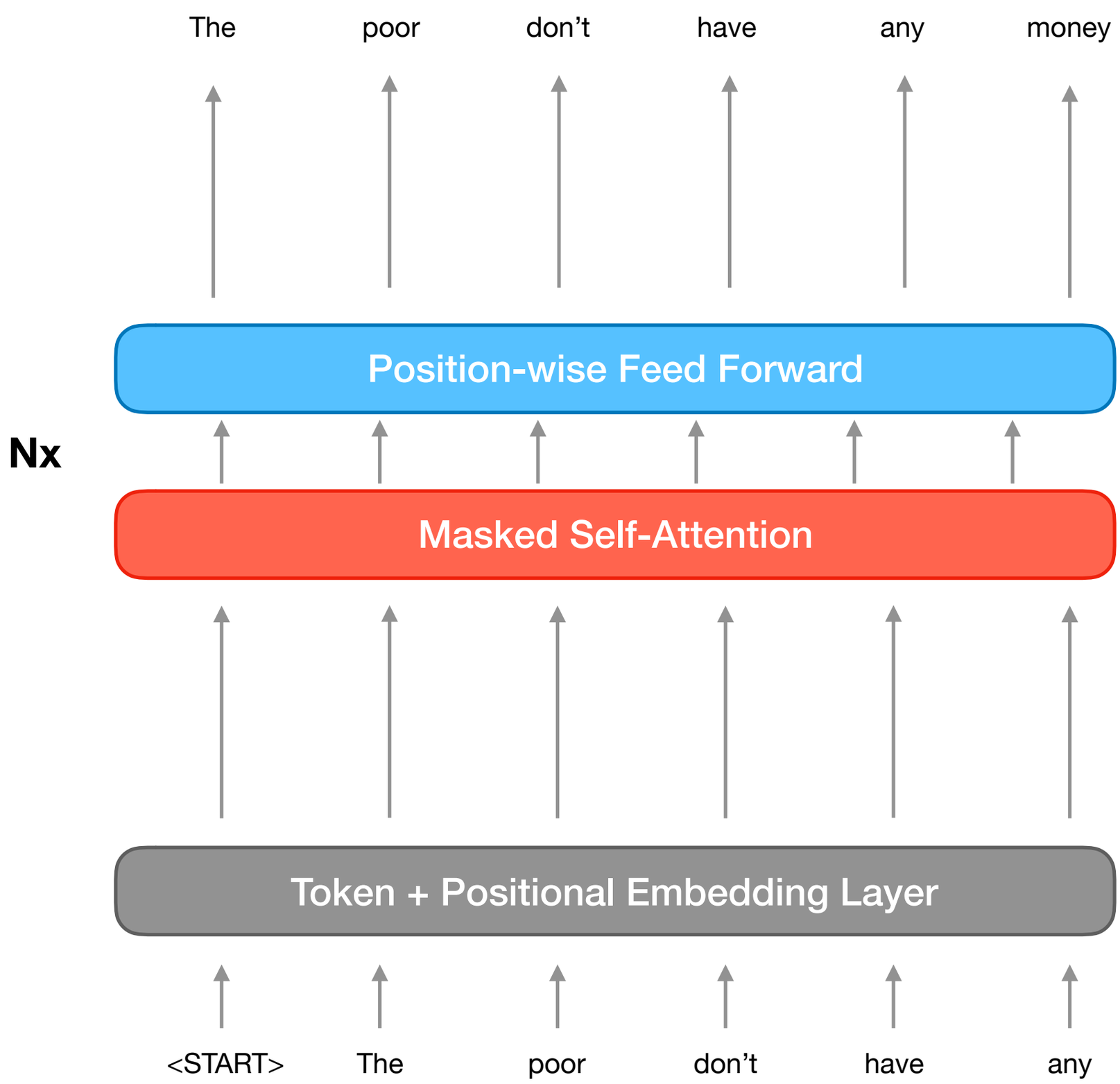
OpenAI GPT


Position-wise Feed Forward
Masked Self-Attention
Token + Positional Embedding Layer
<START>
The
poor
don’t
have
any

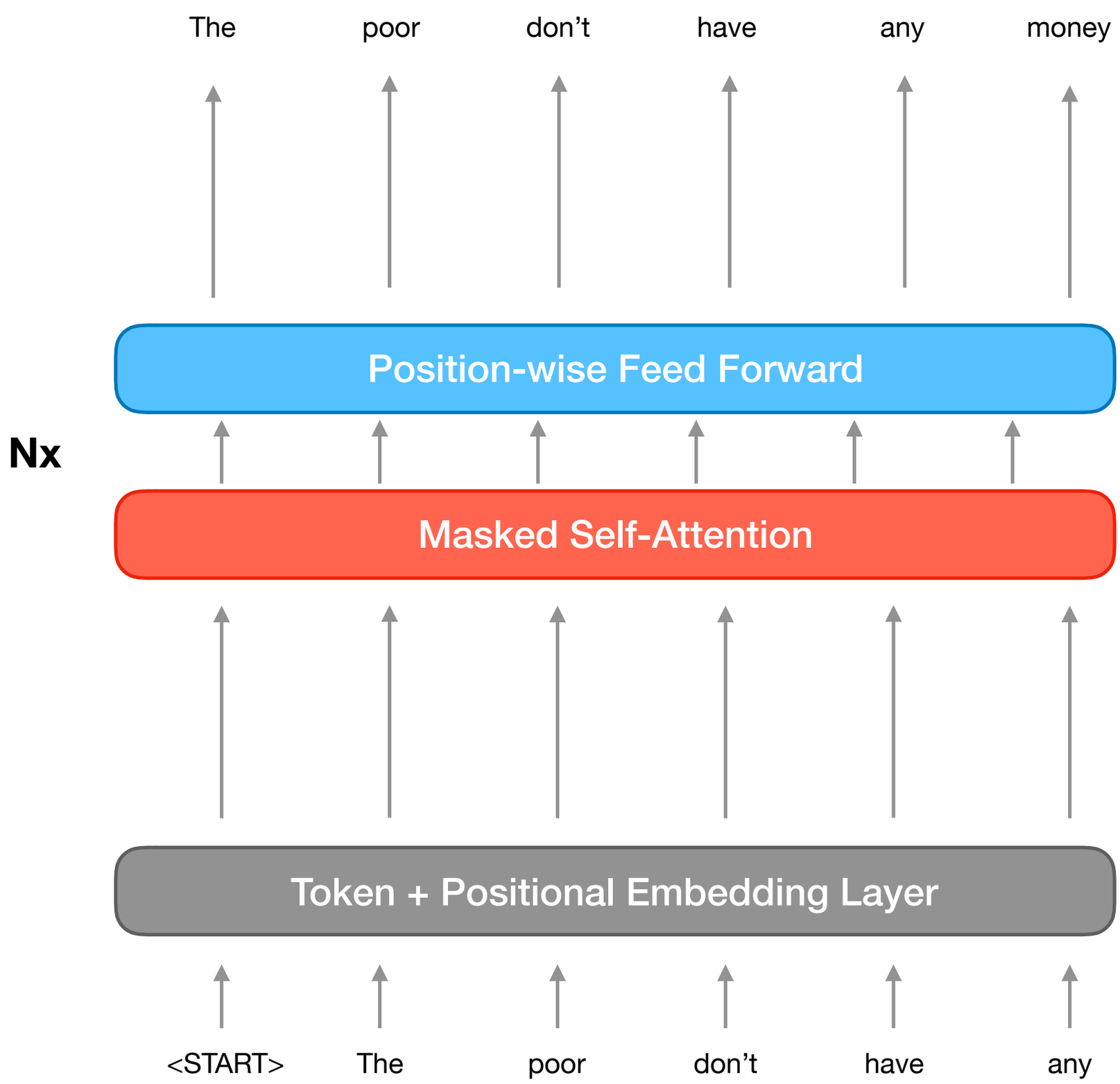
OpenAI GPT


The
poor
don’t
have
any
money
Position-wise Feed Forward
Masked Self-Attention
Token + Positional Embedding Layer
<START>
The
poor
don’t
have
any

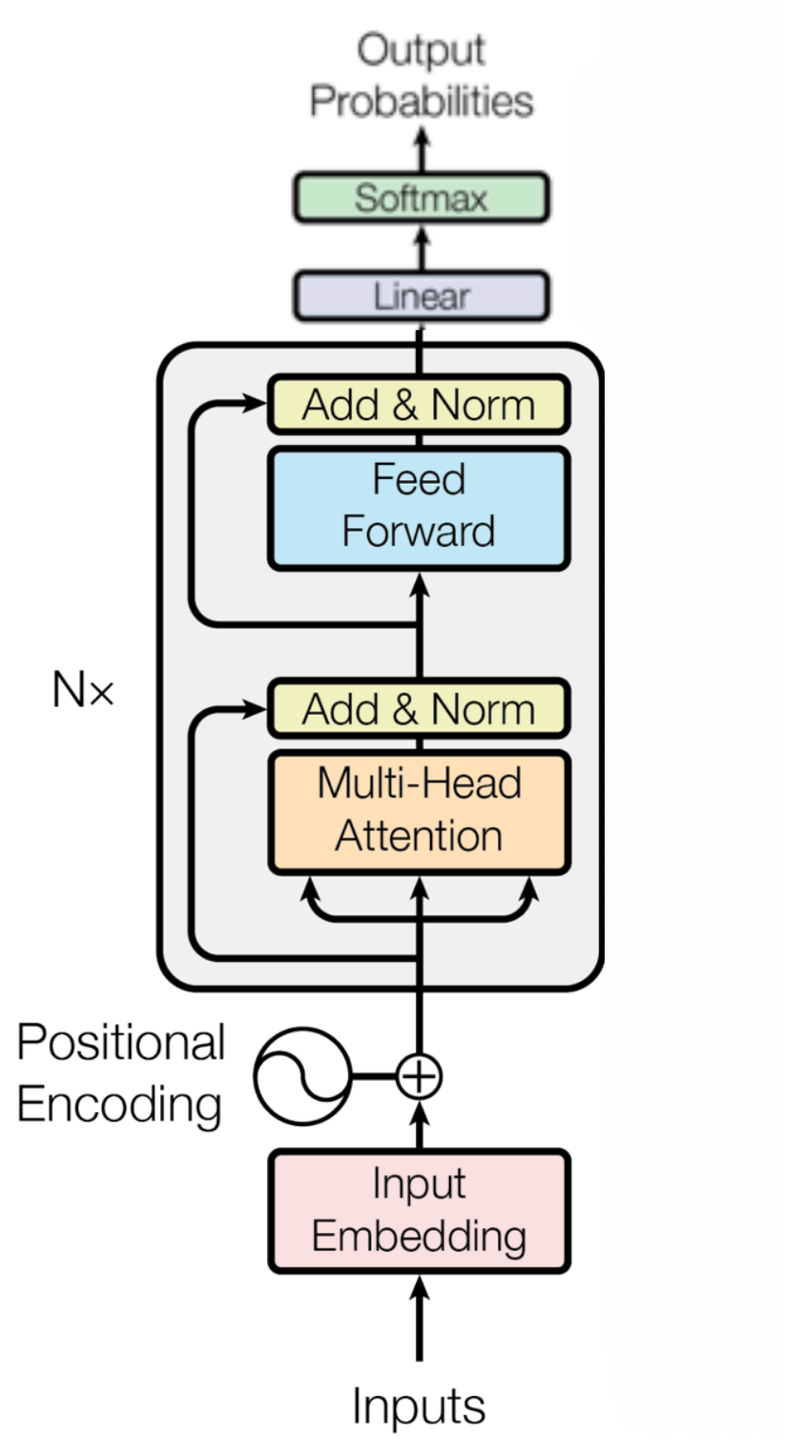
OpenAI GPT


The
poor
don’t
have
any
money
Position-wise Feed Forward
Nx
Masked Self-Attention
Token + Positional Embedding Layer
<START>
The
poor
don’t
have
any

