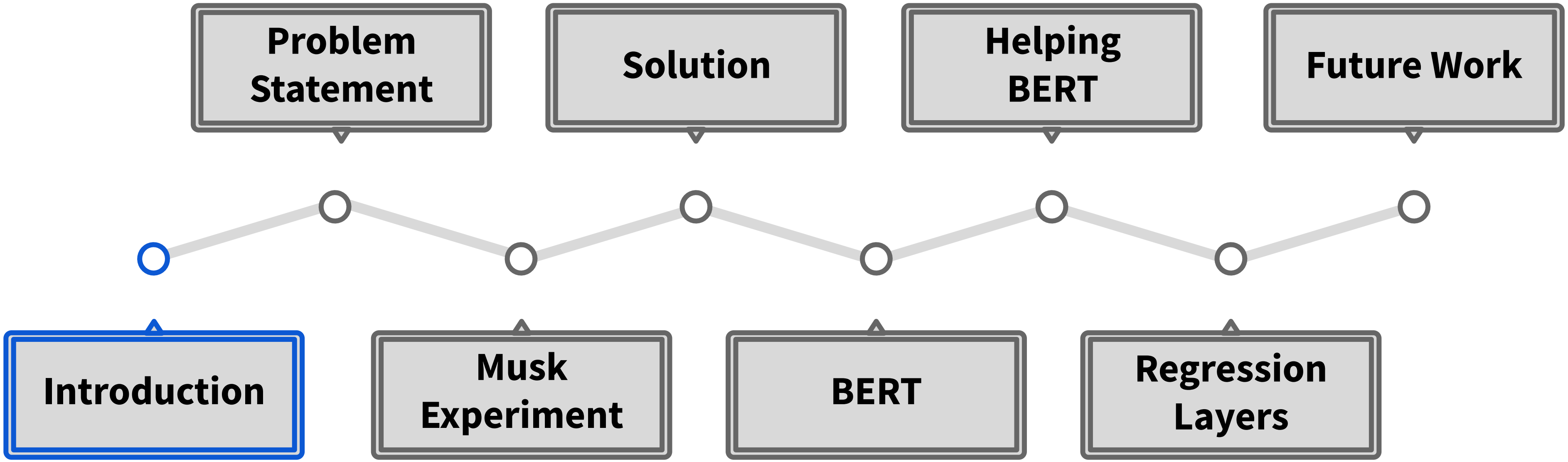
From a “lambo” to a Price Increase: Defining a Twitter Score for Financial Sentiment

Andrea Cicchini, Alexandru Ionut Pascariu, Victor Plesco


Introduction
Problem Statement
Musk Experiment
Solution
BERT
Helping BERT
Regression Layers
Future Work

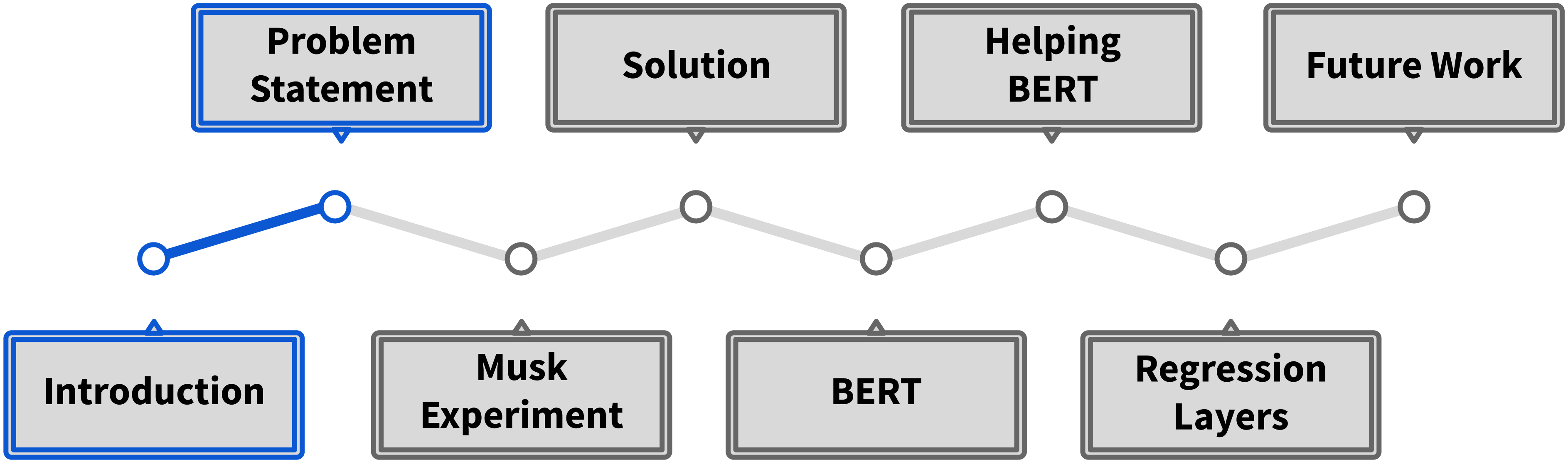
Why cryptocurrencies?

- Big Fluctuations in Price:
  - 1/25 - $0.0083
  - 4/19 - $0.4073 (+4807%)
- High Volatility due to Lack of Institutional Guarantor
  - “*wild west*”!
- Efficient Market Hypothesis (EMH)
  - Inefficient (N. A. Kyriazis, 2019)
- Social Media as Primary Source
  - Emotional Intelligence

🚀 🚀 🚀 🚀

Social Network Sentiment For Financial Prediction

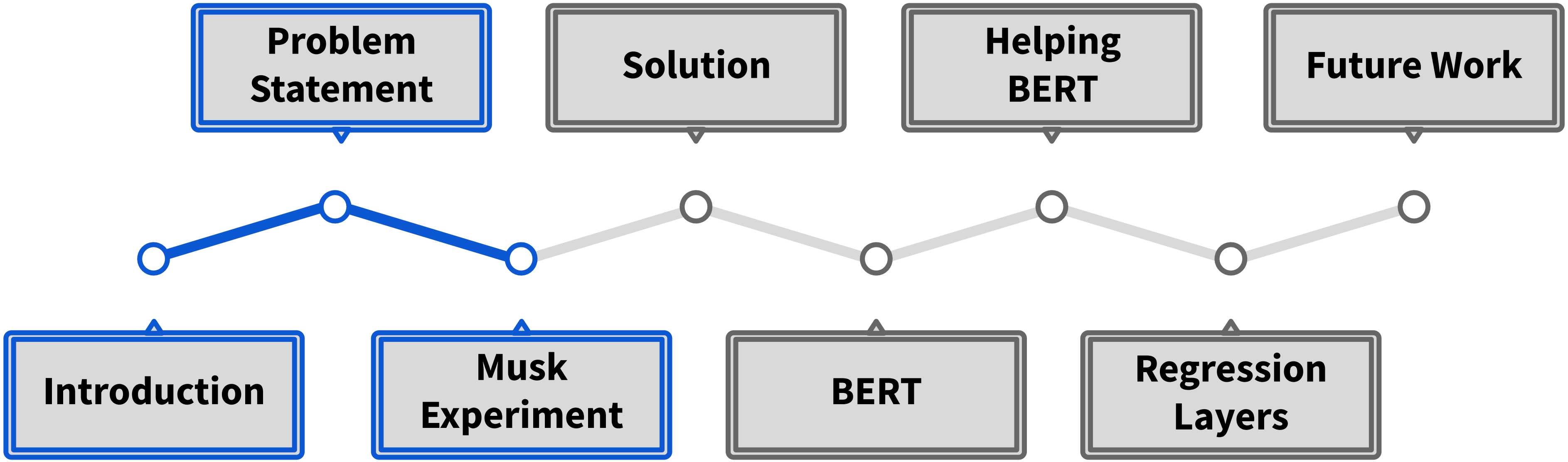
Time Series Analysis + **Sentiment Score**


Introduction
Problem Statement
Musk Experiment
Solution
BERT
Helping BERT
Regression Layers
Future Work

