COMMUJI

Suggest the right emoji for the right message using BERT

PROBLEM

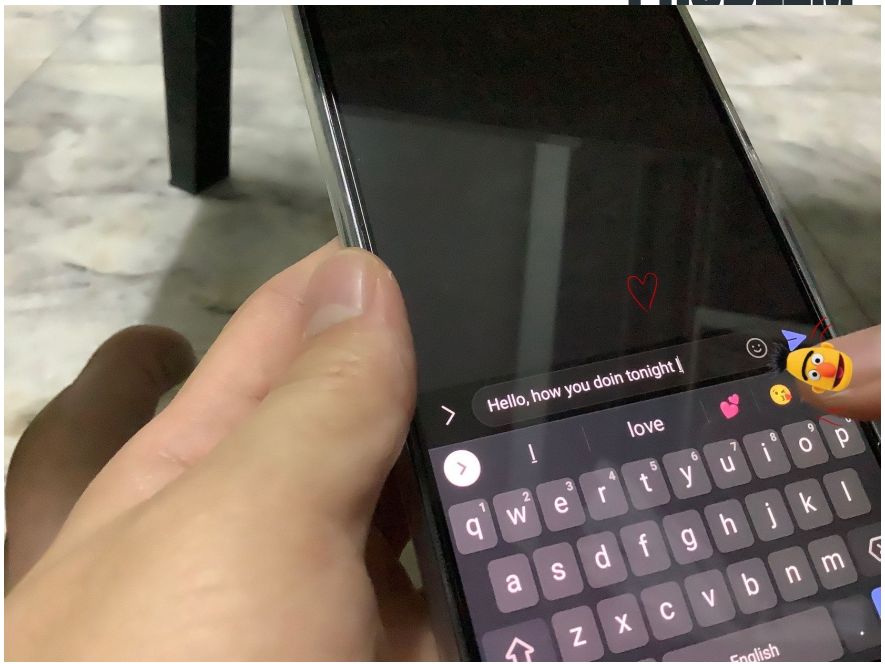


TABLE OF CONTENTS

01.

DATA PREPROCESSING

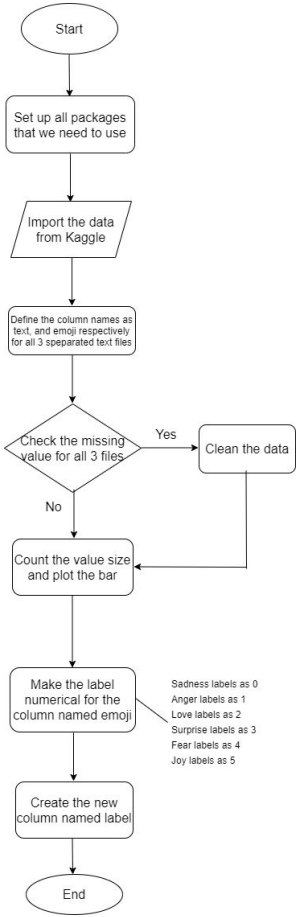


Figure 1 : Data Preprocessing Flowchart

** Limitation: only support English language **

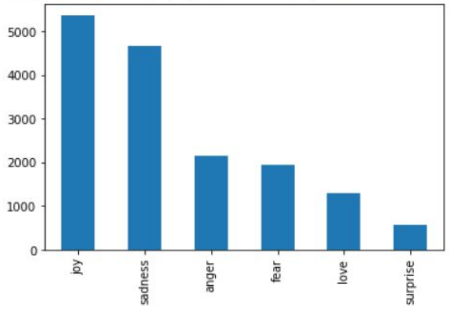
	free text	emoji
1	I feel calm just thinking about it	😂
2	I wish I could re estabLish a reasonable..	😭
3	I wish he could be destroyed 1 day	😠

Dimension of whole data : (20000 rows, 2 columns)

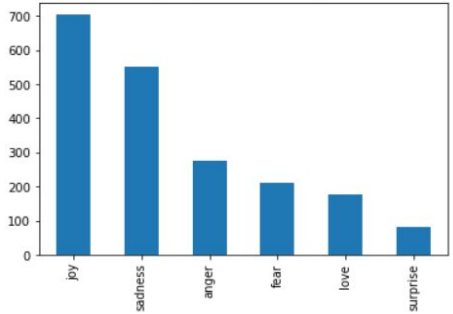
Output : { ‘joy’ : 😂 , ‘sadness’:😭, ‘anger’:😠, ’love’ : ❤️, ‘surprise’ : 😲, ‘fear’ : 😨 }

```
{'anger': 1, 'fear': 4, 'joy': 5, 'love': 2, 'sadness': 0, 'surprise': 3}
```

Training dataset (16k)



Validation dataset (2k)



Test dataset (2k)

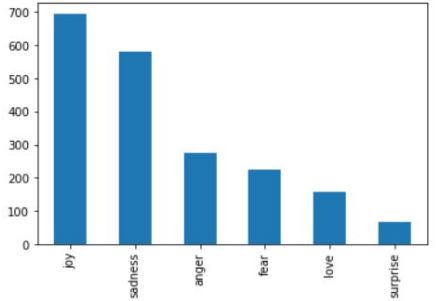


Figure 2 : Training, Validation, Test set count by label

	free text	Label
1	I feel calm just thinking about it	5
2	I wish I could re estabLish a reasonable..	0
3	I wish he could be destroyed 1 day	1

Dimension of whole data : (20000 rows, 2 columns)

```
{'anger': 1, 'fear': 4, 'joy': 5, 'love': 2, 'sadness': 0, 'surprise': 3}
```

02.

HOW THE MODEL WORKS (BERT)

HUGGING FACE

Figure 3 : open-source & platform provider for BERT pretrained model

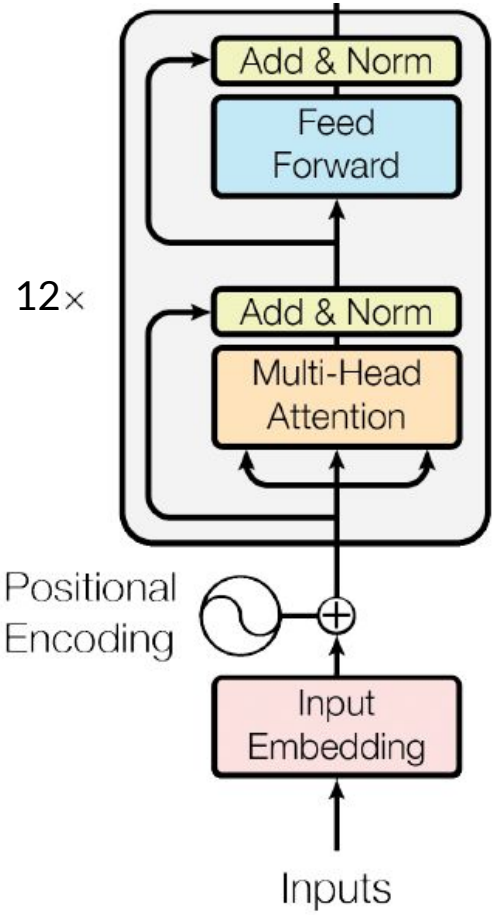
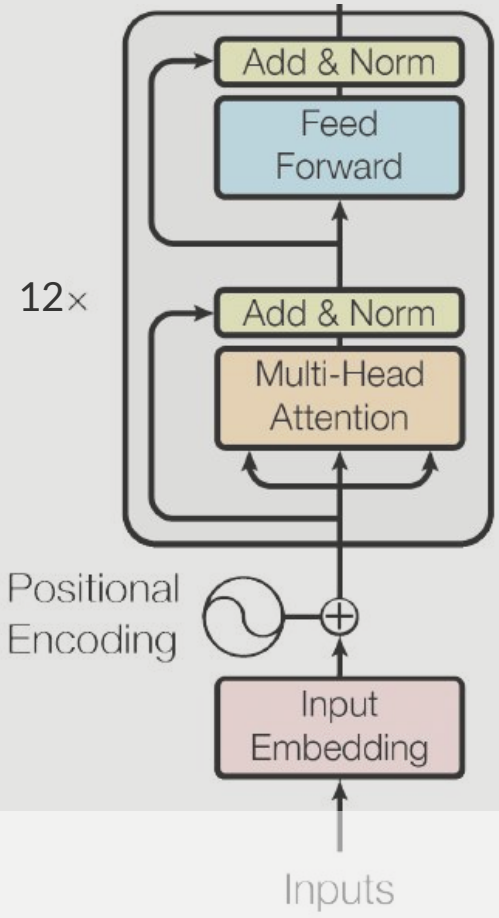