OpenAI GPT

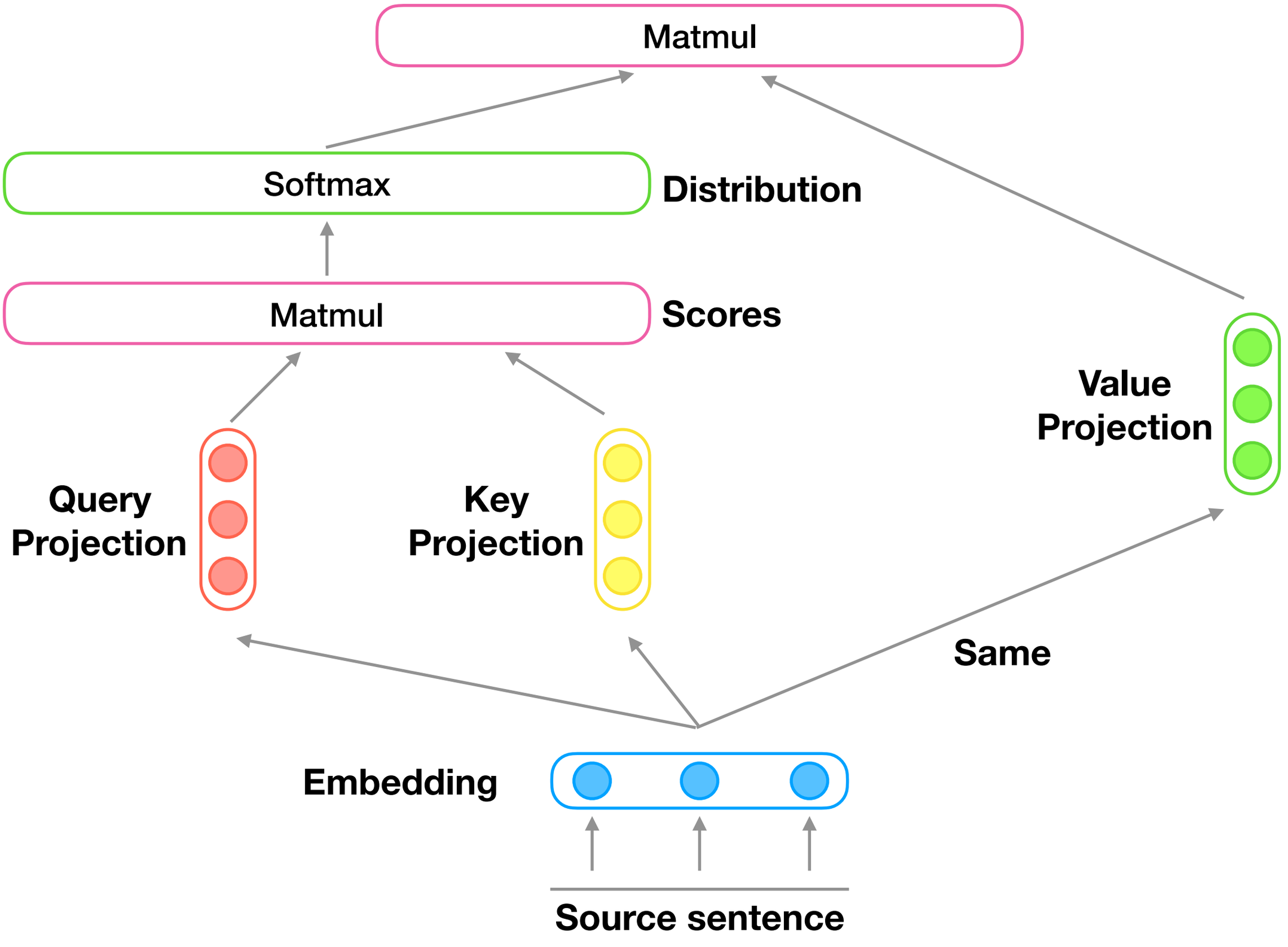

The
poor
don’t
have
any
money
Position-wise Feed Forward
Nx
Masked Self-Attention
Token + Positional Embedding Layer
<START>
The
poor
don’t
have
any

GPT

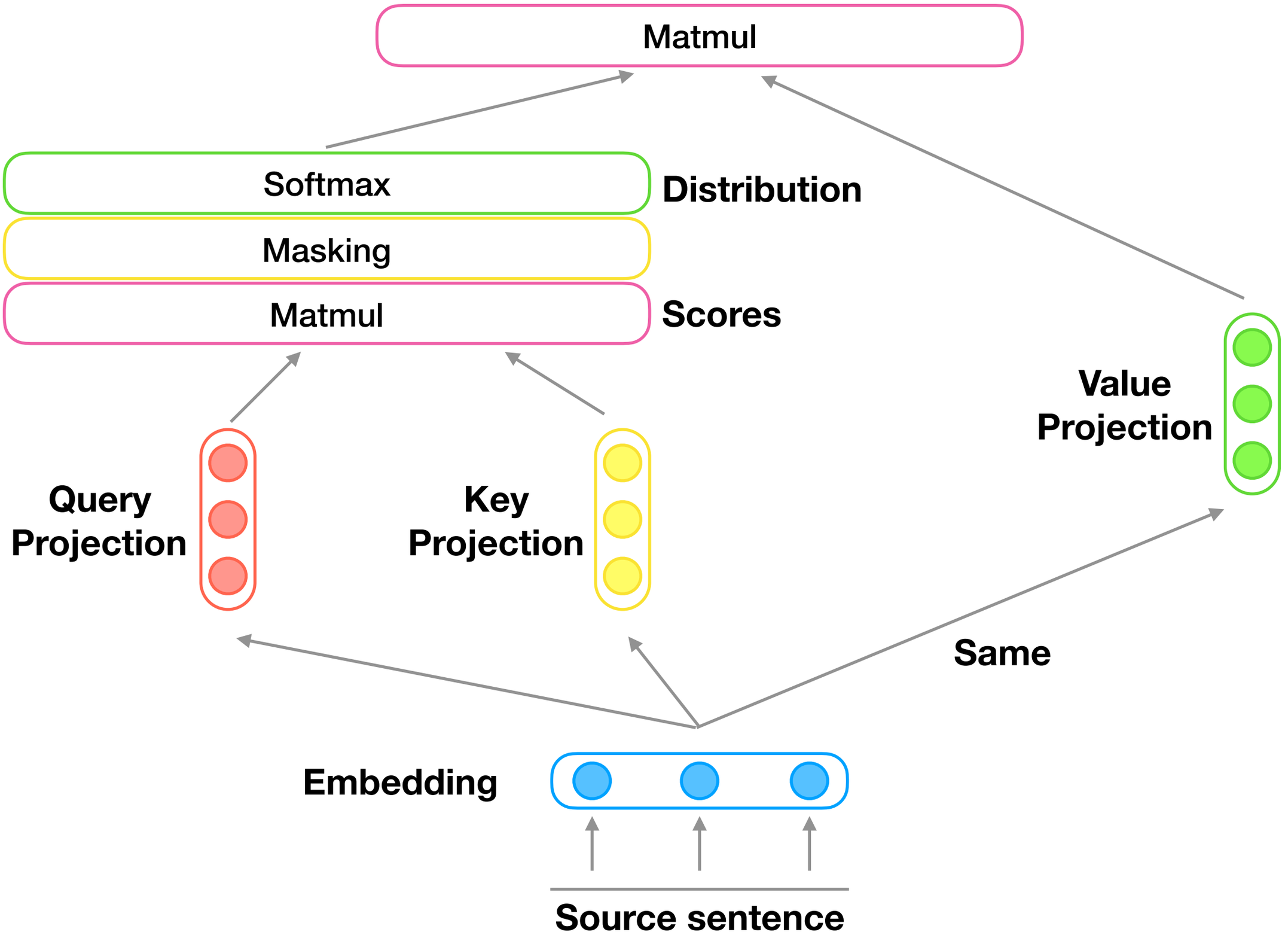

Output Probabilities
Softmax
Linear
Add & Norm
Feed Forward
Add & Norm
Multi-Head Attention
Nx
Positional Encoding
Input Embedding
Inputs

Masking

Self-Attention



Masked Self-Attention



Masking

Source text I am space invader

Attention Scores

0.11	0.04	0.05	0.3
0.19	0.53	0.42	0.37
0.81	0.21	0.05	0.09
0.51	0.43	0.12	0.03

Masking

Future

Masked Attention Scores

Time

0.11	-inf	-inf	-inf
0.19	0.53	-inf	-inf
0.81	0.21	0.05	-inf
0.51	0.43	0.12	0.03

Softmax

Attention Distribution

1	0	0	0
0.48	0.52	0	0
0.45	0.21	0.34	0
0.25	0.16	0.33	0.26

BERT

- New task — masked language modelling
- Bidirectional language model
- Auxiliary task — next sentence prediction
- Deeper