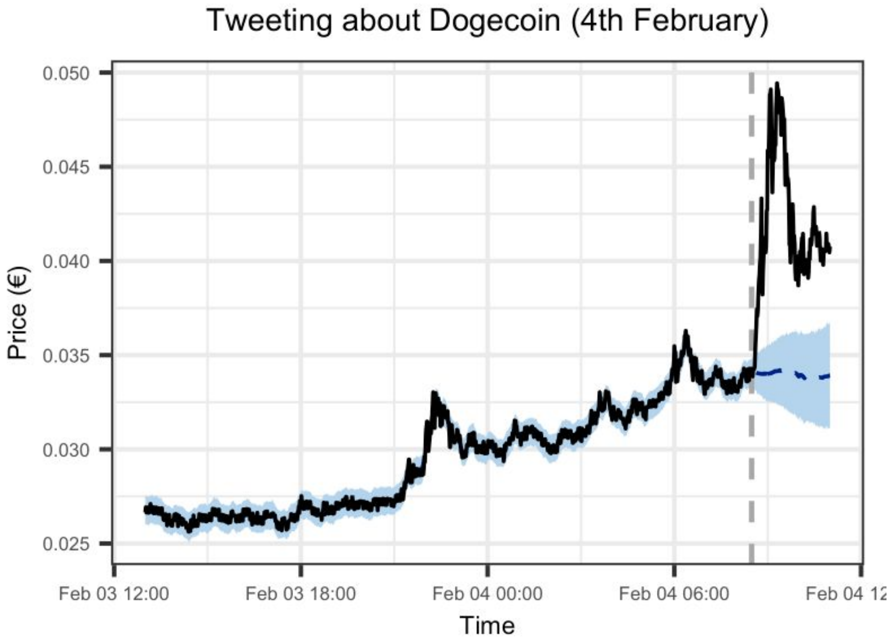
Is sentiment objective?

Which came first, the chicken or the egg?

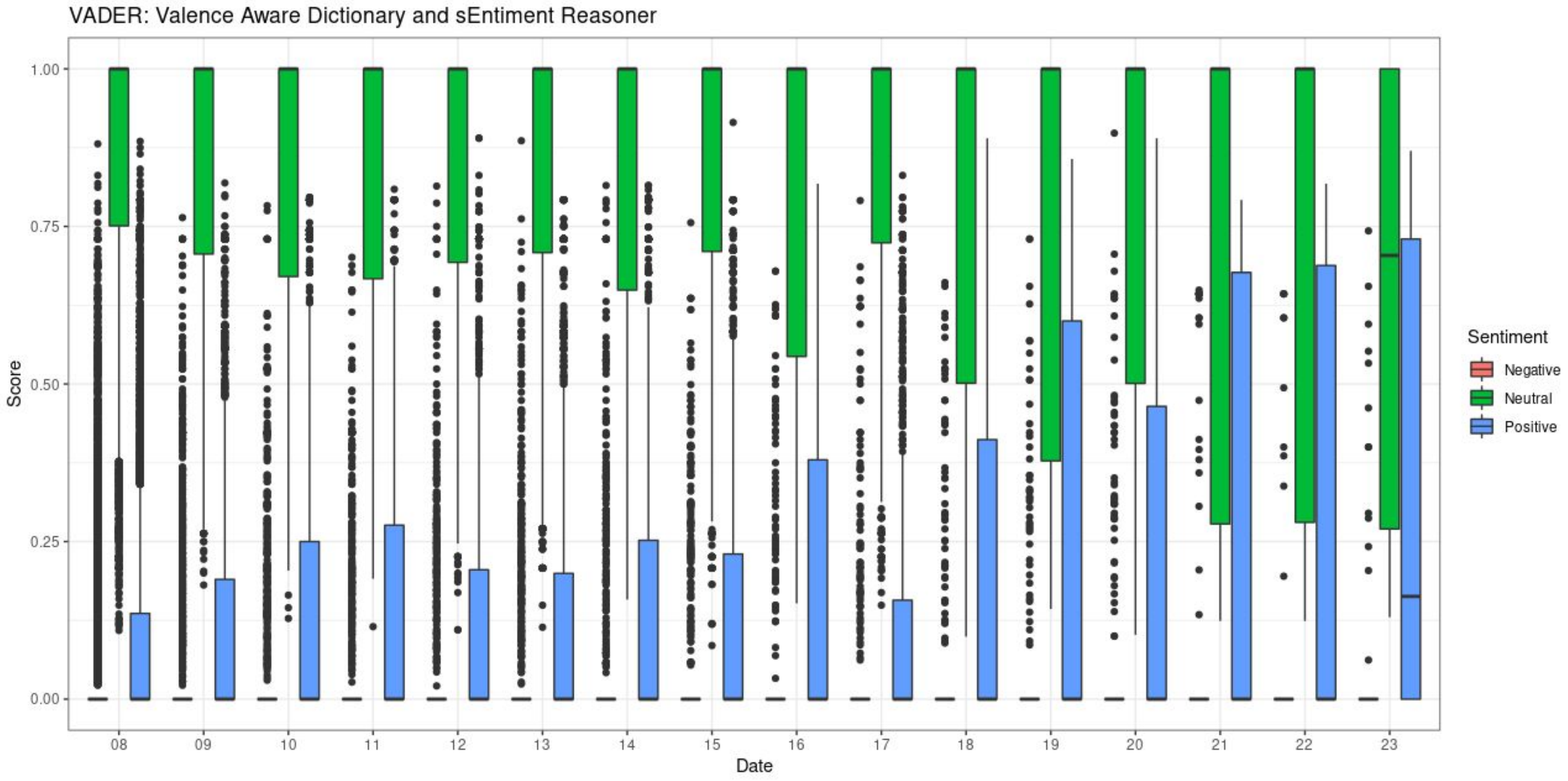

or

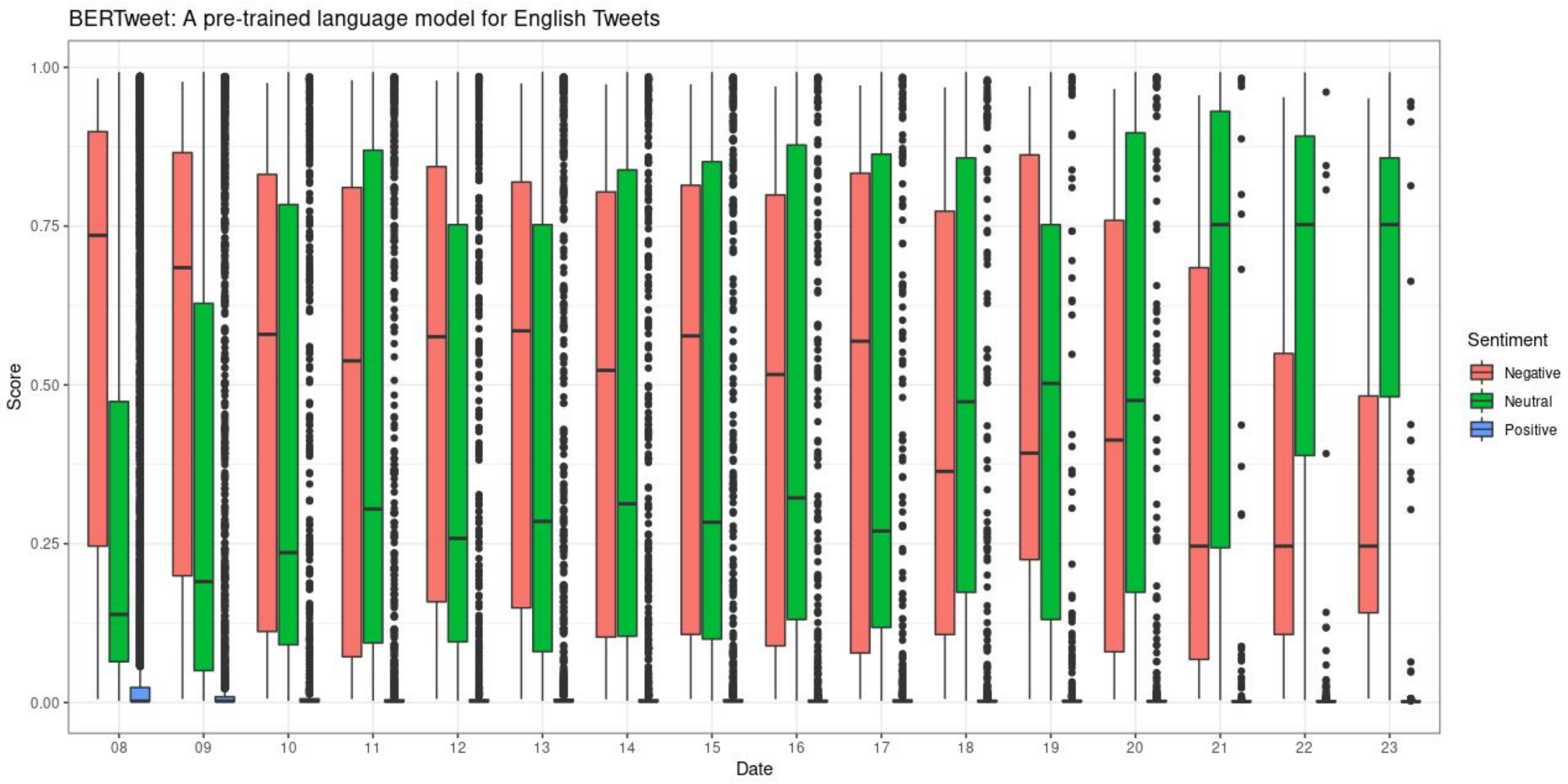


Problem Statement
Solution
Helping BERT
Future Work
Introduction
Musk Experiment
BERT
Regression Layers