Figure 4 : BERT-based uncased

🤗 BERT-based uncased🤗



“I feel romance too”

[CLS] I feel romance too [SEP]

{[CLS]: 101,I: 1045,feel: 2514,

romance: 6298, too: 2205, [SEP]:102 }

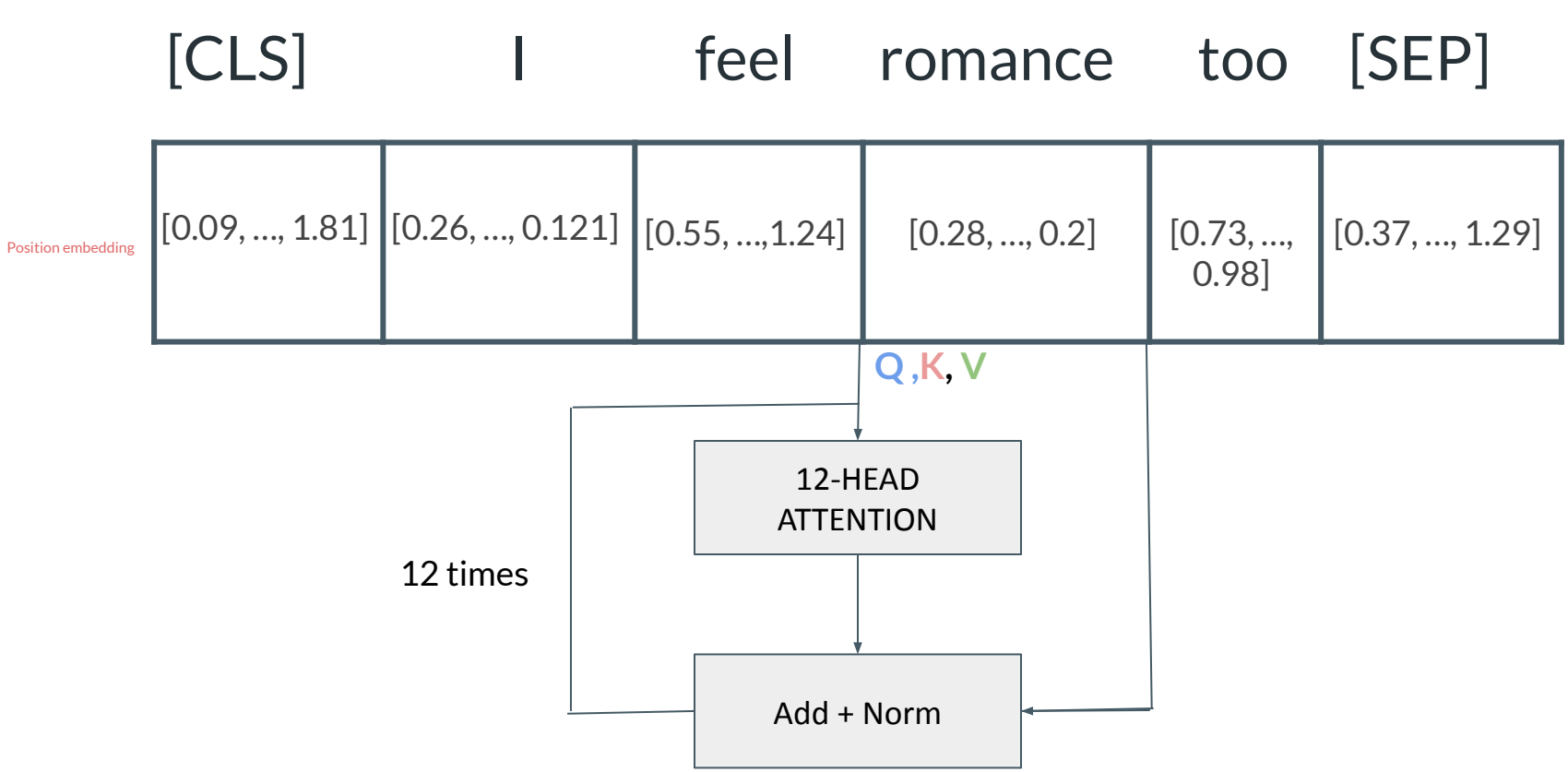
[101, 1045, 2514, 6298, 2205, 102]

[CLS] I feel romance too [SEP]

	[CLS]	I	FEEL	ROMANCE	TOO	[SEP]
INPUT TOKEN	101	1045	2514	6298	2205	102

[CLS] I feel romance too [SEP]

	[CLS]	I	FEEL	ROMANCE	TOO	[SEP]
INPUT EMBEDDING (SIZE = 768*1]	[0.01, ..., 0.91]	[0.24, ..., 0.12]	[0.5, ..., 0.34]	[0.1, ..., 0.1]	[0.7, ..., 0.78]	[0.35, ..., 0.29]
POSITION ENCODING [SIZE = 768*1]	[0.08,..,0.9]	[0.02,..,0.001]	[0.05,..,0.9]	[0.18,..,0.1]	[0.03,..,0.2]	[0.02,..,1]


[CLS]
I
feel
romance
too
[SEP]
Position embedding
[0.09, ..., 1.81]
[0.26, ..., 0.121]
[0.55, ...,1.24]
[0.28, ..., 0.2]
[0.73, ..., 0.98]
[0.37, ..., 1.29]
Q ,K, V
12-HEAD ATTENTION
12 times
Add + Norm