Masked Language Model

The poor don’t have any money

Masked Language Model

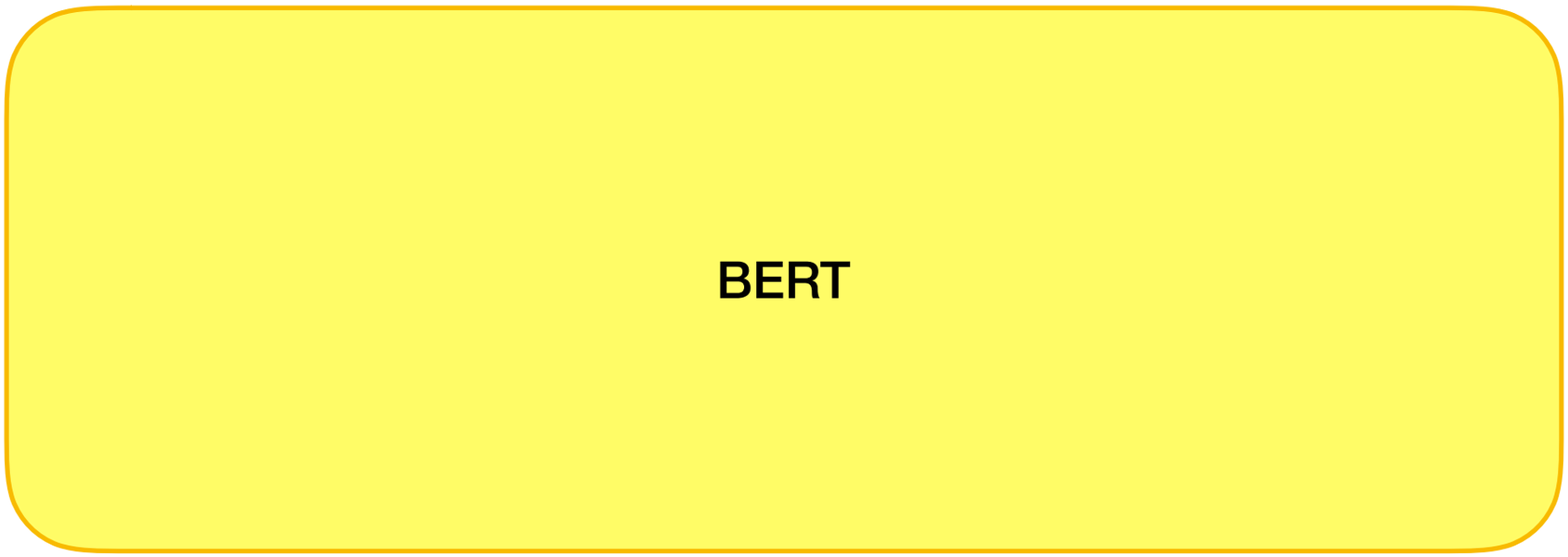
The poor don’t have any money

Masked Language Model

- Masking 15% tokens (20% of them by random token)

Masked Text	The	[MASK]	don’t	have	any	[MASK]

Masked Language Model



Masked Text The [MASK] don’t have any [MASK]

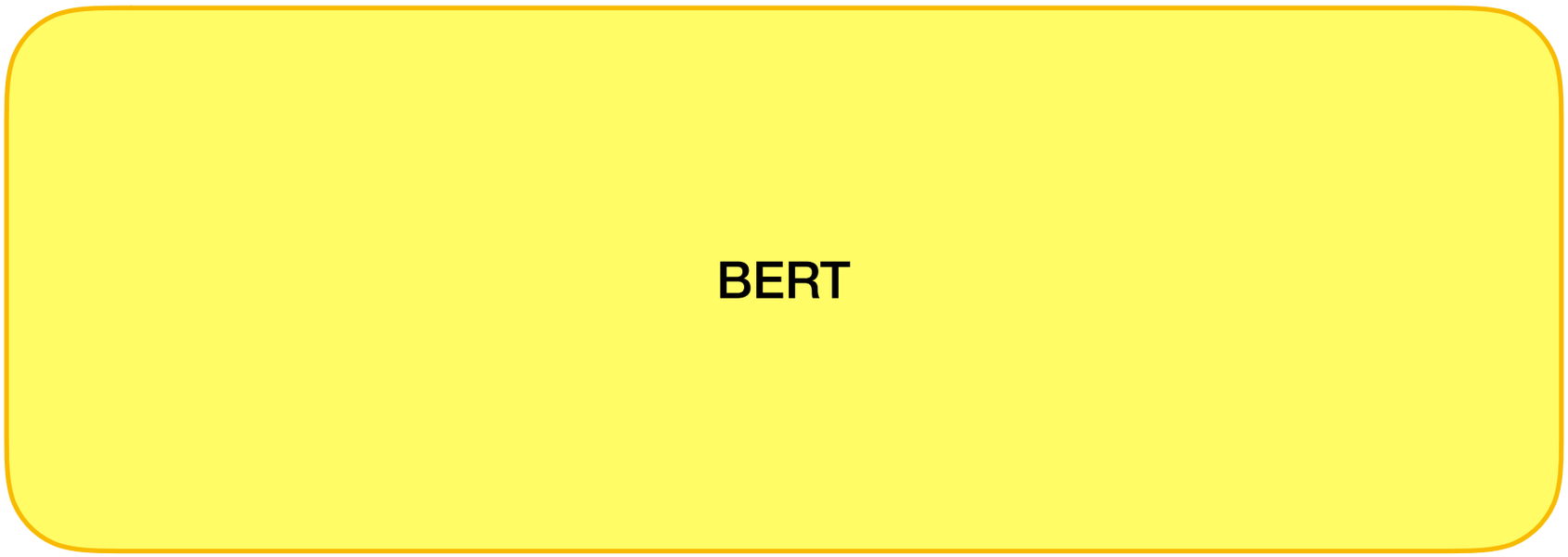
Masked Language Model

Target Text	The	poor	don’t	have	any	money

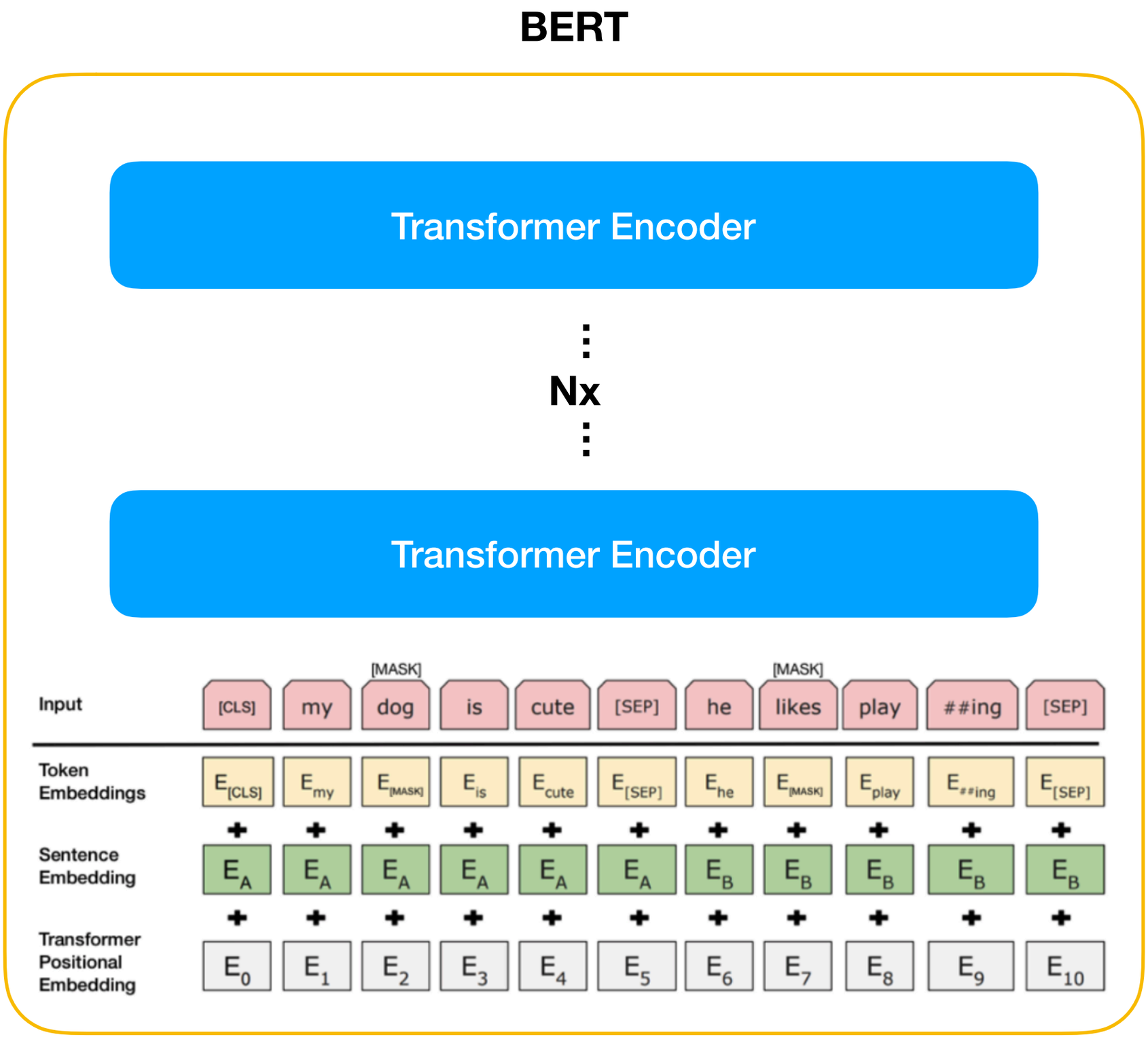
BERT

Masked Text	The	[MASK]	don’t	have	any	[MASK]