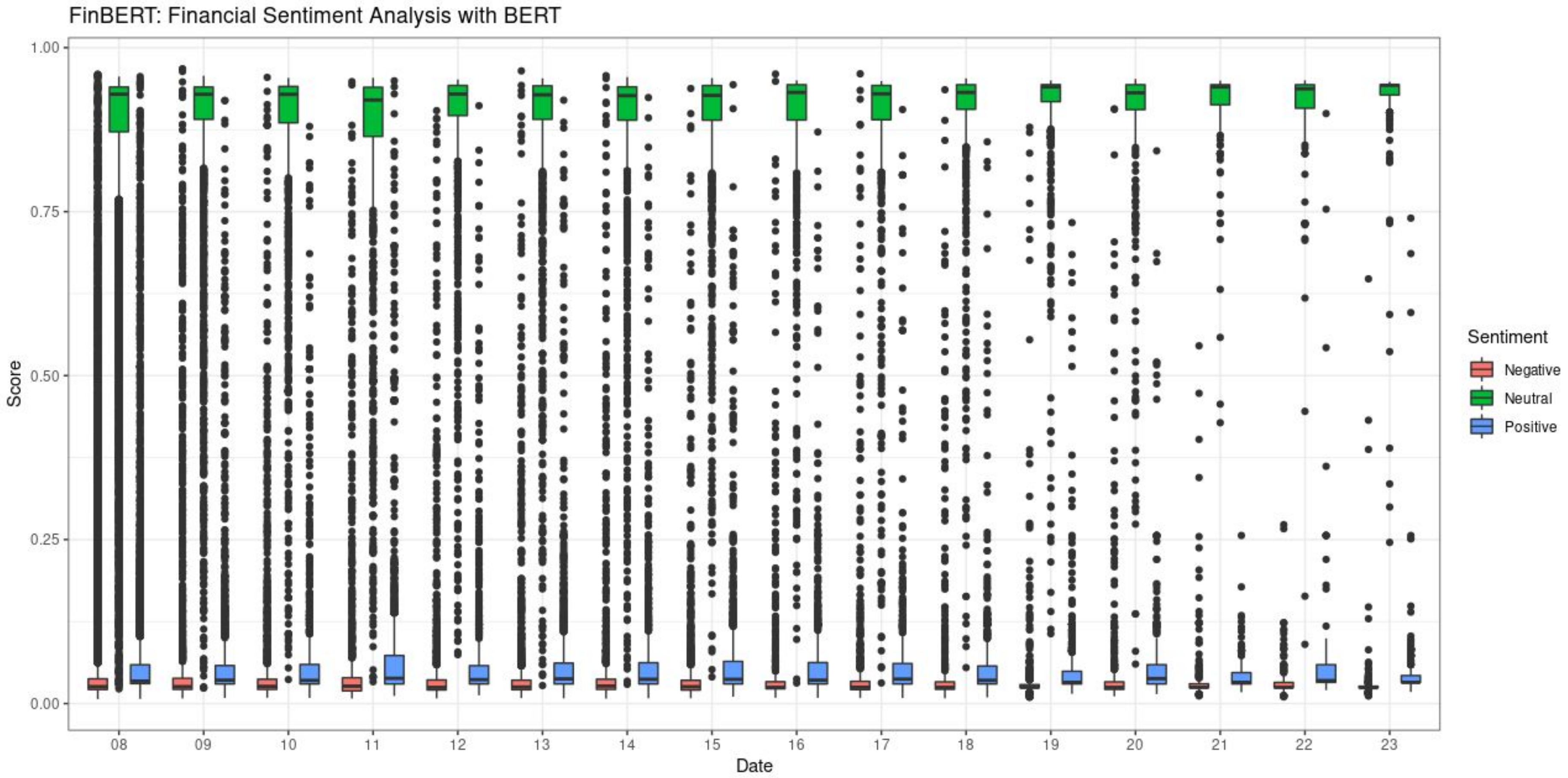
Elon Musk, The Father of Dogecoin

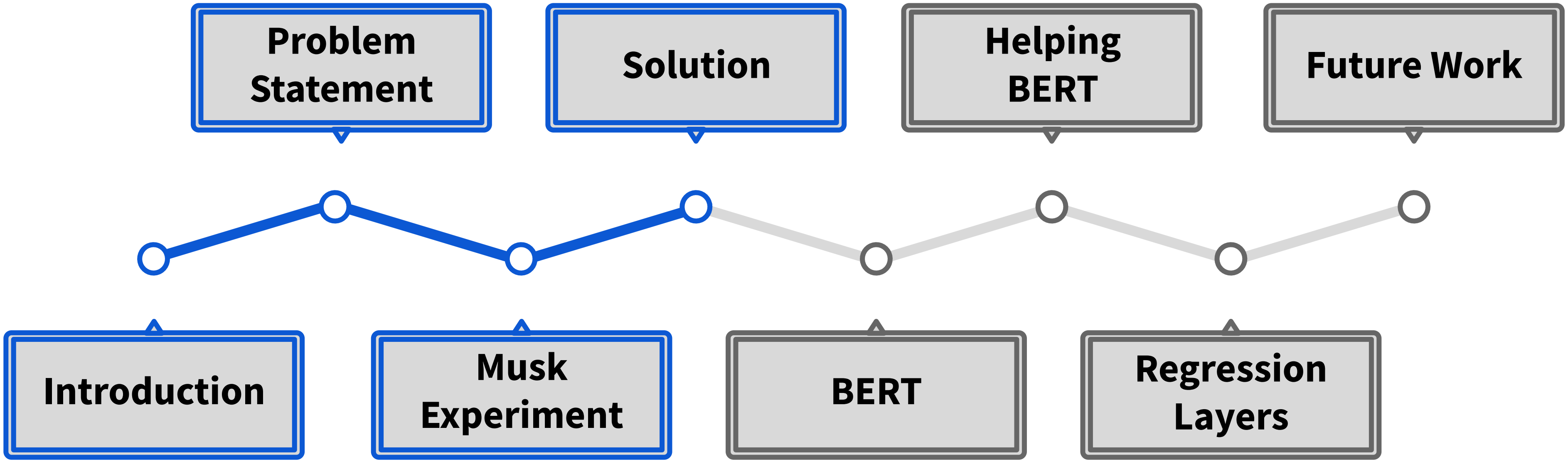


Can Sentiment Analysis Detect It?