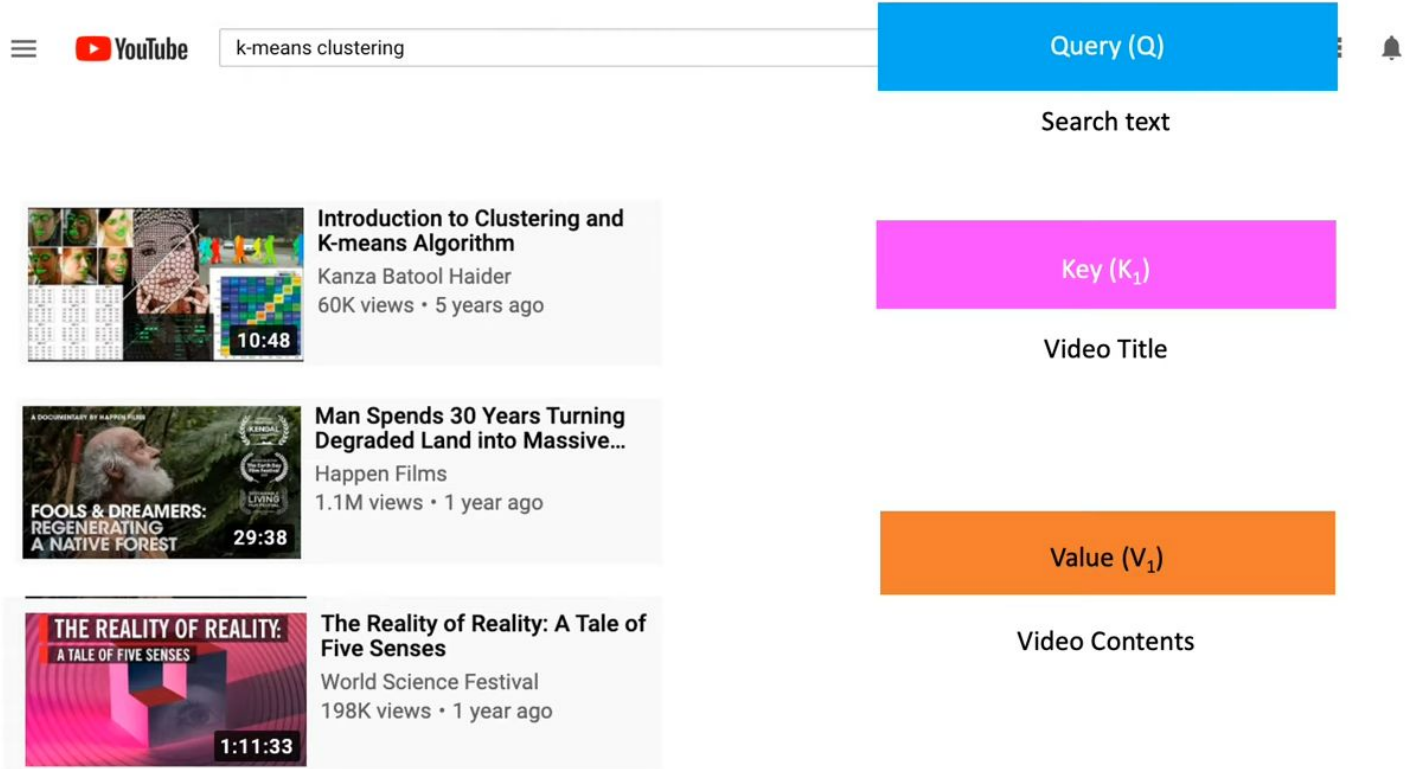


Figure 5 : Query vs Key vs Value Analogy

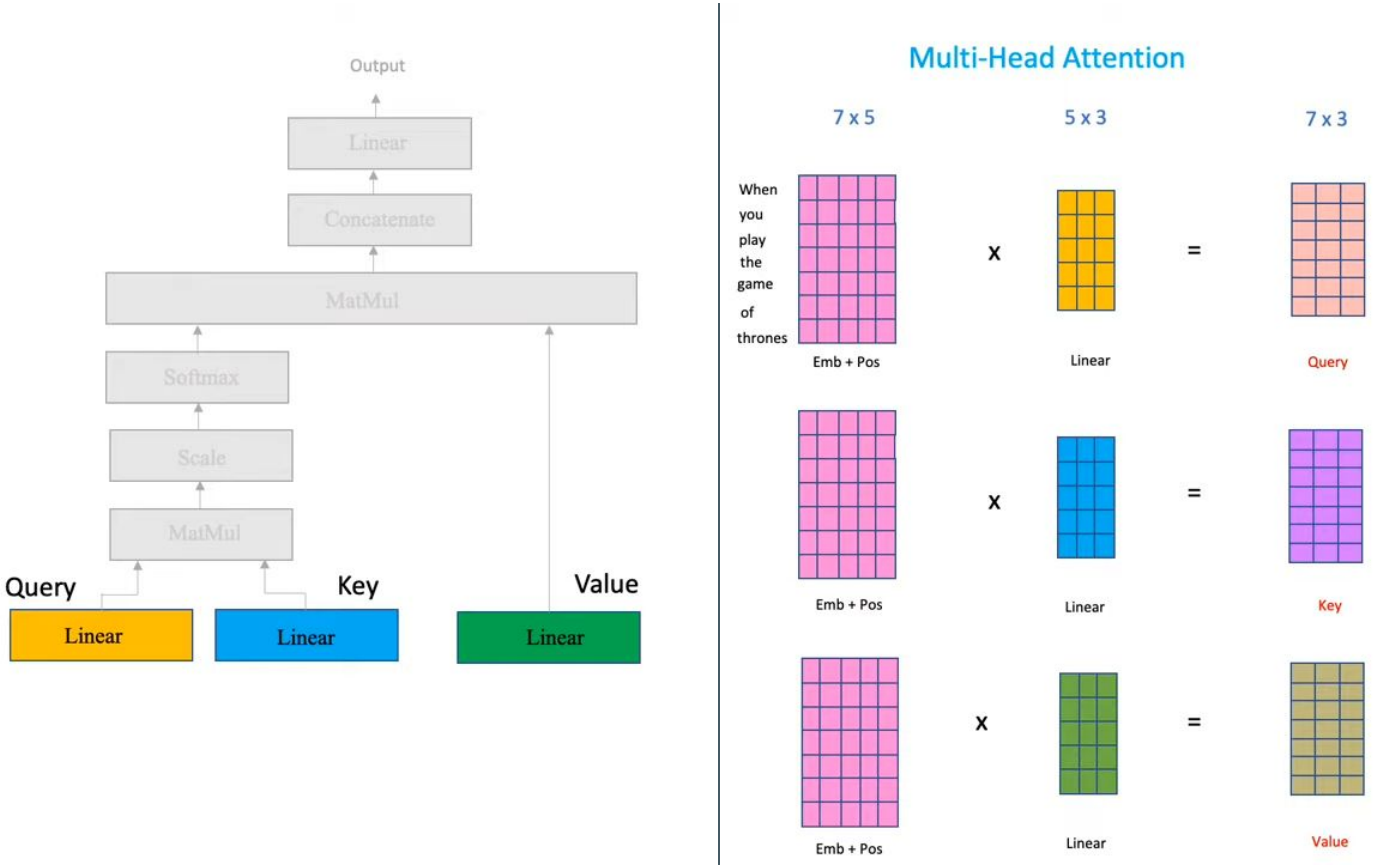


Figure 6: How Query vs