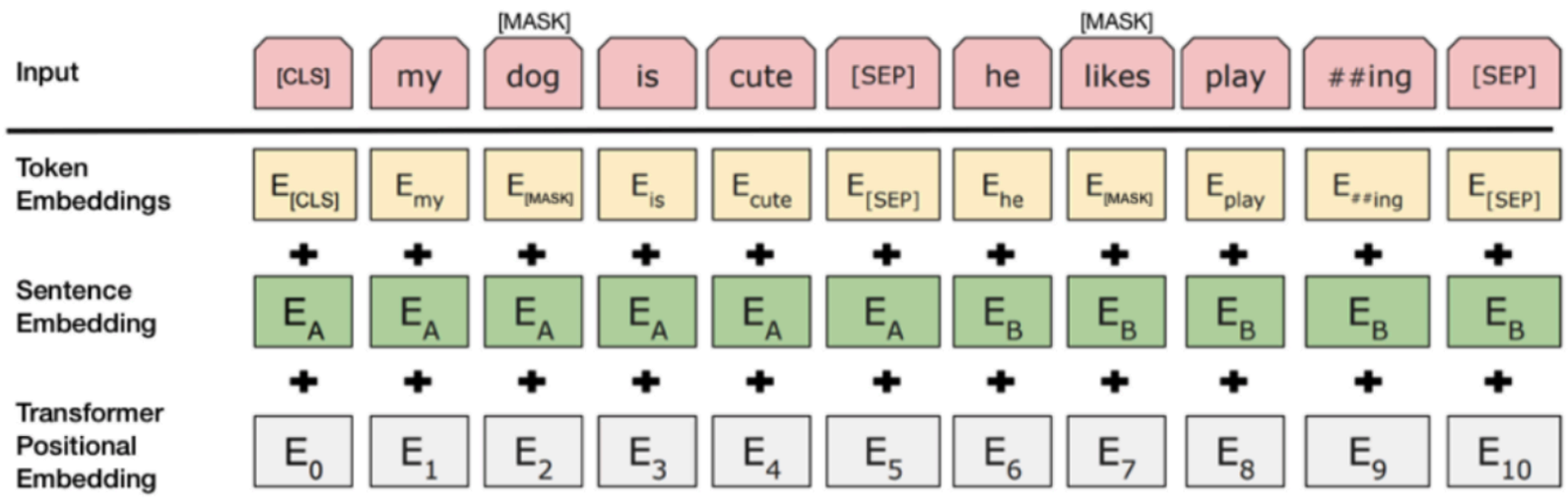
Next Sentence Prediction



Next Sentence Prediction

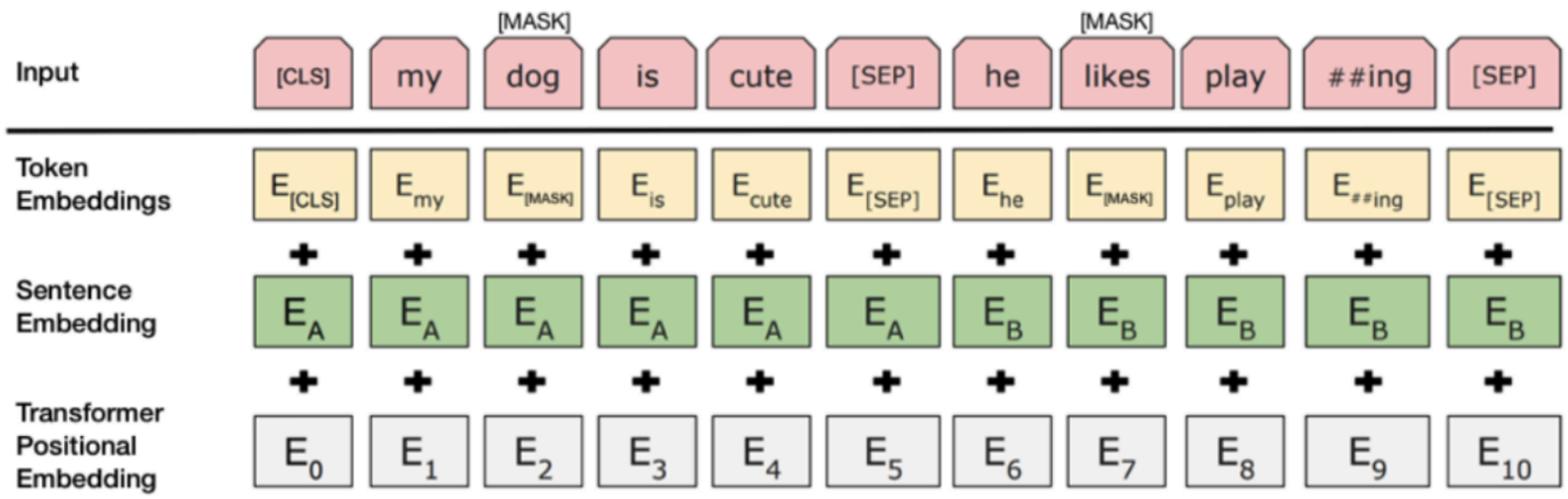


Next Sentence Prediction



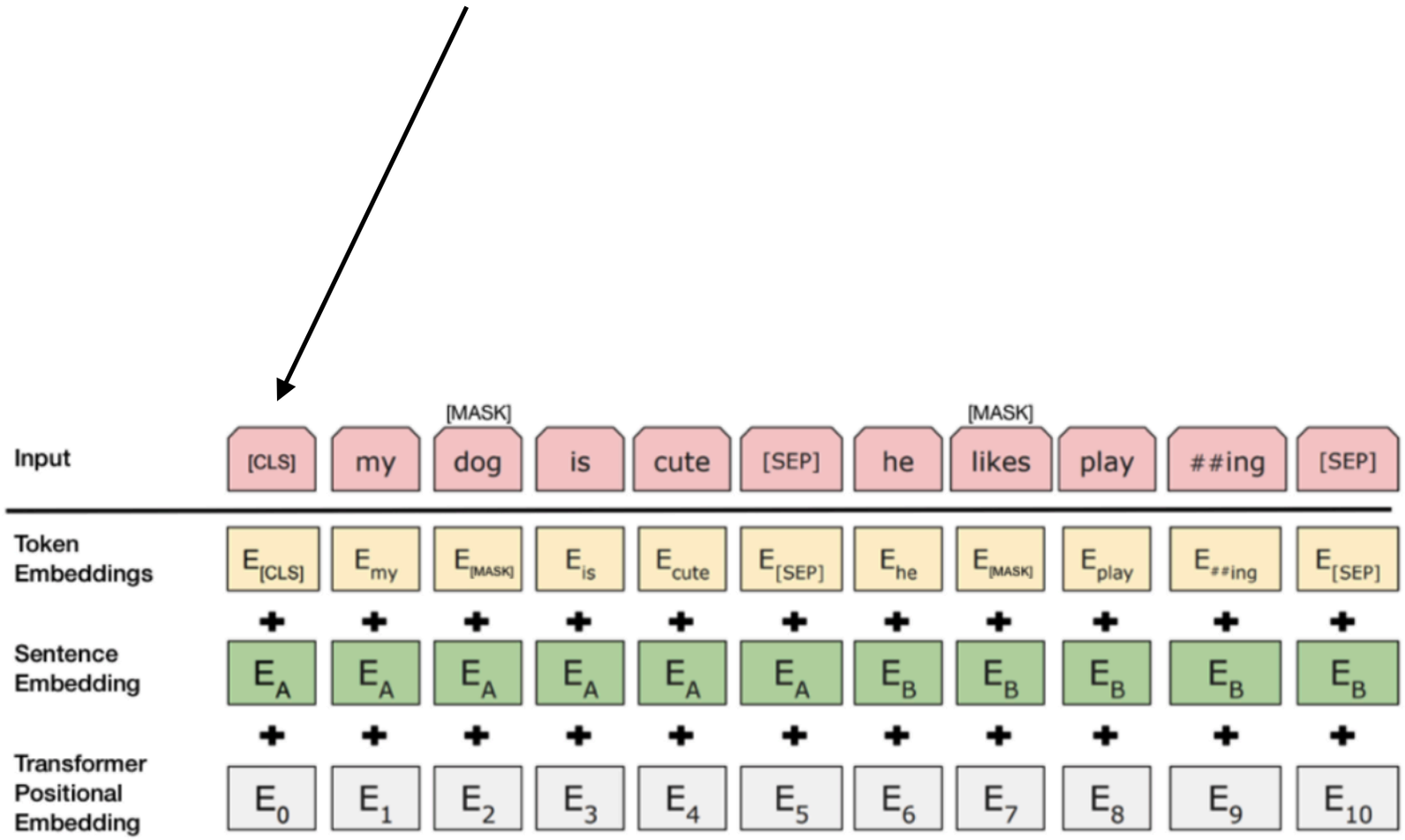
Next Sentence Prediction

- 50% actually next sentence
- 50% randomly sampled from corpus
- NSP token — [CLS]



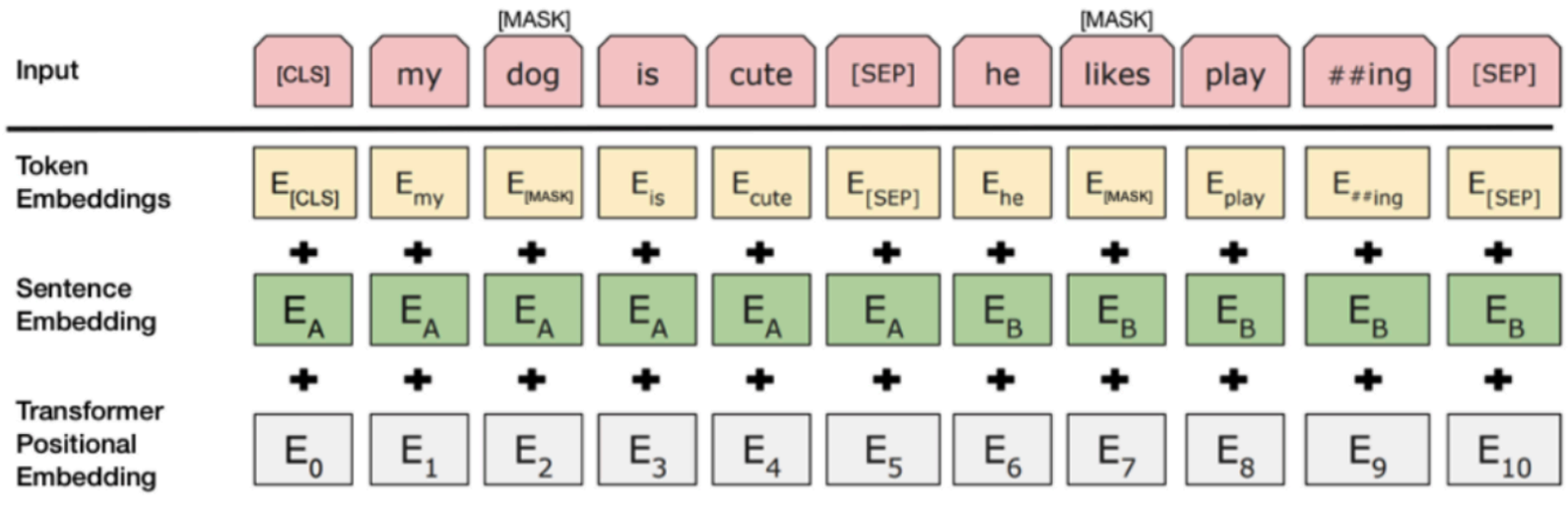
Next Sentence Prediction

- 50% actually next sentence
- 50% randomly sampled from corpus
- NSP token — [CLS]