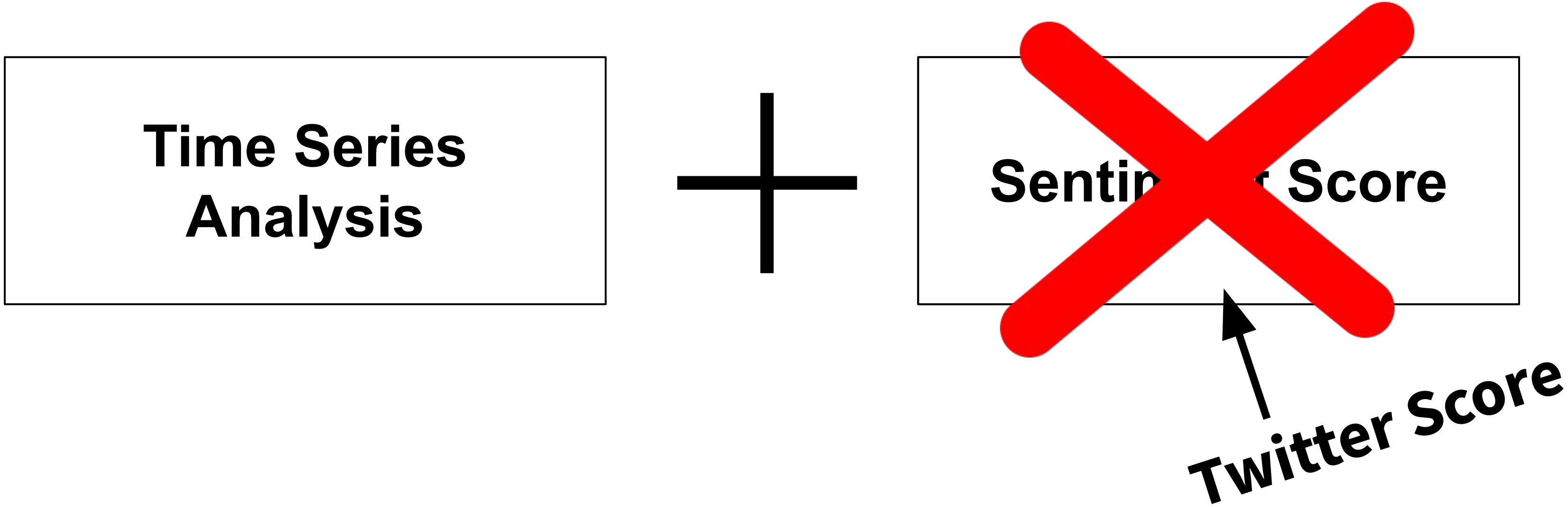

VADER: Valence Aware Dictionary and sEntiment Reasoner
Score
1.00
0.75
0.50
0.25
0.00
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
Date
Sentiment
Negative
Neutral
Positive

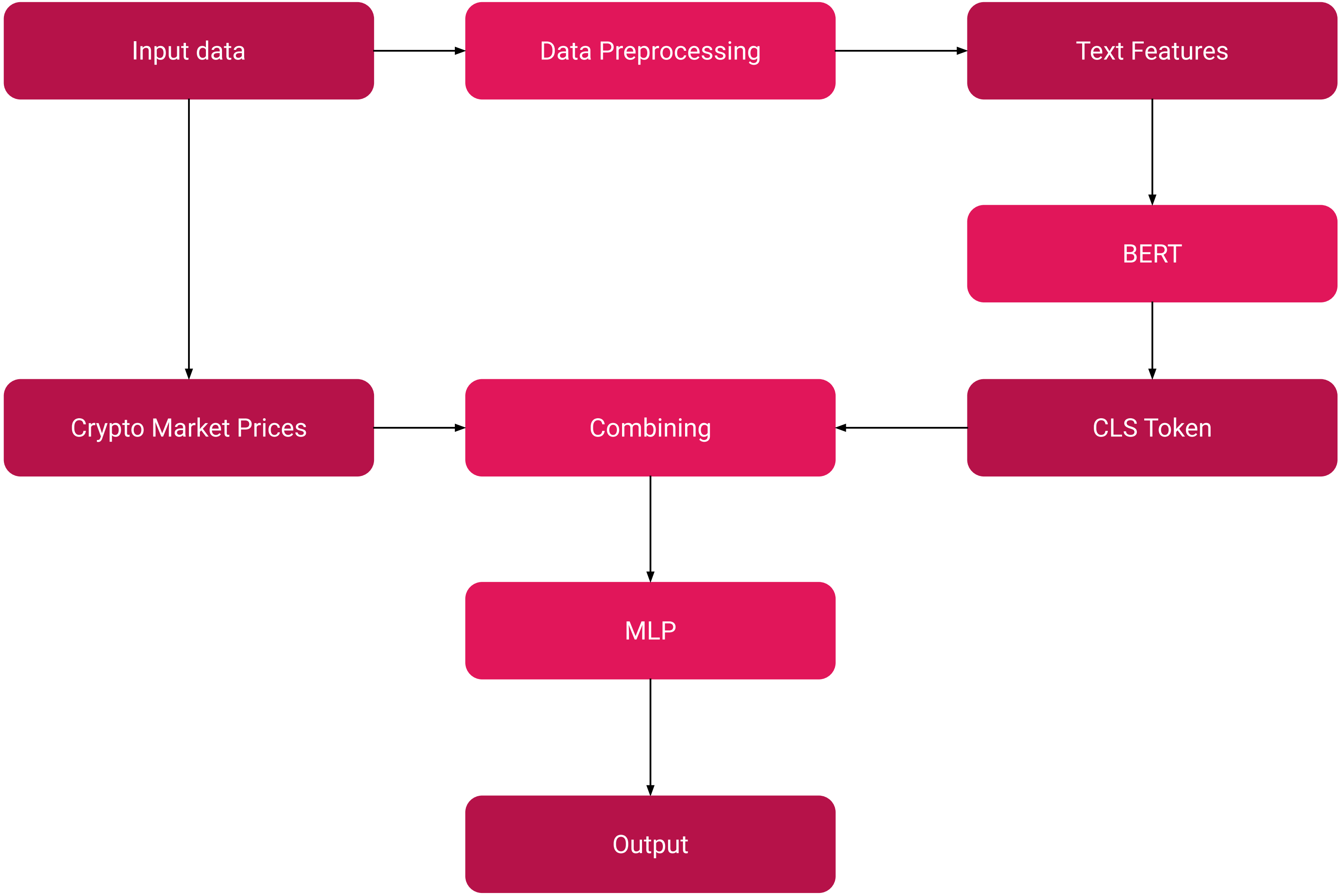

BERTweet: A pre-trained language model for English Tweets
Score
1.00
0.75
0.50
0.25
0.00
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
Date
Sentiment
Negative
Neutral
Positive

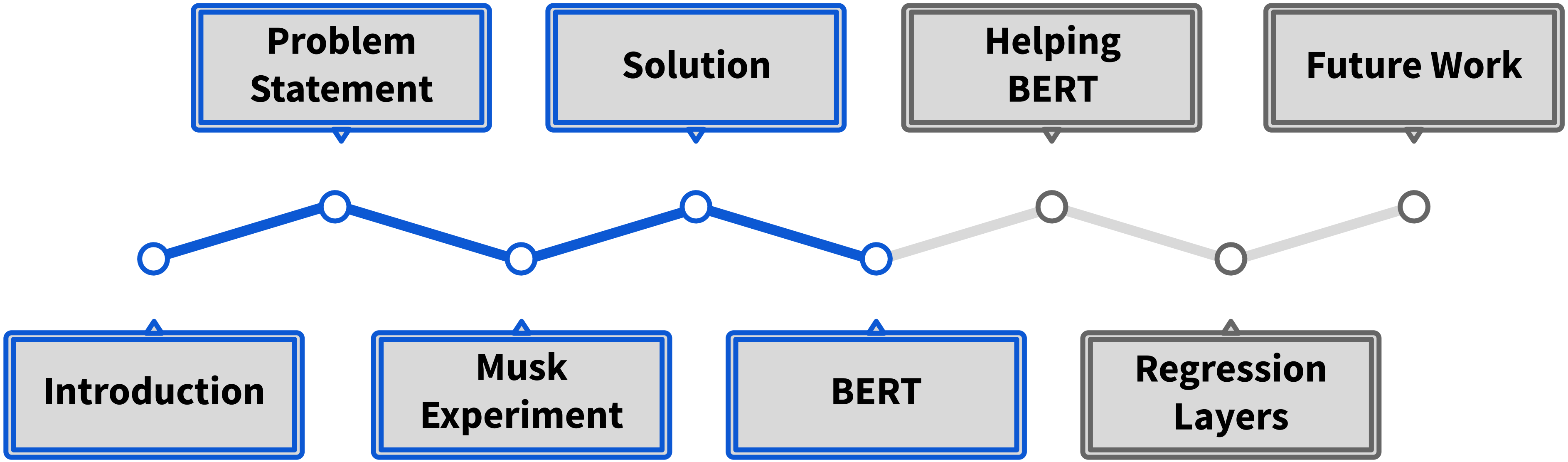

FinBERT: Financial Sentiment Analysis with BERT
Score
1.00
0.75
0.50
0.25
0.00
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
Date
Sentiment
Negative
Neutral
Positive


Introduction
Problem Statement
Musk Experiment
Solution
BERT
Helping BERT
Regression Layers
Future Work