Key vs Value were obtained

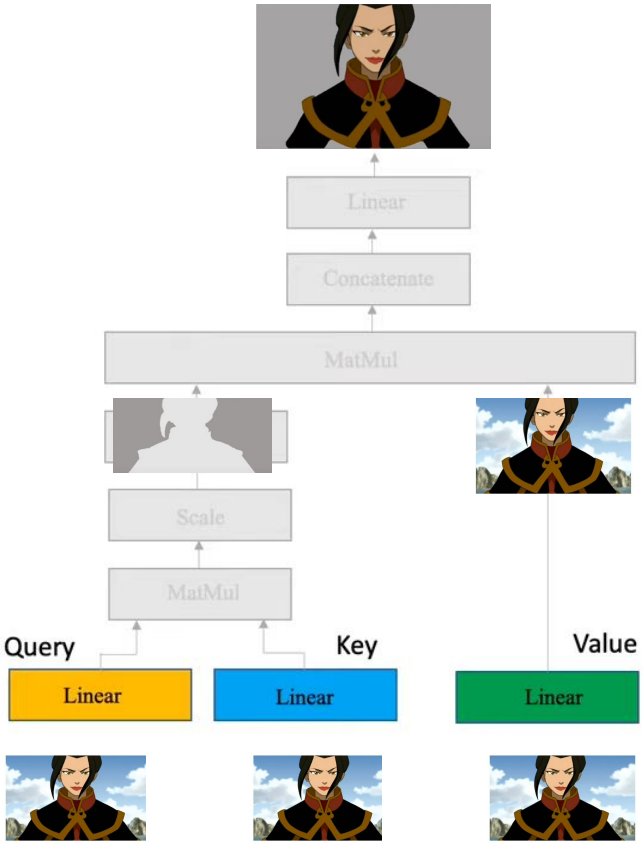
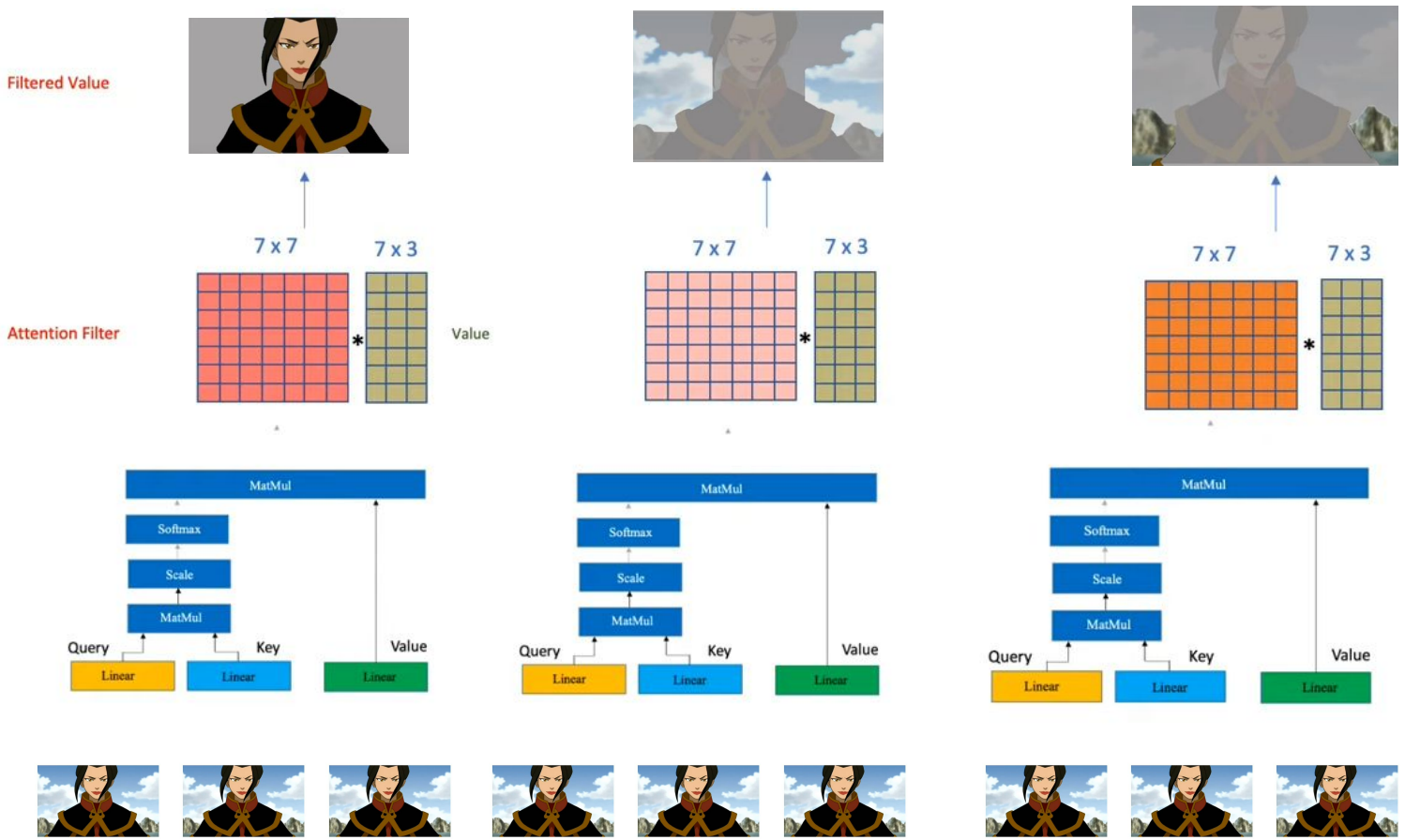
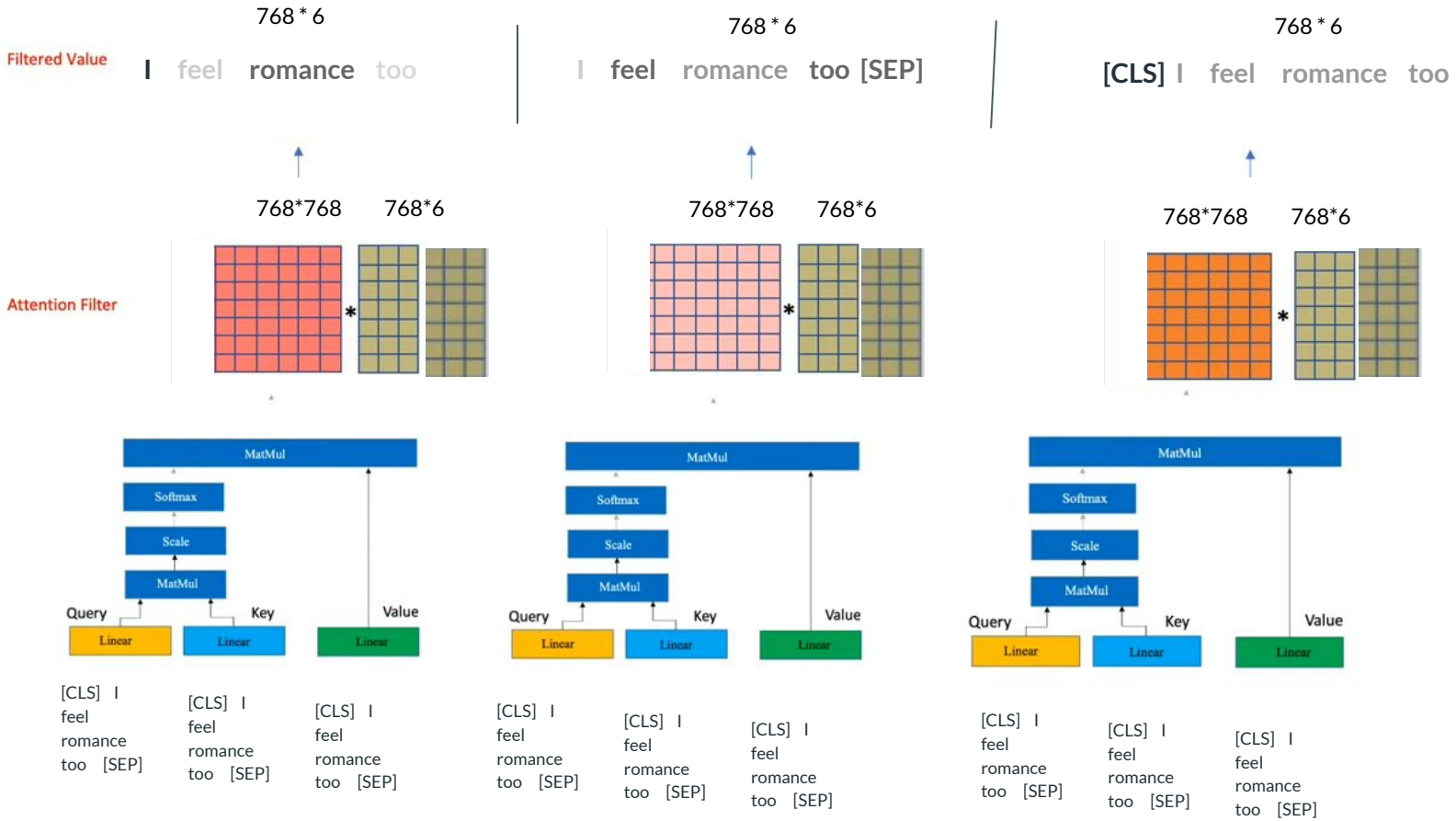