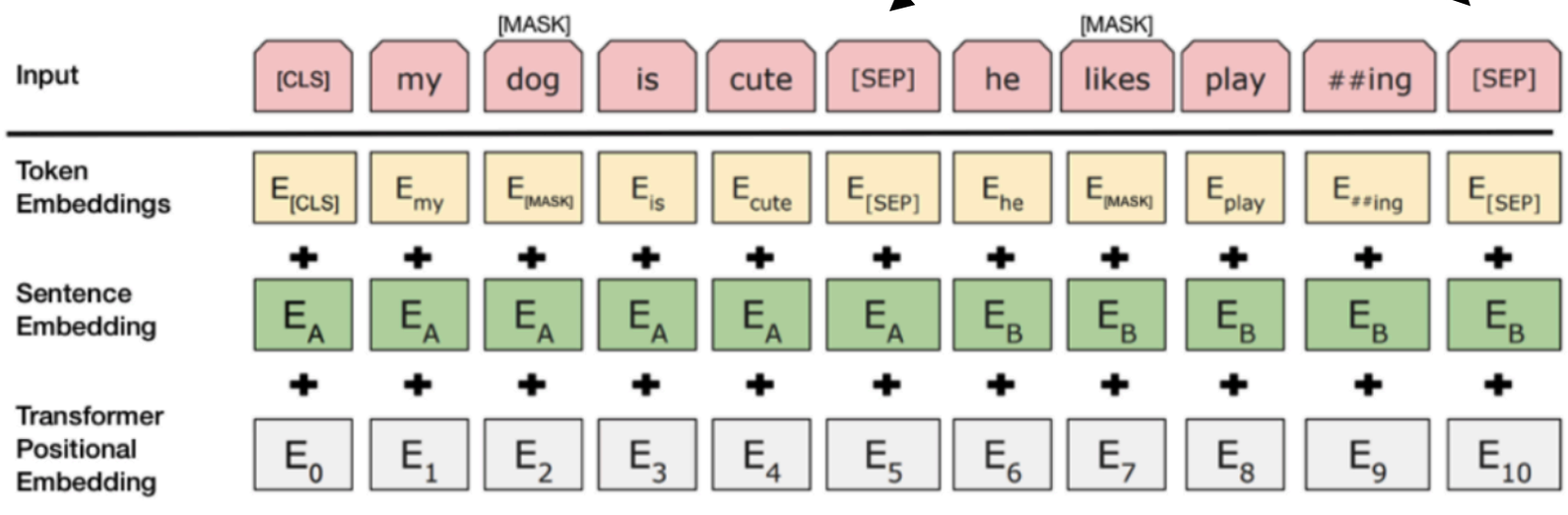
Next Sentence Prediction

- 50% actually next sentence
- 50% randomly sampled from corpus
- NSP token — [CLS]
- Token for separating sentence — [SEP]

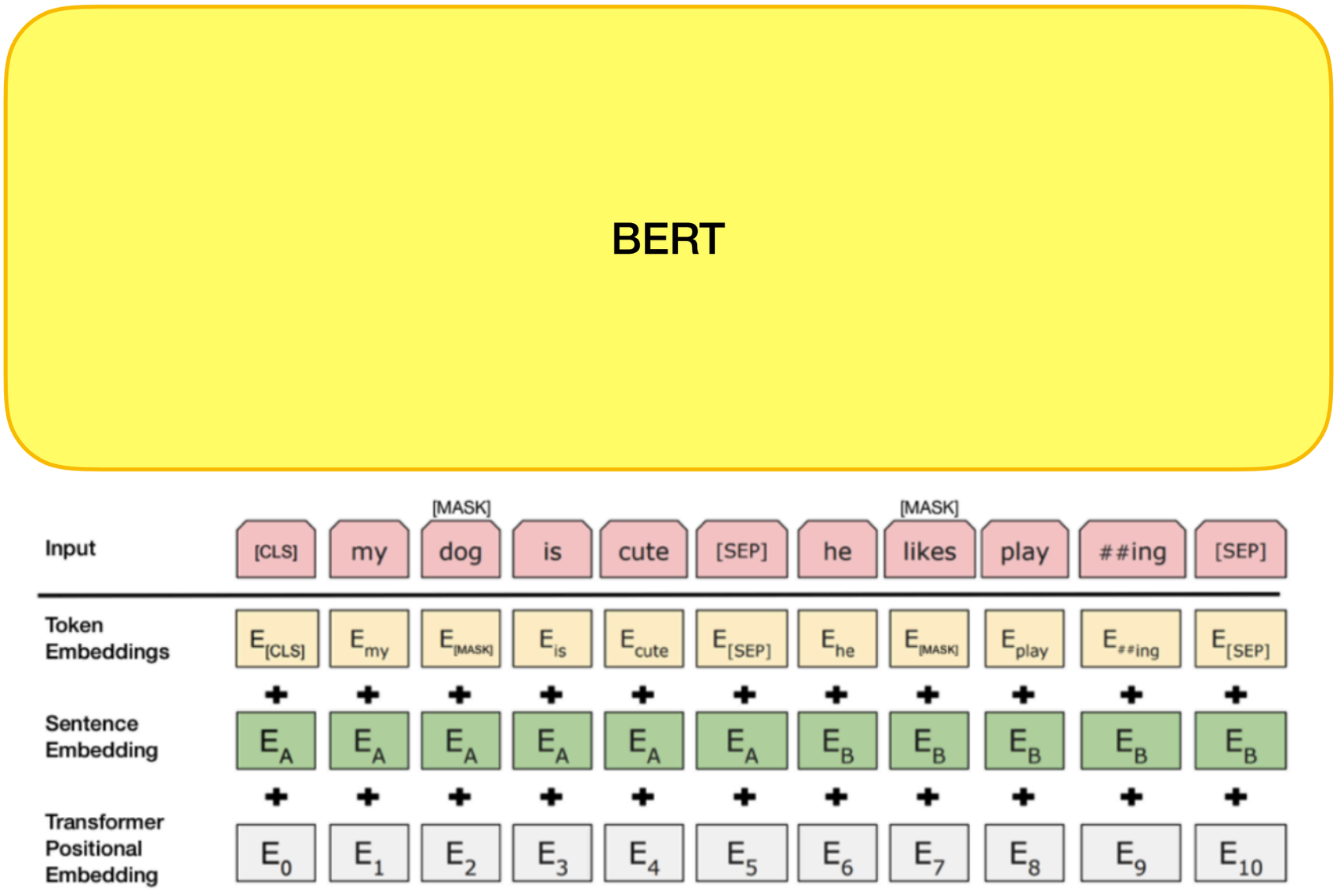


Next Sentence Prediction

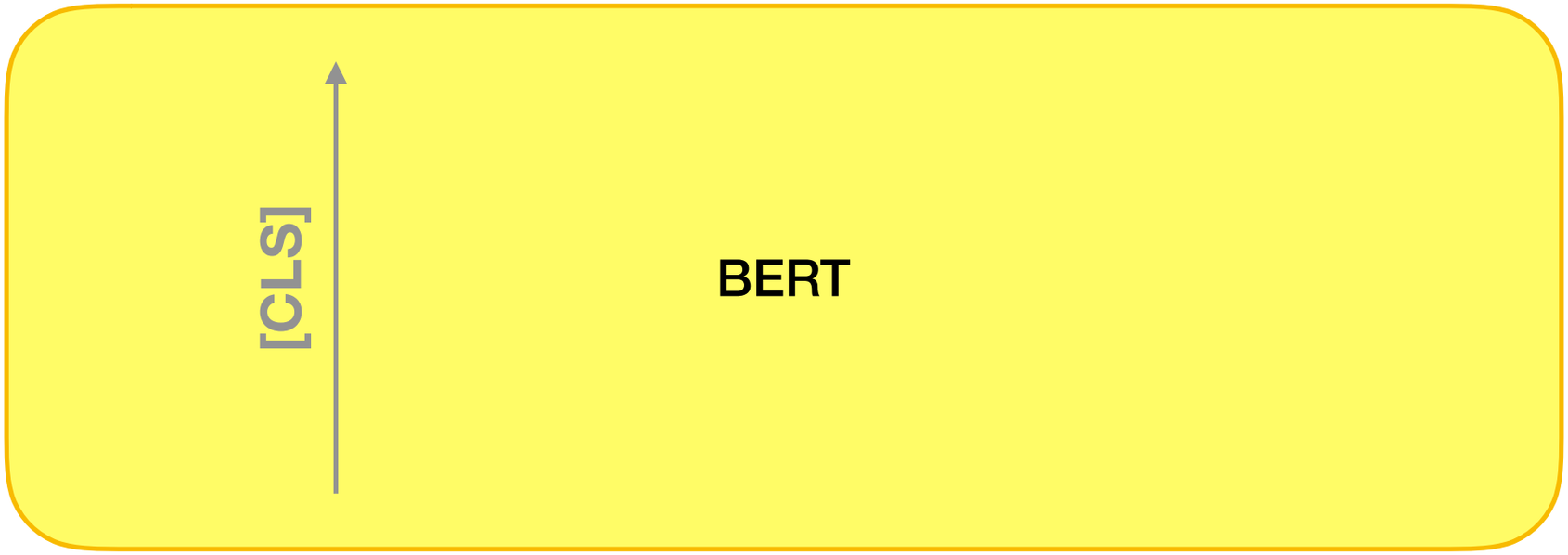
- 50% actually next sentence
- 50% randomly sampled from corpus
- NSP token — [CLS]
- Token for separating sentence — [SEP]