A New Sentiment Score: Twitter Score

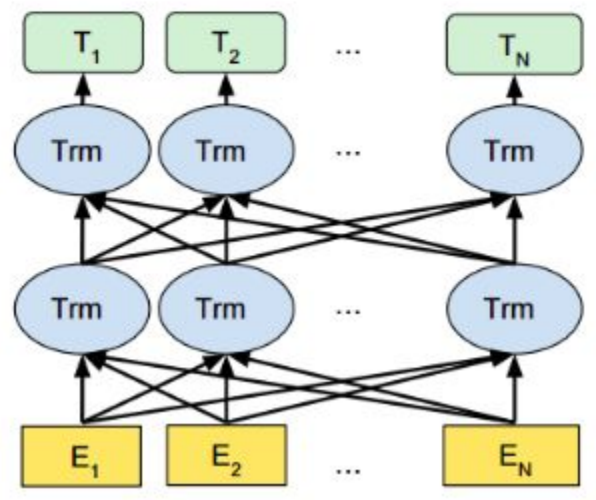



Input data
Data Preprocessing
Text Features
BERT
Crypto Market Prices
Combining
CLS Token
MLP
Output



Problem Statement
Solution
Helping BERT
Future Work
Introduction
Musk Experiment
BERT
Regression Layers

BERT



– Bidirectional
– Encoder
– Representations
– Transformers

BERT Corpus & Parameters

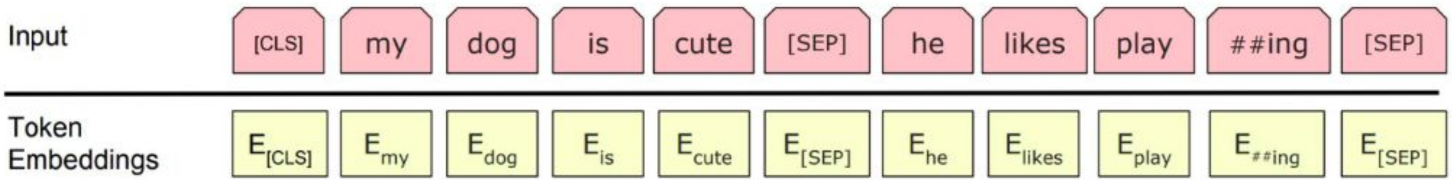
For the pre-training corpus it was used:

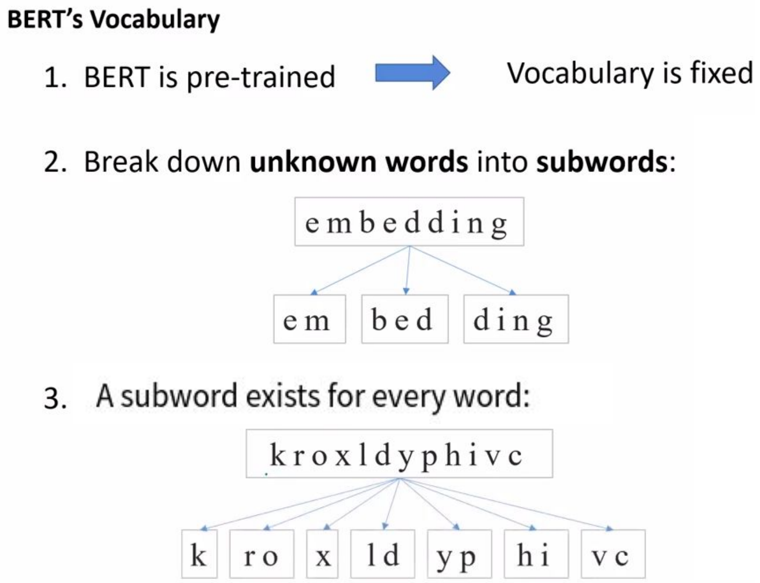
→The BooksCorpus (800M words) (Zhu et al., 2015)

→ English Wikipedia (2,500M words). It was ignored list, tables and header

```
==========================================================================
Layer (type:depth-idx)                              Param #
==========================================================================
├─BertEmbeddings: 1-1                               --
|    └─Embedding: 2-1                               23,440,896
|    └─Embedding: 2-2                               393,216
|    └─Embedding: 2-3                               1,536
|    └─LayerNorm: 2-4                               1,536
|    └─Dropout: 2-5                                 --
├─BertEncoder: 1-2                                  --
|    └─ModuleList: 2-6                              --
|    |    └─BertLayer: 3-1                          7,087,872
|    |    └─BertLayer: 3-2                          7,087,872
|    |    └─BertLayer: 3-3                          7,087,872
|    |    └─BertLayer: 3-4                          7,087,872
|    |    └─BertLayer: 3-5                          7,087,872
|    |    └─BertLayer: 3-6                          7,087,872
|    |    └─BertLayer: 3-7                          7,087,872
|    |    └─BertLayer: 3-8                          7,087,872
|    |    └─BertLayer: 3-9                          7,087,872
|    |    └─BertLayer: 3-10                         7,087,872
|    |    └─BertLayer: 3-11                         7,087,872
|    |    └─BertLayer: 3-12                         7,087,872
├─BertPooler: 1-3                                   --
|    └─Linear: 2-7                                  590,592
|    └─Tanh: 2-8                                    --
==========================================================================
Total params: 109,482,240
```

BERT - Word Embedding and Tokens

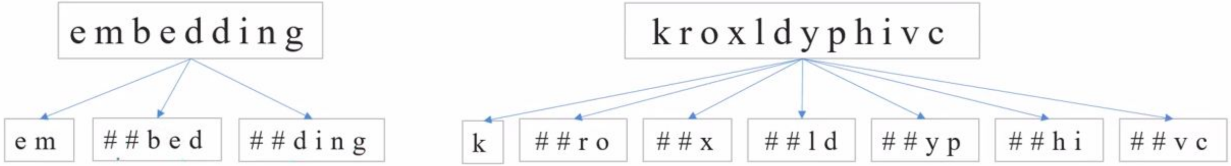




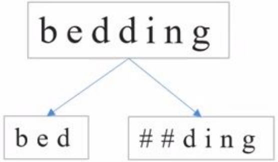
BERT - Word Embedding and Tokens

Types of Subword

All subwords start with “##” …



Except for the first subword in a word.



Special Tokens

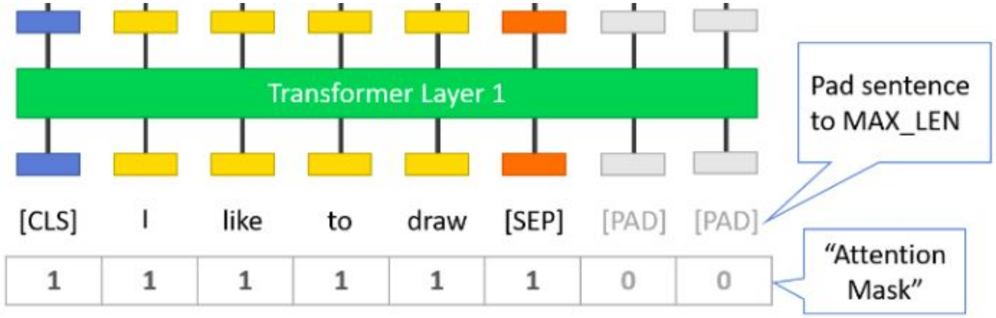
CLS → A special token representing the class of the input