Figure 7 : Multi-head attention

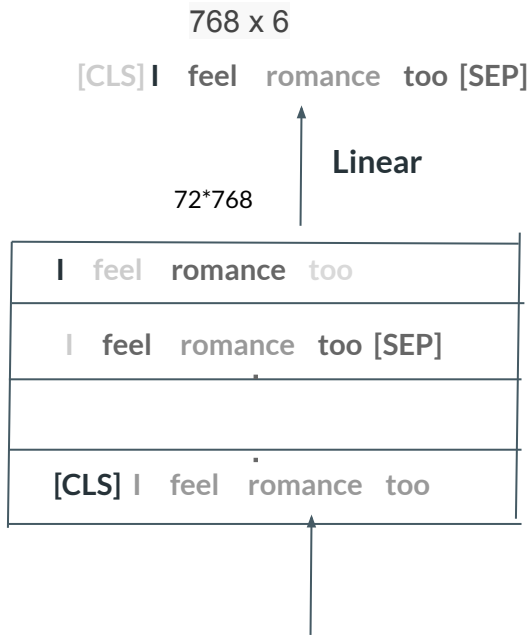

Filtered Value
7 x 7
7 x 3
Attention Filter
Value
MatMul
Softmax
Scale
MatMul
Query
Key
Value
Linear
Linear
Linear
Attention head 1
Attention head 2
Attention head 3

Multi-head attention



Note: The scale of the matrix might not be proportional equal

Multi-head attention



Output (of Multi head attention)

Concat Filtered value

768 x 6
I feel romance too

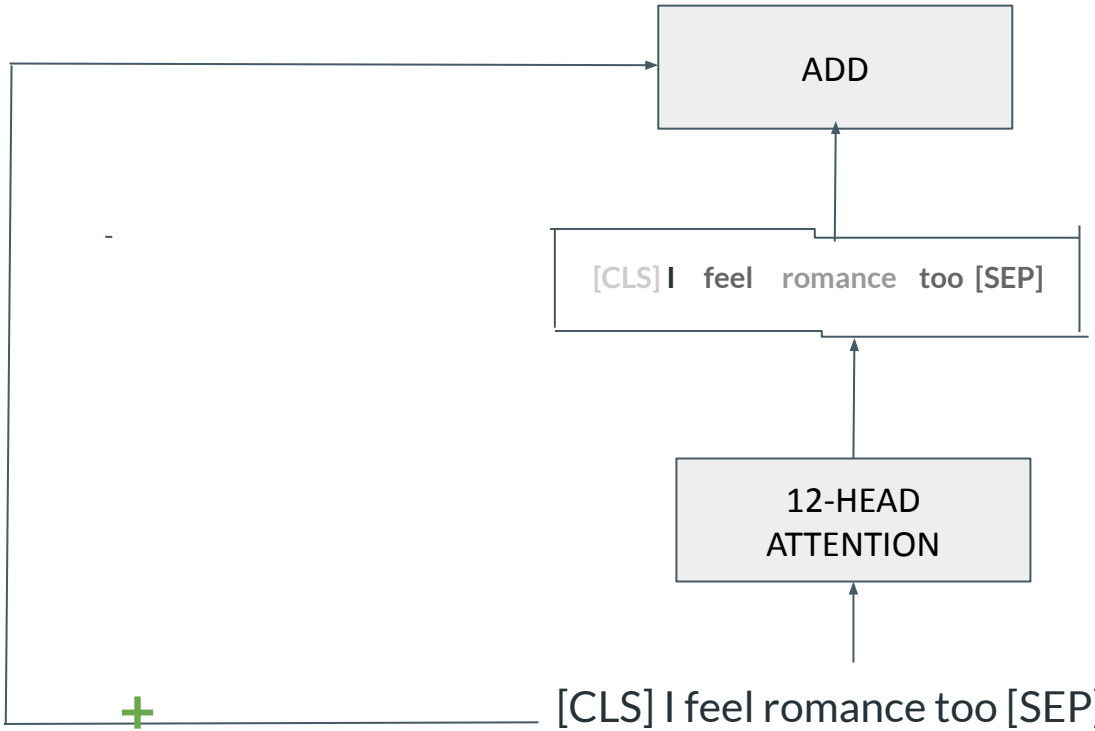
768 x 6
I feel romance too [SEP]

....

768 x 6
[CLS] I feel romance too

Filtered value

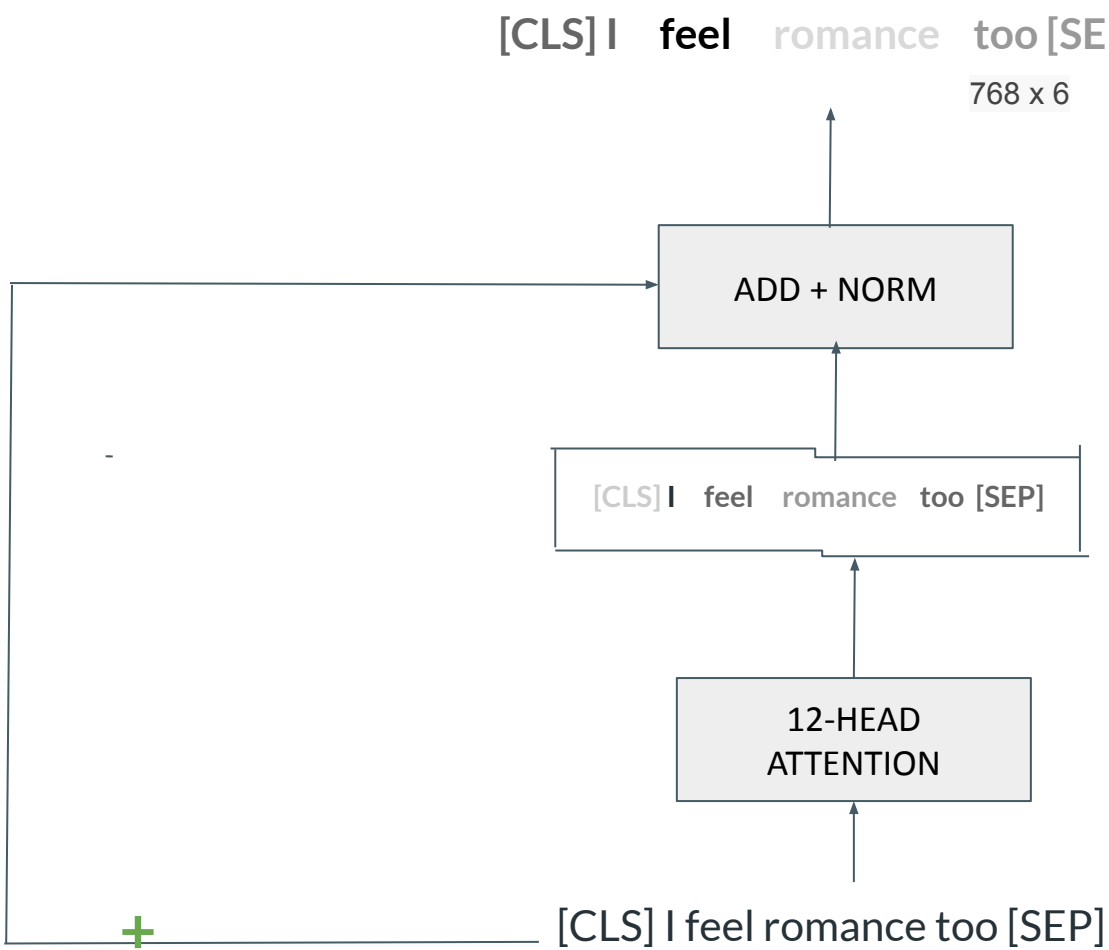
Attention head 1 Attention head 2 ... Attention head 12