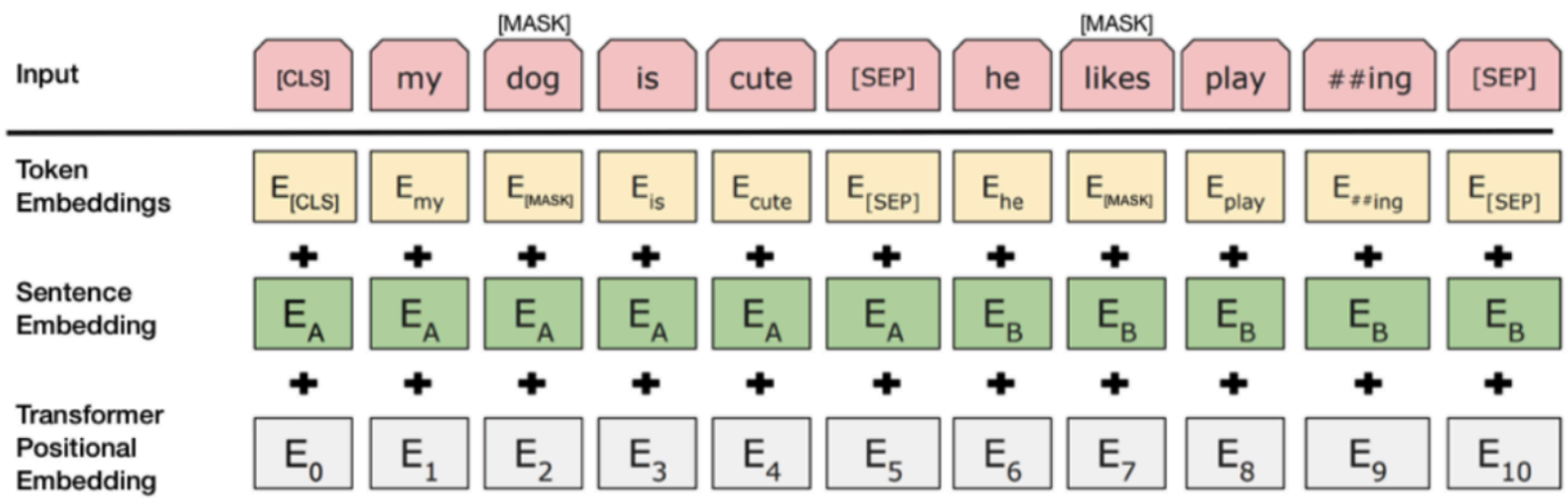


Next Sentence Prediction

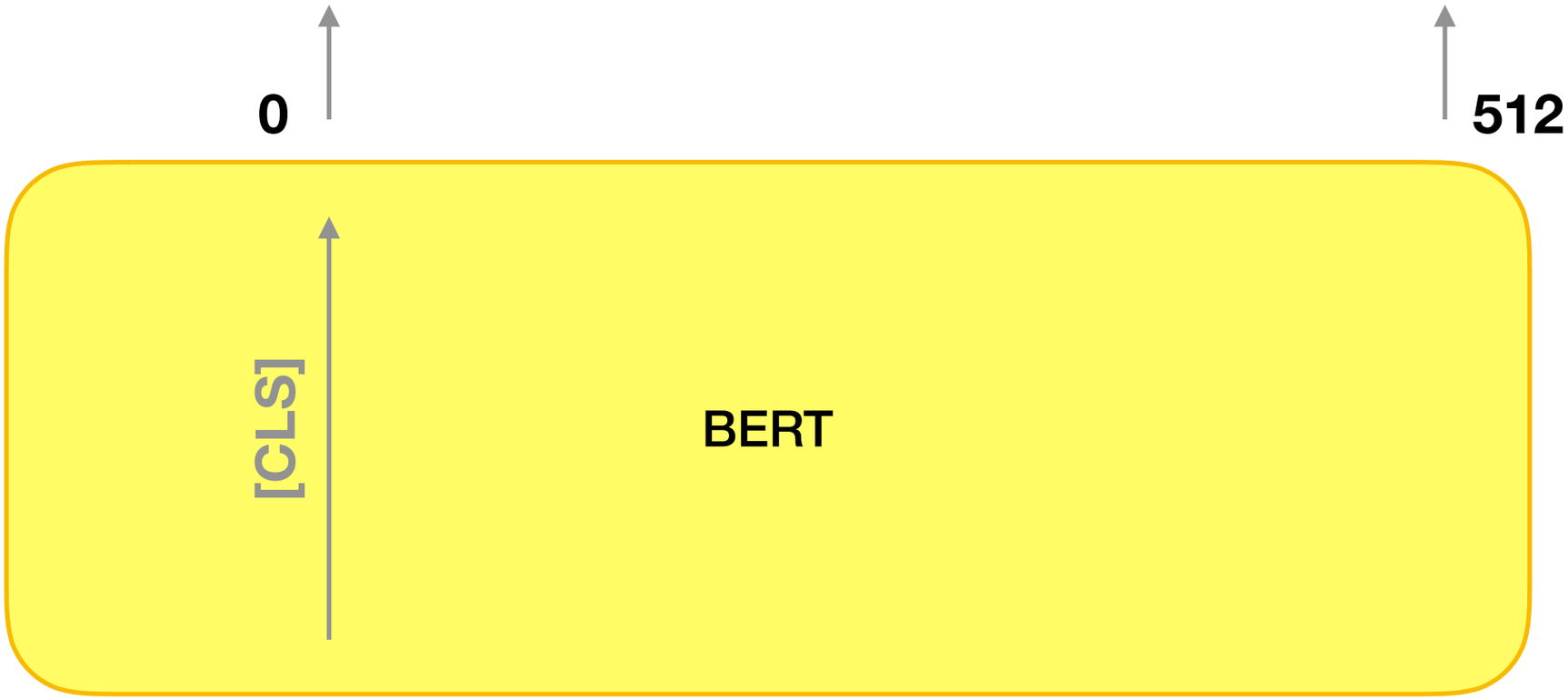


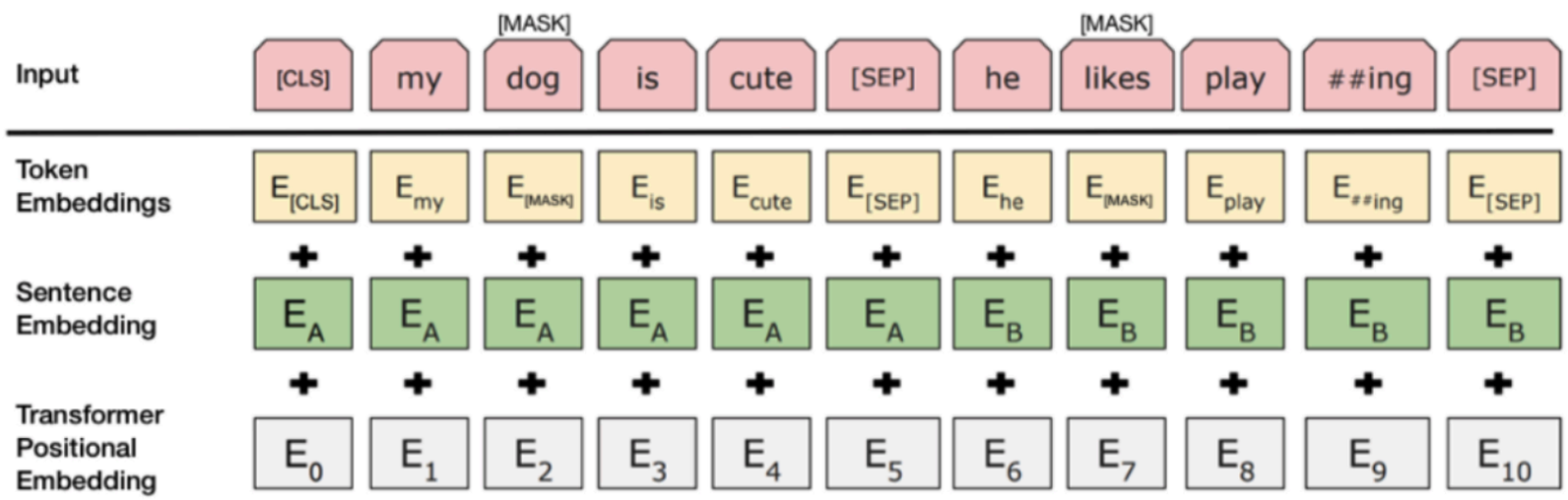
Next Sentence Prediction



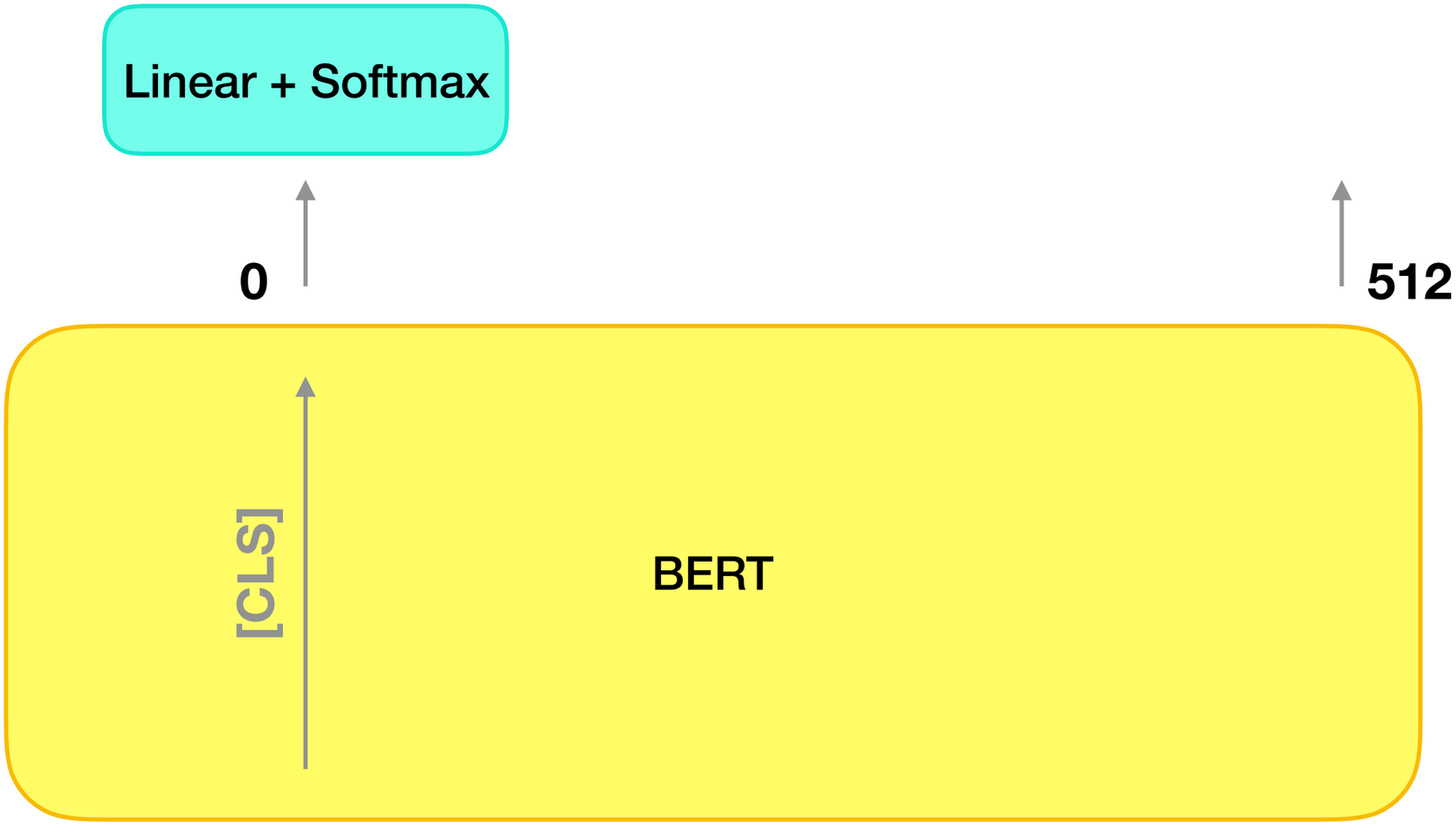


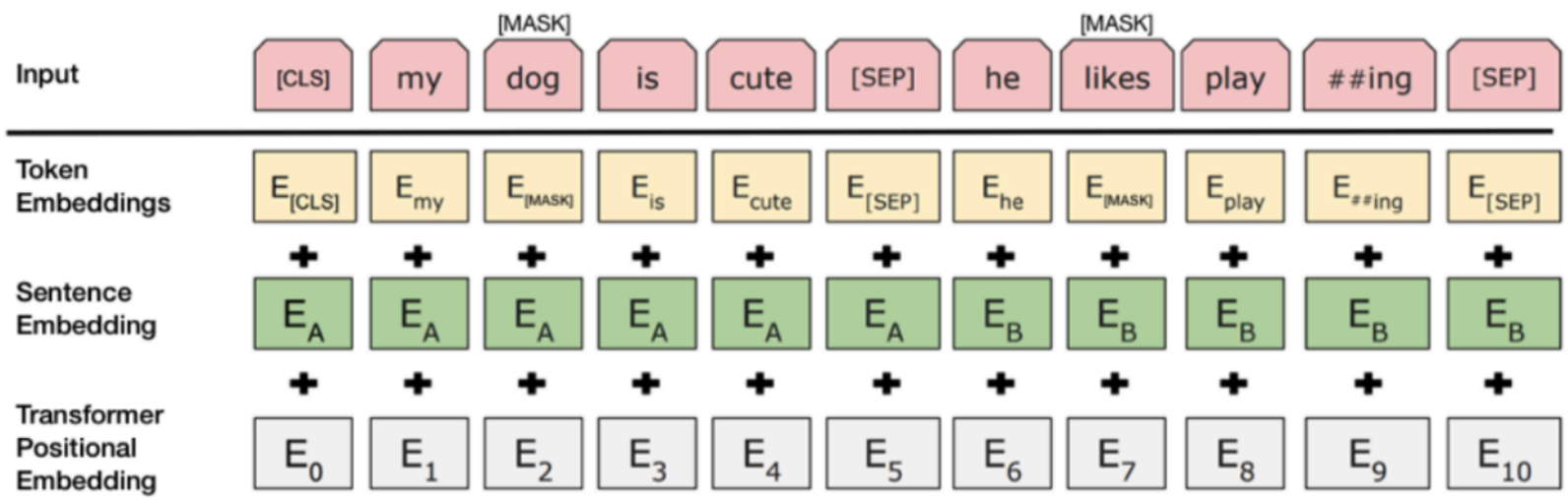
Next Sentence Prediction



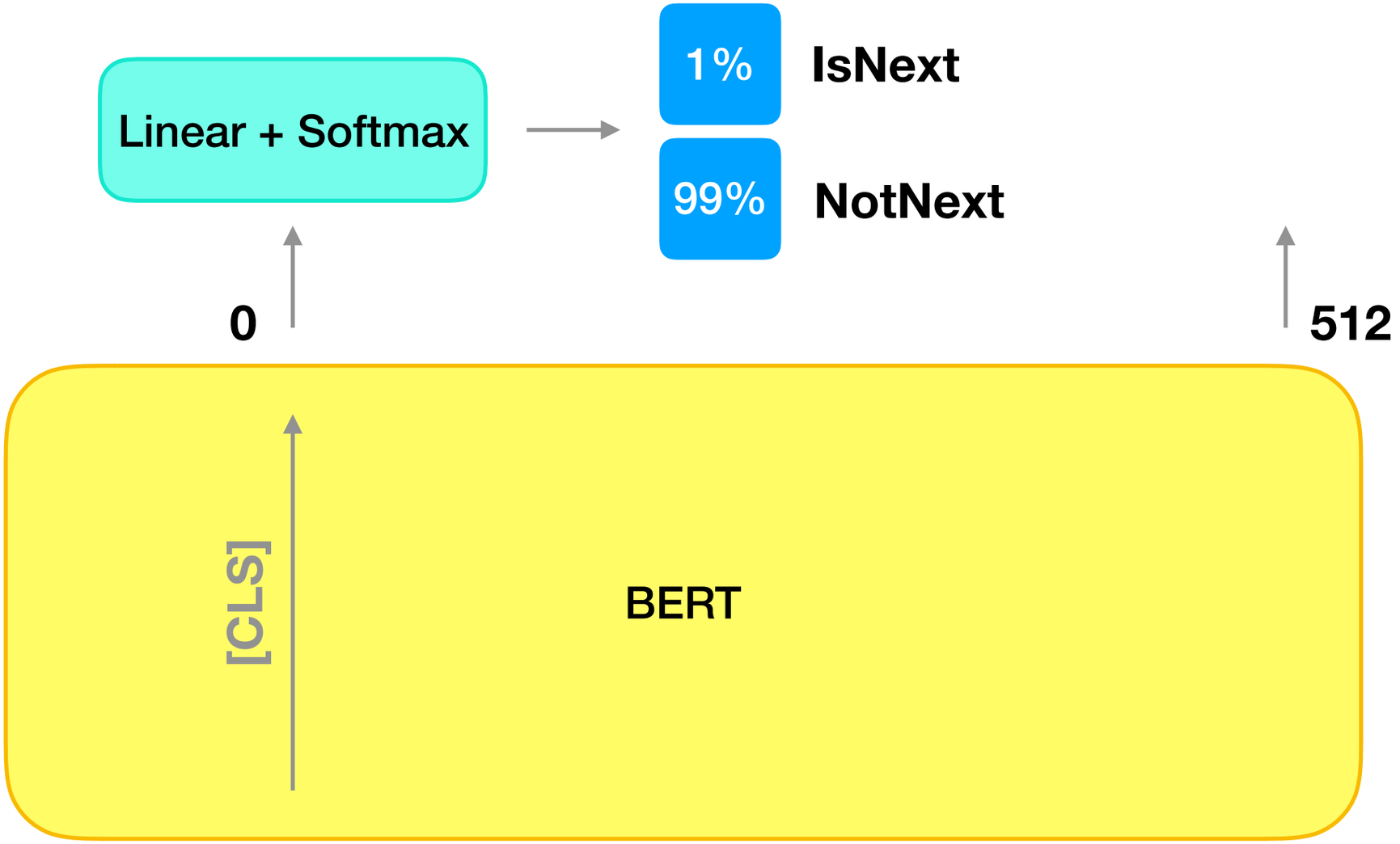


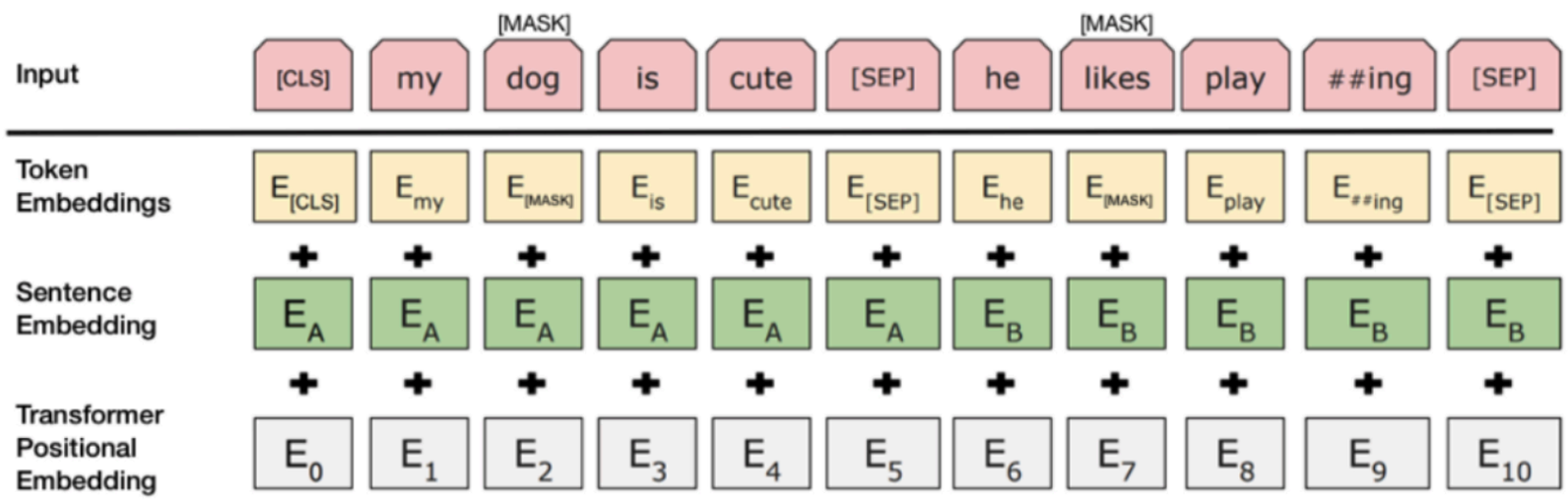
Next Sentence Prediction





Next Sentence Prediction





BERT Summary