SEP → A special token separating two different sentences in the same input

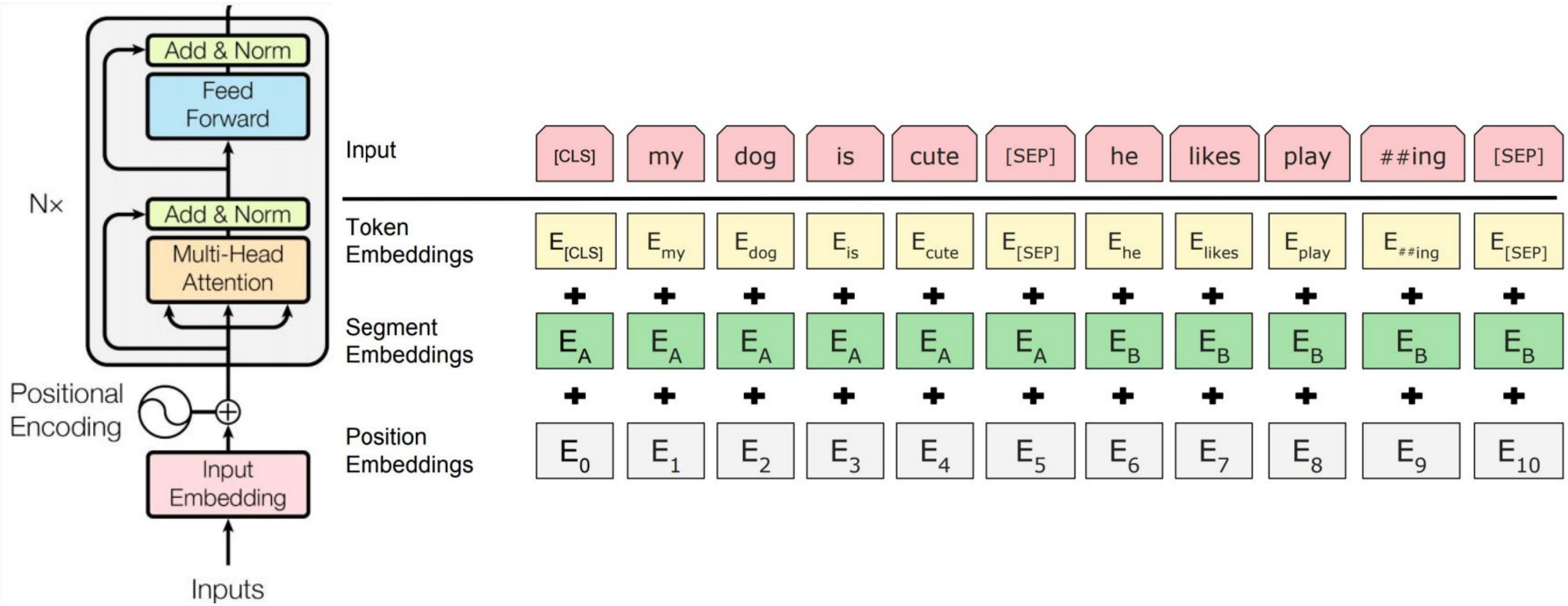
UNKNOWN → A special token representing an out-of-vocabulary token

PAD → A special token used to make arrays of tokens the same size for batching purpose. Will then be ignored by attention mechanisms or loss computation.

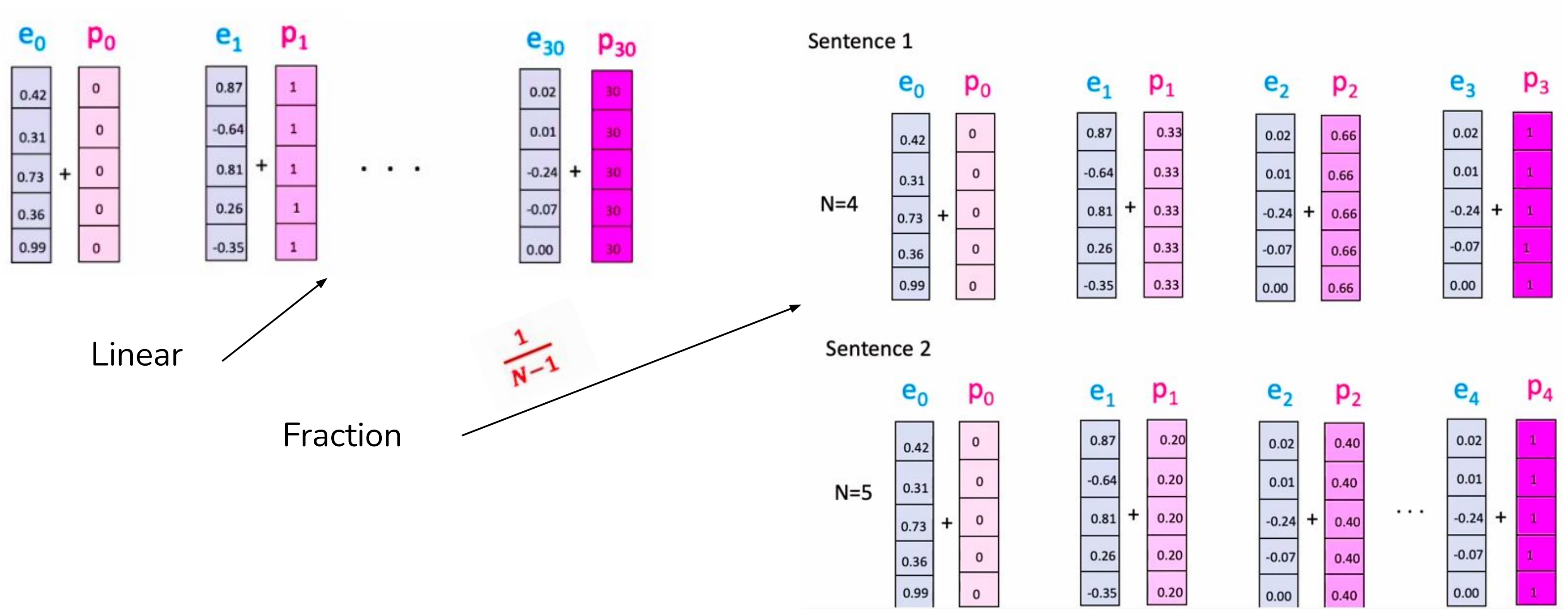
MASK → A special token representing a masked token



BERT - Inner Working



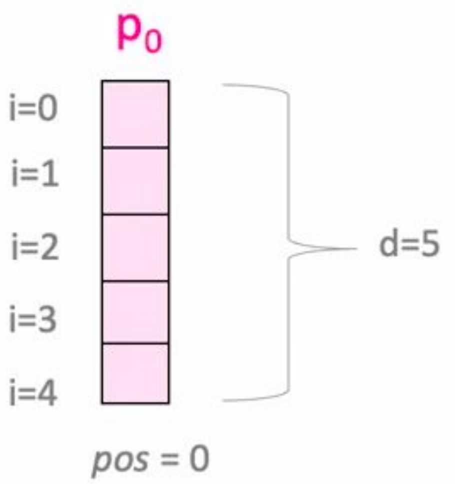
Position Embeddings

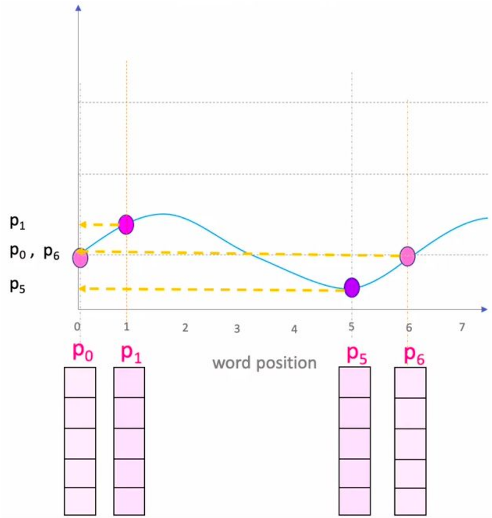


Position Embeddings

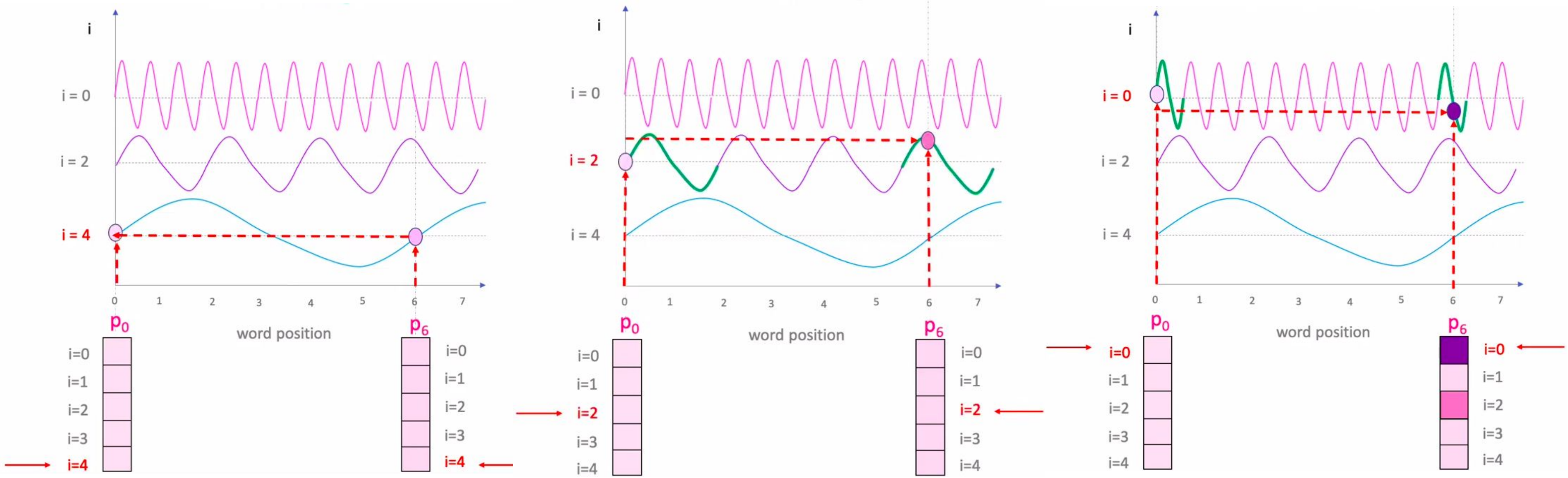
$$PE_{(pos,2i)} = \sin\left(\frac{pos}{10000^{\frac{2i}{d}}}\right)$$