= Position embedding + Filtered value

Output (of Multi head attention)

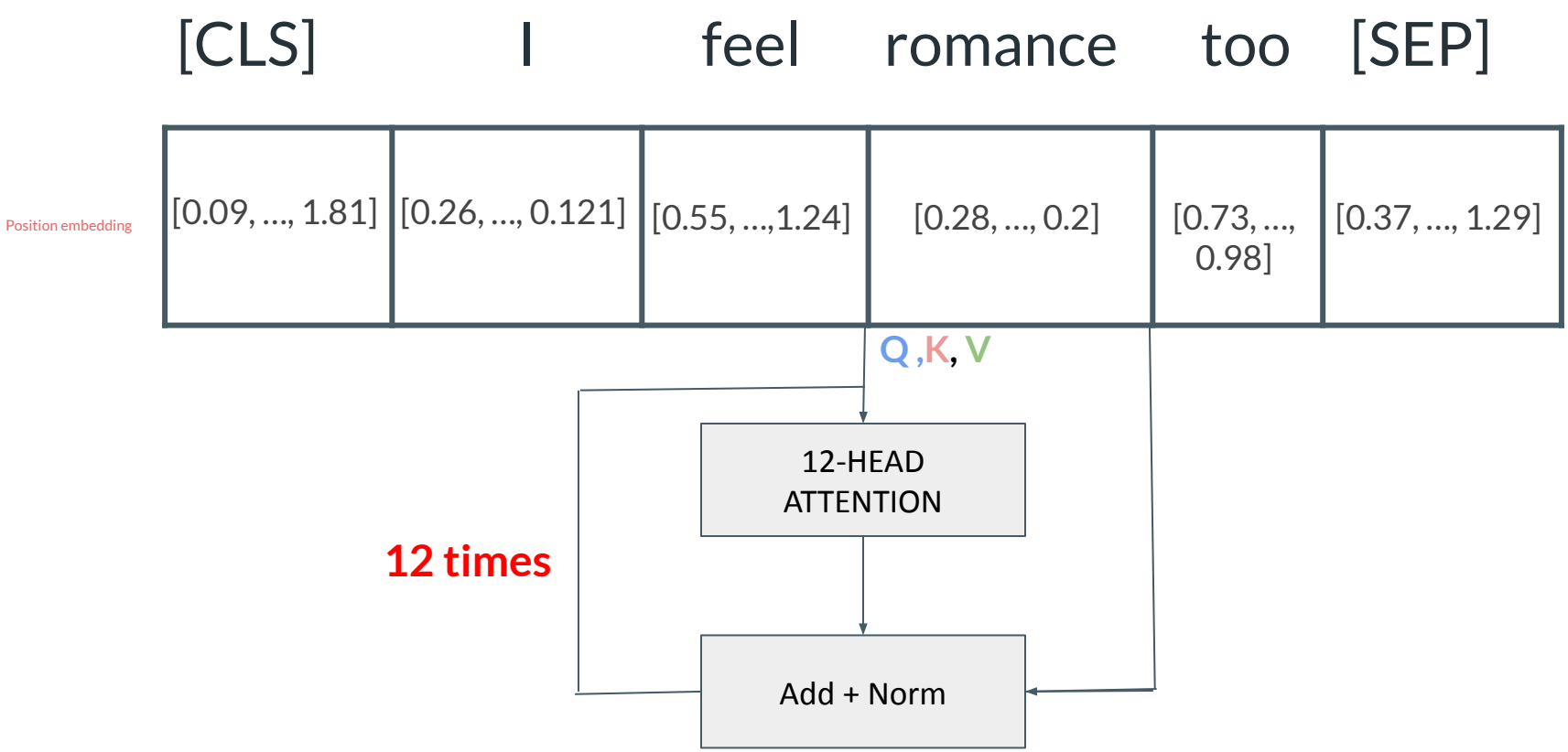
Position embedding



= LayerNormalize(ADD)

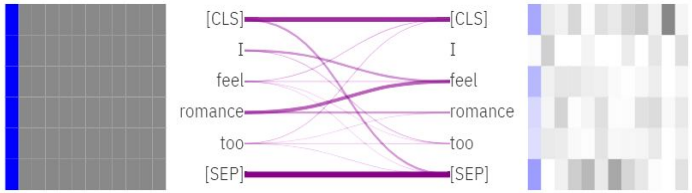
Output (of Multi head attention)

Position embedding

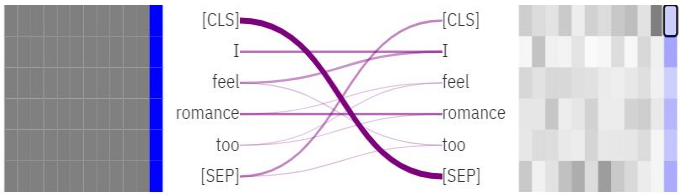


NOW REPEAT IT FOR 12 TIMES ; SINCE THERE ARE 12 ATTENTION LAYERS

MULTI HEAD ATTENTION WEIGHT (HEAD, LAYER)



LAYER 1 HEAD 1



LAYER 1 HEAD 12

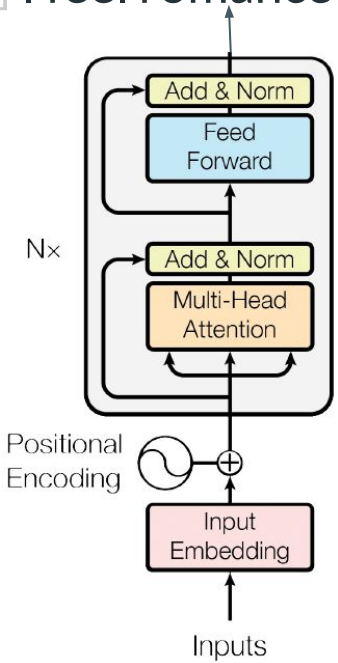
LAYER 12 HEAD 1

LAYER 12 HEAD 12

RECAP

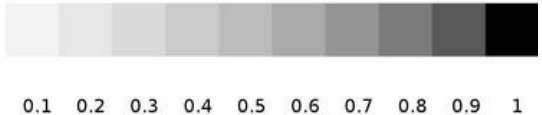
768 x 6

[CLS] I feel romance too [SEP]



768 x 6

“[CLS] I feel romance too [SEP]”



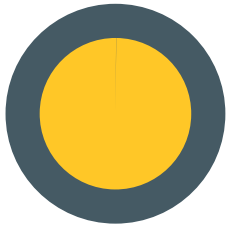
03.