- New data structure
- Weak training signal (masked 15% of tokens)
- Because of the weak signal is trained much longer
- Have auxiliary task
- But maybe it not necessary
- Because there are results that do not show improvements using NSP

RoBERTa

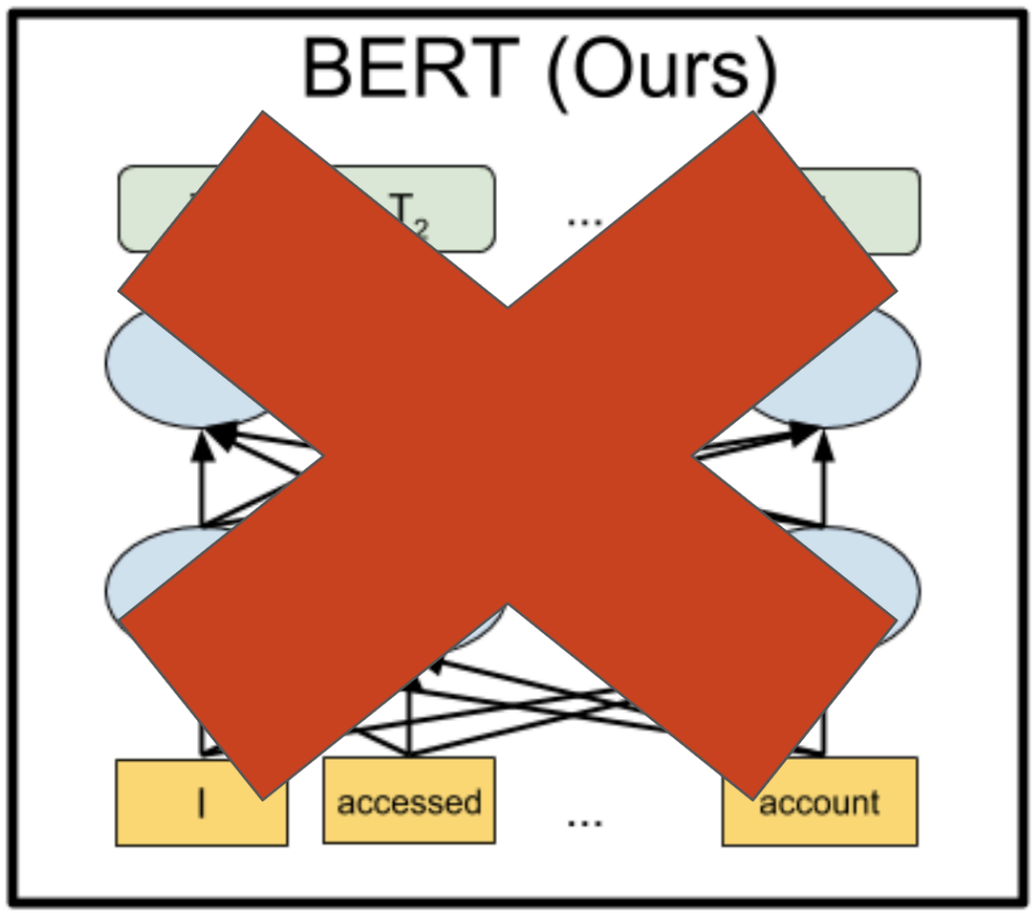
- More data, bigger batches, longer training
- Removing NSP
- Training on longer sequences
- Dynamically changing the masking pattern applied to the training data

ALBERT

- Cross-layer parameter sharing
- Factorized embedding parameterization
- Sentence-order prediction

ALBERT

Cross-layer parameter sharing



- Do you use **12** transformer layers?
- Better! We use **one** transformer layer and apply it 12 times!

ALBERT

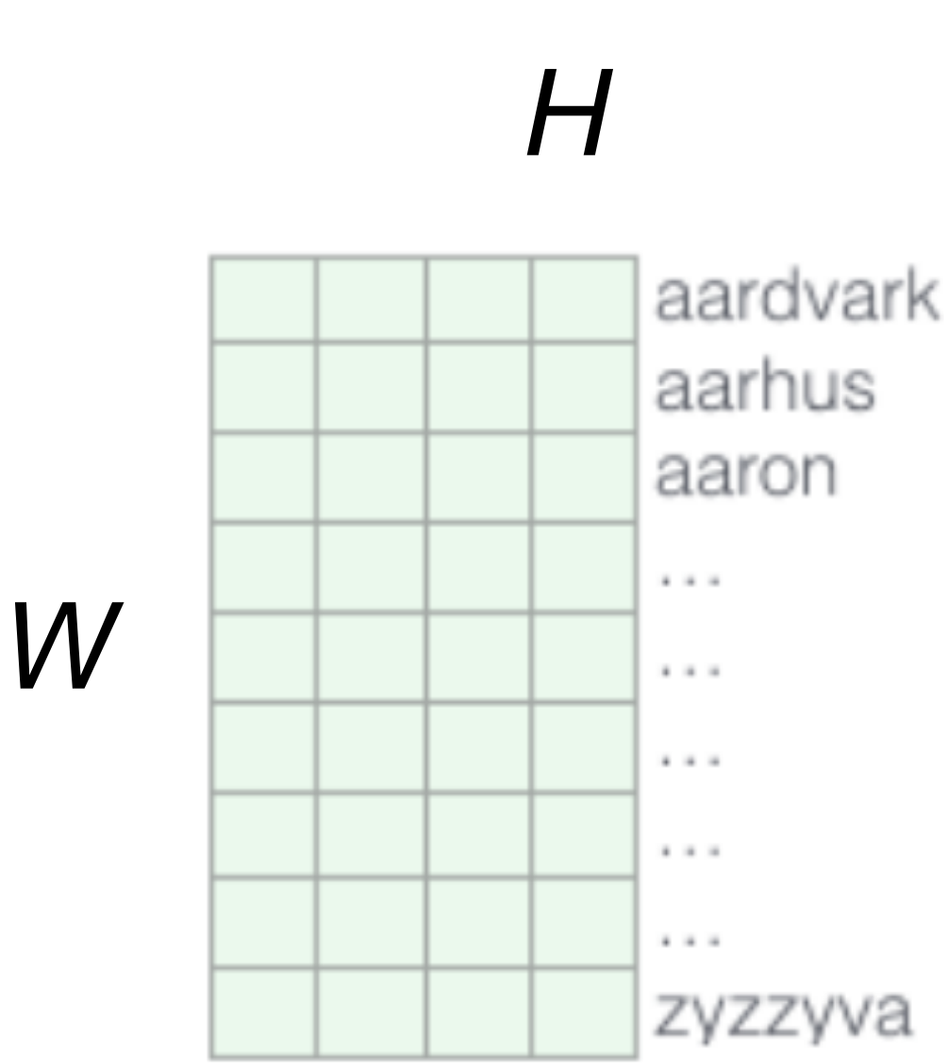
Model		Parameters	Layers	Hidden	Embedding	Parameter-sharing
BERT	base	108M	12	768	768	False
	large	334M	24	1024	1024	False
	xlarge	1270M	24	2048	2048	False
ALBERT	base	12M	12	768	128	True
	large	18M	24	1024	128	True
	xlarge	59M	24	2048	128	True
	xxlarge	233M	12	4096	128	True

Table 2: The configurations of the main BERT and ALBERT models analyzed in this paper.

Model		Parameters	SQuAD1.1	SQuAD2.0	MNLI	SST-2	RACE	Avg	Speedup
BERT	base	108M	90.5/83.3	80.3/77.3	84.1	91.7	68.3	82.1	17.7x
	large	334M	92.4/85.8	83.9/80.8	85.8	92.2	73.8	85.1	3.8x
	xlarge	1270M	86.3/77.9	73.8/70.5	80.5	87.8	39.7	76.7	1.0
ALBERT	base	12M	89.3/82.1	79.1/76.1	81.9	89.4	63.5	80.1	21.1x
	large	18M	90.9/84.1	82.1/79.0	83.8	90.6	68.4	82.4	6.5x
	xlarge	59M	93.0/86.5	85.9/83.1	85.4	91.9	73.9	85.5	2.4x
	xxlarge	233M	**94.1/88.3**	**88.1/85.1**	**88.0**	**95.2**	**82.3**	**88.7**	1.2x

ALBERT

Factorized embedding parameterization

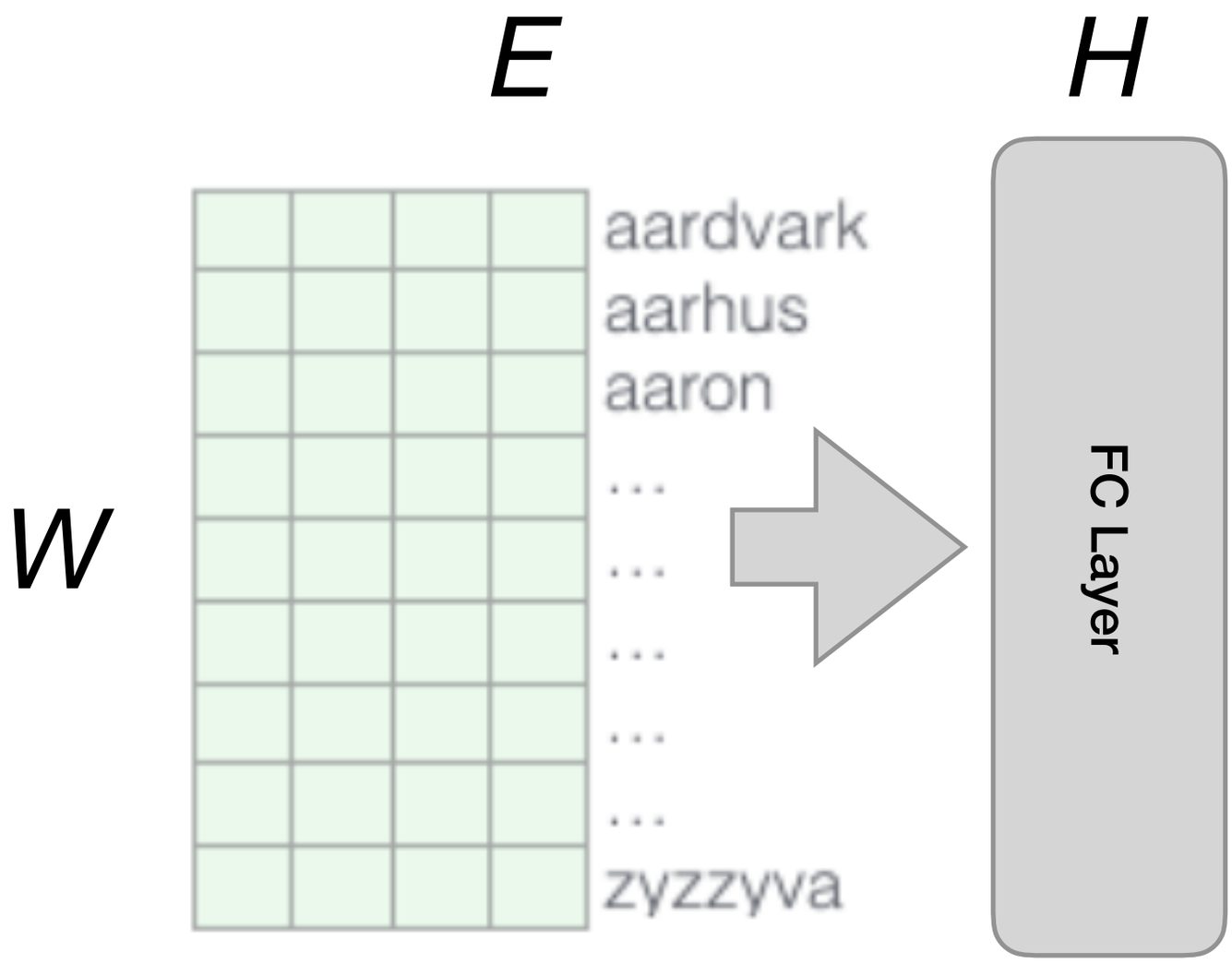


Memory complexity:
O(*W* x *H*)

W = 30000
H = 2048
W x *H* ≃ **61M**

ALBERT

Factorized embedding parameterization



Memory complexity:
O(*W* x *E* + *E* x *H*)

W = 30000
E = 256
H = 2048
W x *E*+*E* x *H* ≃ **8M**

ALBERT

Sentence-order prediction

Thanks for your Attention!

Boris Zubarev

@bobazooba