TRAINING THE MODEL (BERT)

TWO PHASES OF TRAINING BERT

1.

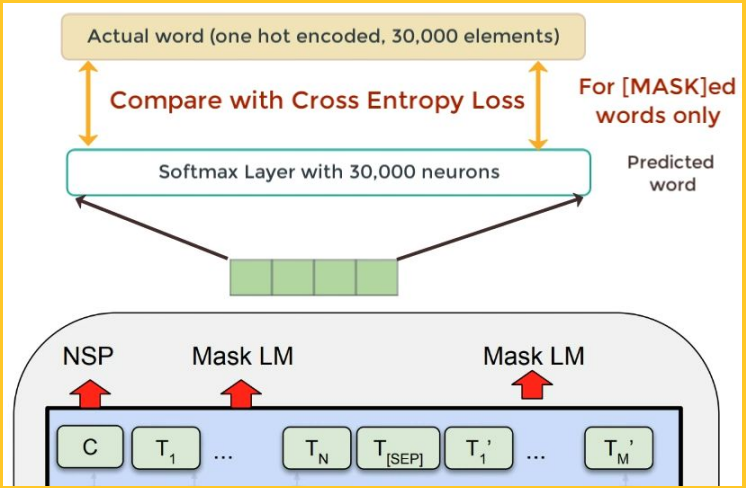
Pre-Training phase

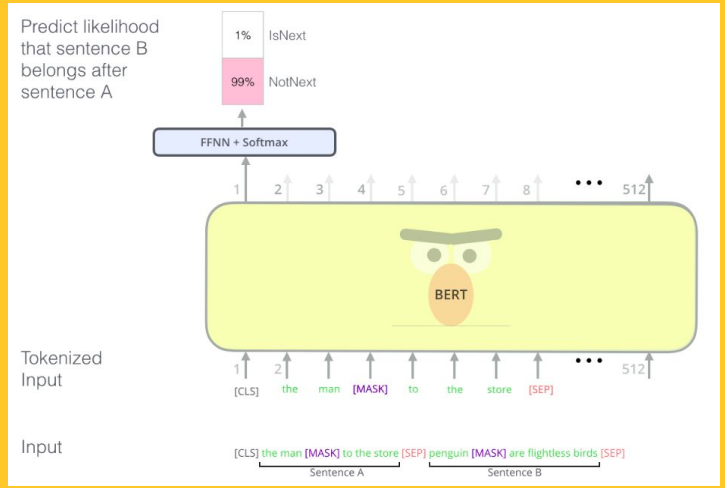


2.

Fine-Tuning phase

Phase 1 : Pre-Training



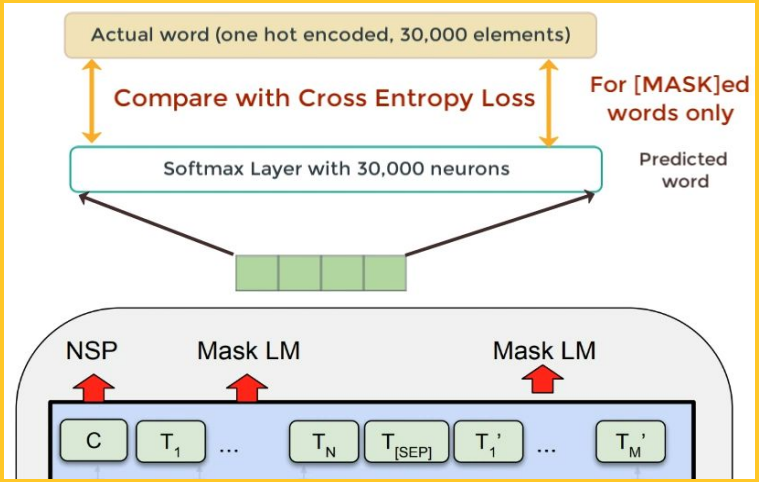


1. Loss function: (MLM)

2. Loss function: (NSP)

- Let the model familiar with the words first -

Phase 1 : Pre-Training



Loss function: (MLM)

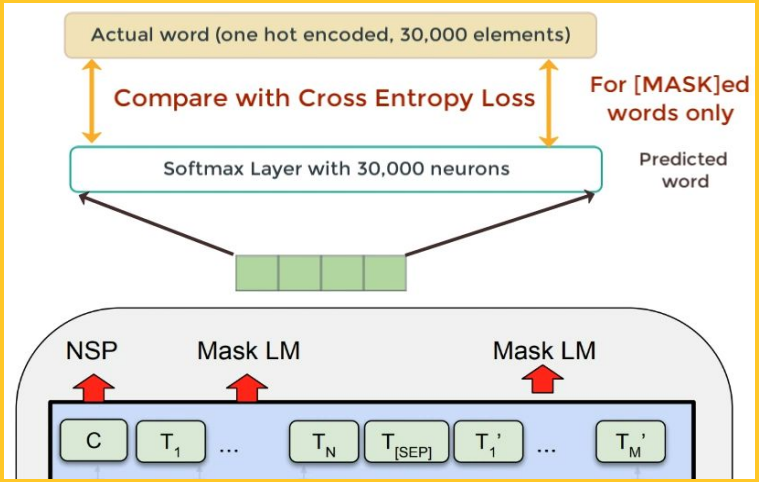
[CLS] I feel romance too [SEP]

Masked LM

[CLS] I feel [MASK] too [SEP]

Step 1 : Randomly mask a word [15%]

Phase 1 : Pre-Training



Loss function: (MLM)

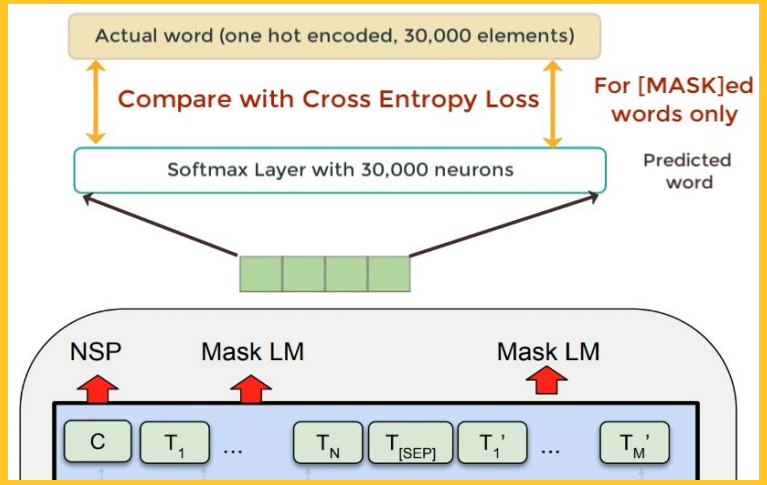
[CLS] I feel [MASK] too [SEP]

Predict

[CLS] I feel [bad] too [SEP]

Step 2 : Predict the word in [MASK]

Phase 1 : Pre-Training



Loss function: (MLM)

[CLS] I feel [bad] too [SEP]

[0.01, …, 0.42, …..0.12]

LOSS

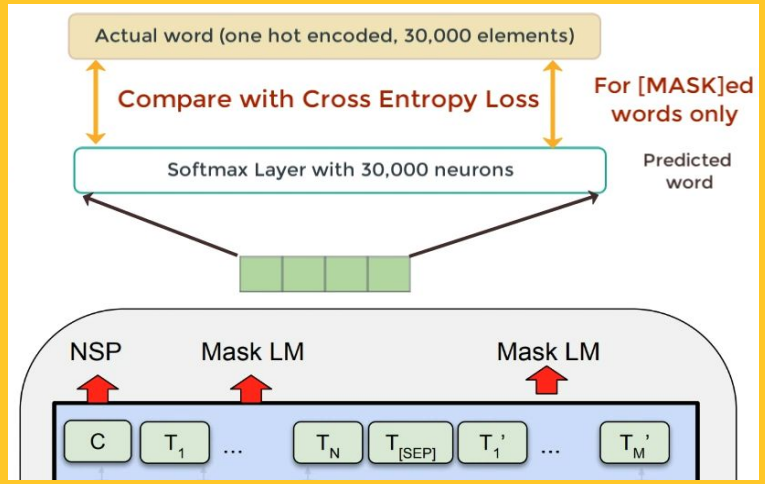
- log(0.42) = 0.37

[0, …, 1, …..0]

[CLS] I feel [romance] too [SEP]

Step 3 : Calculate loss function

Phase 1 : Pre-Training



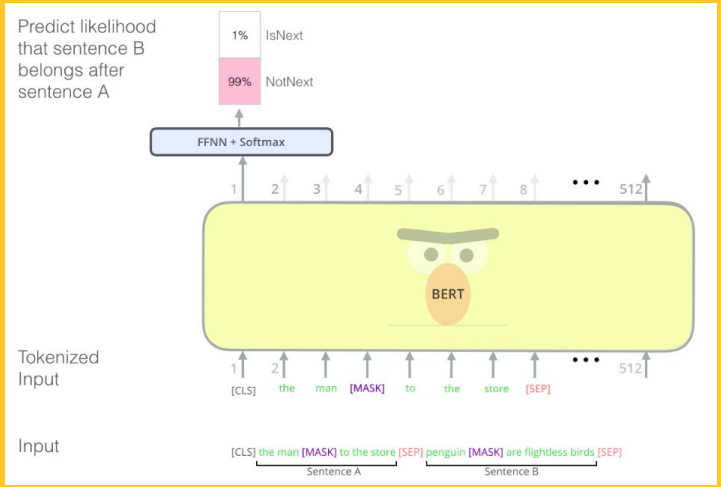
Loss function: (MLM)

Loss : 0.37

Back Propagation

Step 4 : Update parameter

Phase 1 : Pre-Training



Loss function: (NSP)

Sentence A :[CLS] I feel romance too [SEP],

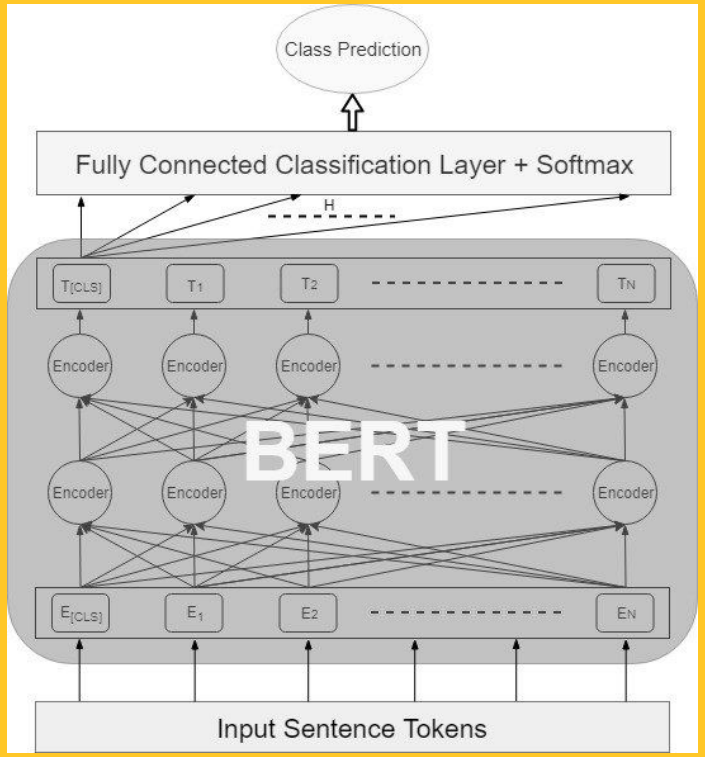
Sentence B :[CLS] My feeling ? [SEP]

Next Sentence Prediction

0 :(Sentence A followed by Sentence B)

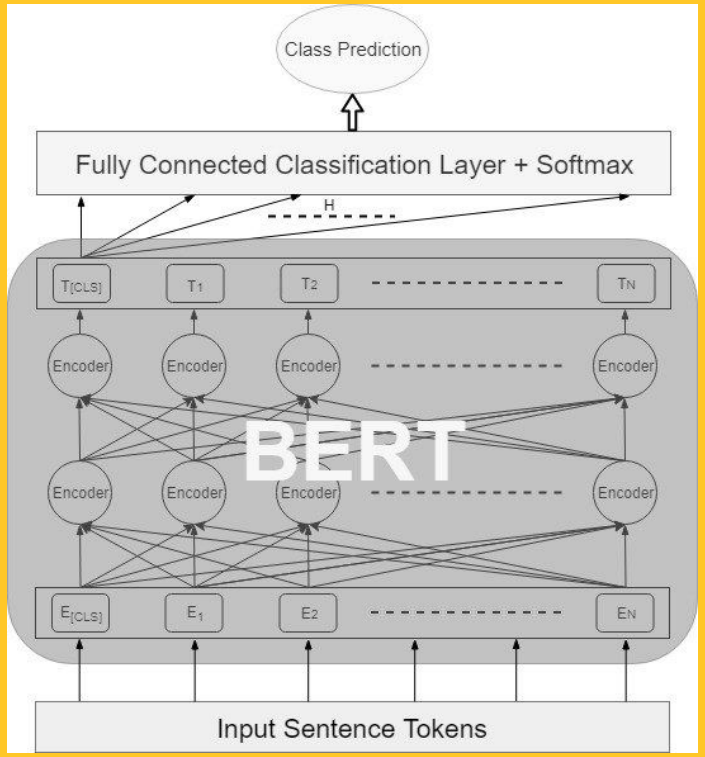
1 :(Sentence B followed by Sentence A)

Phase 2 : Fine-Tuning



- **Just plug fully connected layer + Softmax**

Phase 2 : Fine-Tuning



Vocabulary size : 30k

[CLS] I feel romance too [SEP]

768 x 6

Feed forward NN +

SoftMax 6 layers

Loss

[p(😂) ,p(😭), p(😠), p(❤️), p(😲),p(😰)]

1 x 6

04.

PERFORMANCE EVALUATION

PERFORMANCE EVALUATION

	😂	😭	😠	❤️	😲	😰
ACCURACY	559/581	245/275	130/159	62/66	196/224	656/695

Weighted F1 score on test set :
0.9247

PREDICTION EXAMPLE

Text	Predict
What if the creepy murderer is in your house	😨
Im sorry to hear it	😔
How you know what I want	😡
Fuck, Congratulation my lovely brother	❤️

MISCLASSIFICATION EXAMPLE

Text	Label	Predict
i know what it feels like he stressed glaring down at her as she squeezed more soap onto her sponge	😡	😔
i feel very strange today	'😨'	'😮'
ive this bad feeling that im being hated	'😔'	'😡'
i feel assaulted by all directions	'😔'	'😨'

05.

CONCLUSION