$$PE_{(pos,2i+1)} = \cos\left(\frac{pos}{10000^{\frac{2i}{d}}}\right)$$

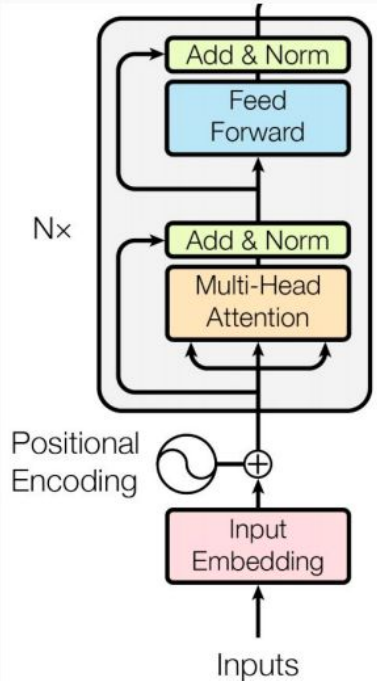


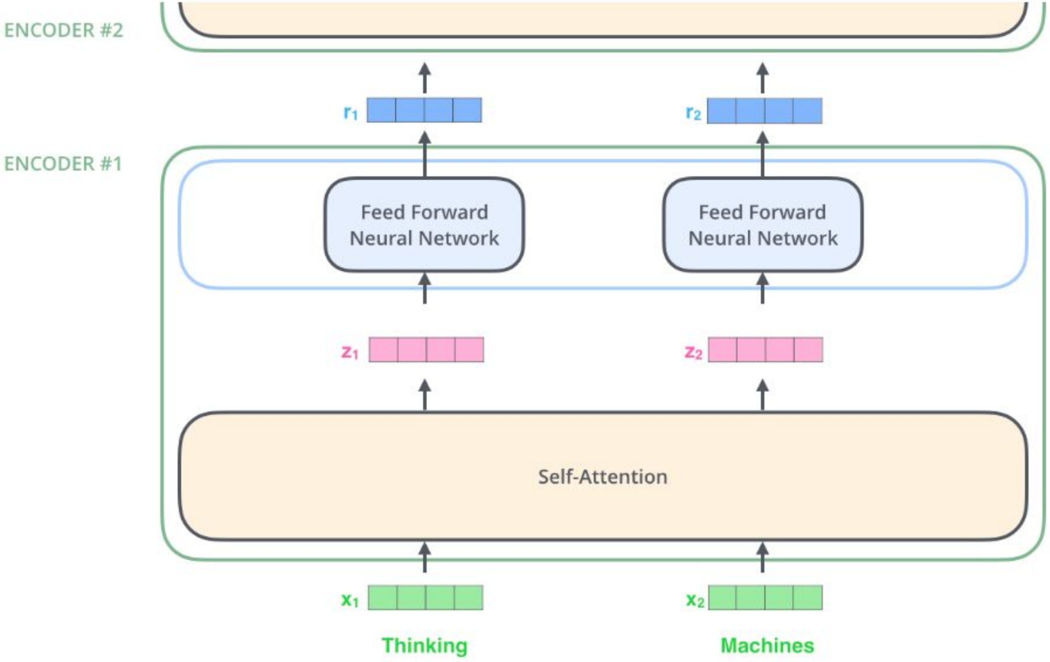


Position Embeddings

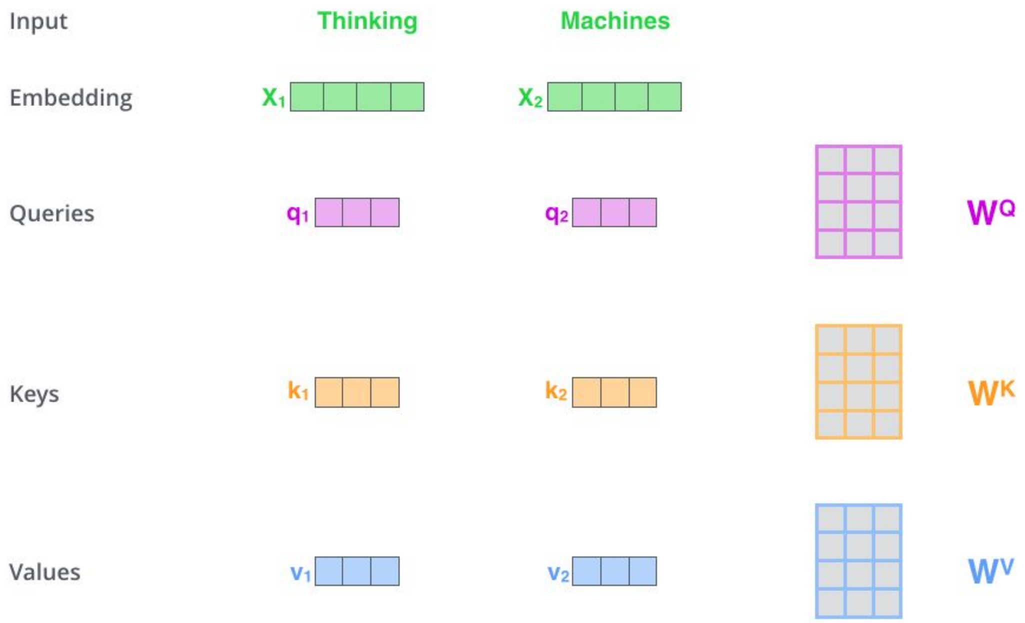


Bert - Inner Working



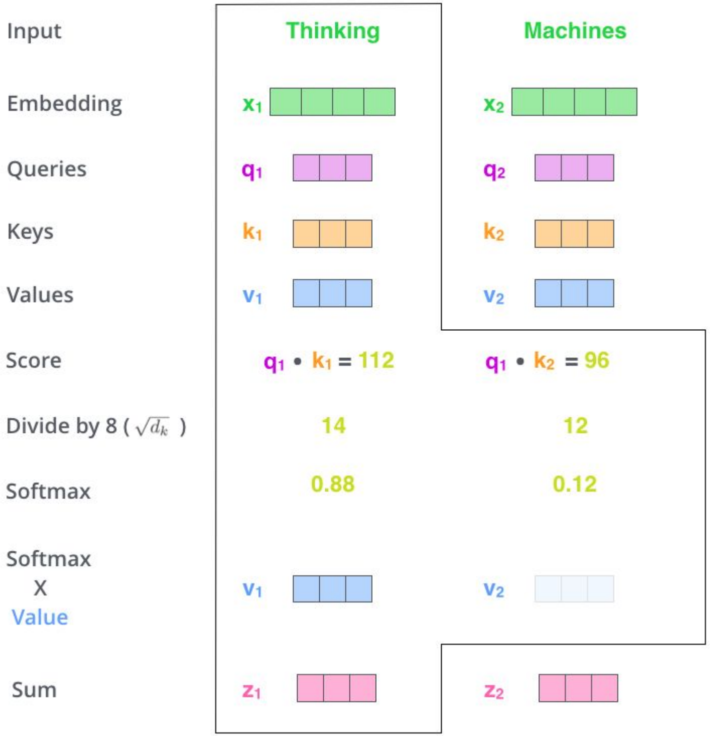


Bert - Inner Working Self Attention in Vector Form

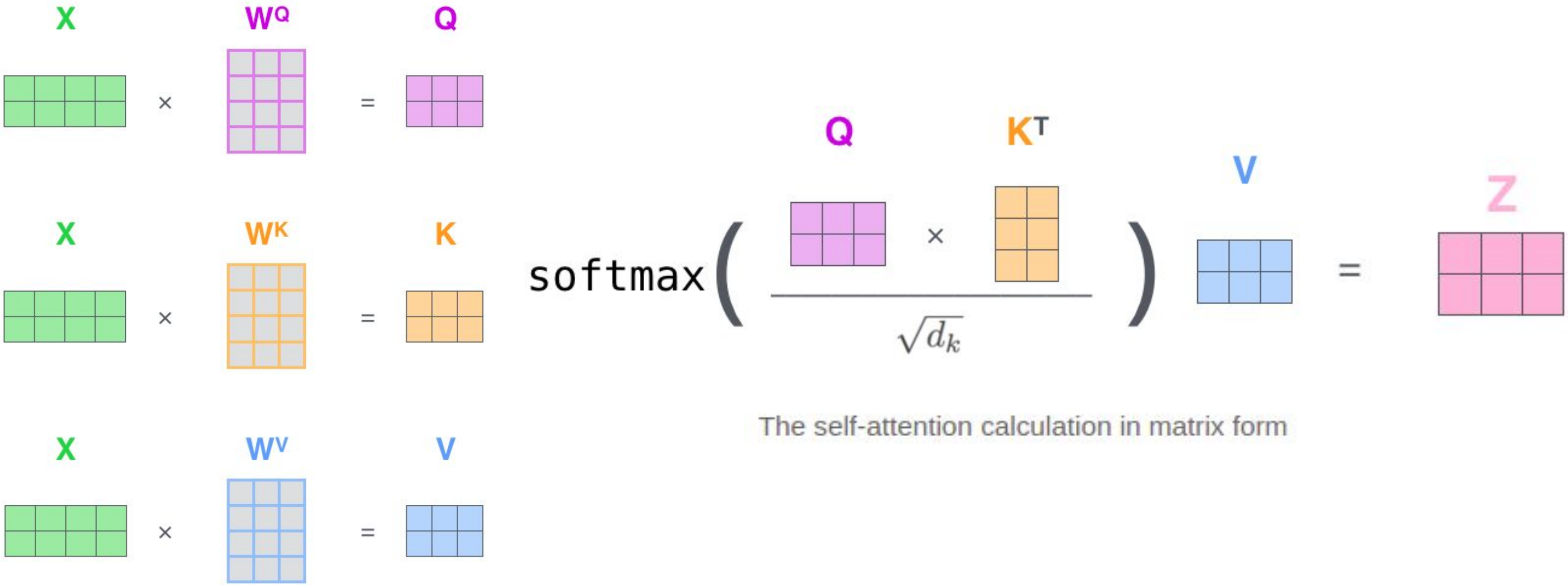


Multiplying x1 by the WQ weight matrix produces q1, the "query" vector associated with that word. We end up creating a "query", a "key", and a "value" projection of each word in the input sentence.

Bert - Inner Working Self Attention in Vector Form

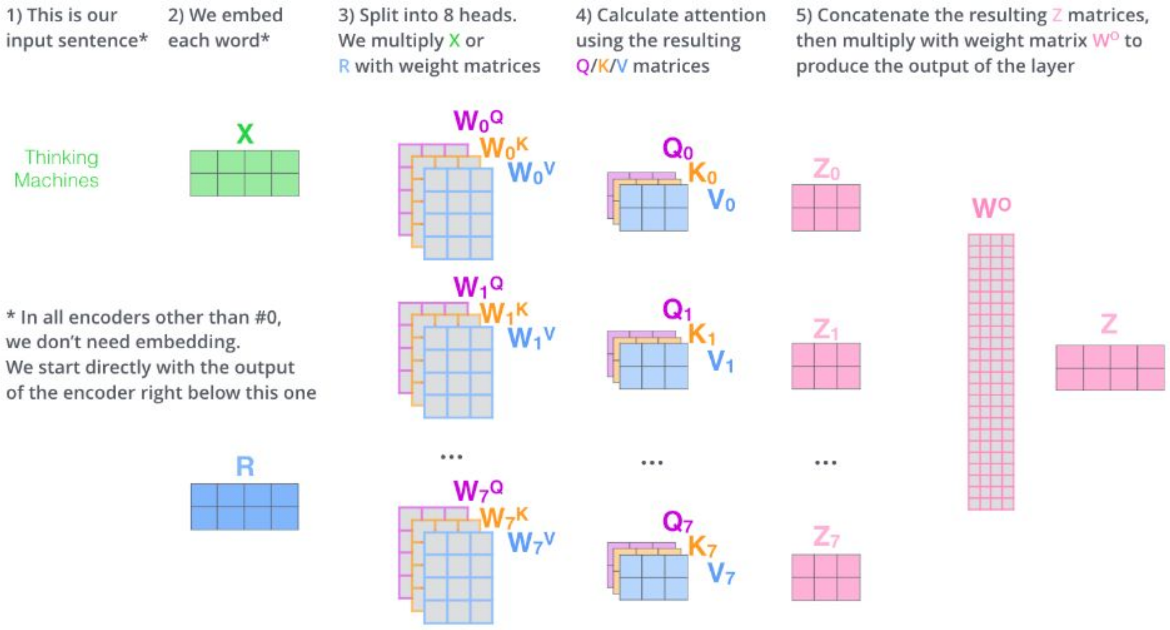


BERT - Inner Working Self Attention in Matrix Form



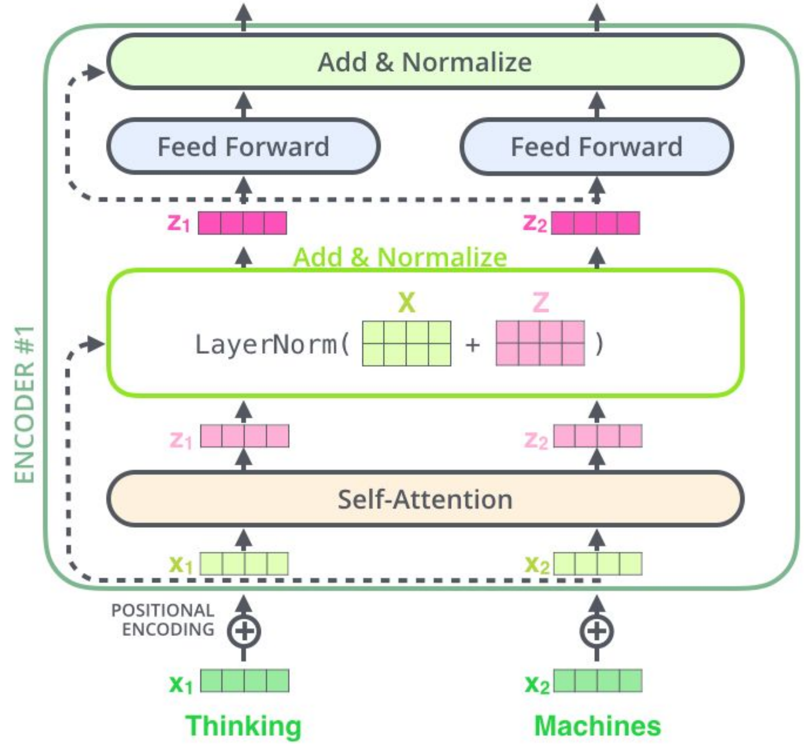
The self-attention calculation in matrix form

BERT - Inner Working Multi-Headed Attention



```
(bert): BertModel(
  (embeddings): BertEmbeddings(
    (word_embeddings): Embedding(30522, 768, padding_idx=0)
    (position_embeddings): Embedding(512, 768)
    (token_type_embeddings): Embedding(2, 768)
    (LayerNorm): LayerNorm((768,), eps=1e-12, elementwise_affine=True)
    (dropout): Dropout(p=0.1, inplace=False)
  )
  (encoder): BertEncoder(
    (layer): ModuleList(
      (0): BertLayer(
        (attention): BertAttention(
          (self): BertSelfAttention(
            (query): Linear(in_features=768, out_features=768, bias=True)
            (key): Linear(in_features=768, out_features=768, bias=True)
            (value): Linear(in_features=768, out_features=768, bias=True)
            (dropout): Dropout(p=0.1, inplace=False)
          )
          (output): BertSelfOutput(
            (dense): Linear(in_features=768, out_features=768, bias=True)
            (LayerNorm): LayerNorm((768,), eps=1e-12, elementwise_affine=True)
            (dropout): Dropout(p=0.1, inplace=False)
          )
        )
        (intermediate): BertIntermediate(
          (dense): Linear(in_features=768, out_features=3072, bias=True)
        )
        (output): BertOutput(
          (dense): Linear(in_features=3072, out_features=768, bias=True)
          (LayerNorm): LayerNorm((768,), eps=1e-12, elementwise_affine=True)
          (dropout): Dropout(p=0.1, inplace=False)
        )
      )
```

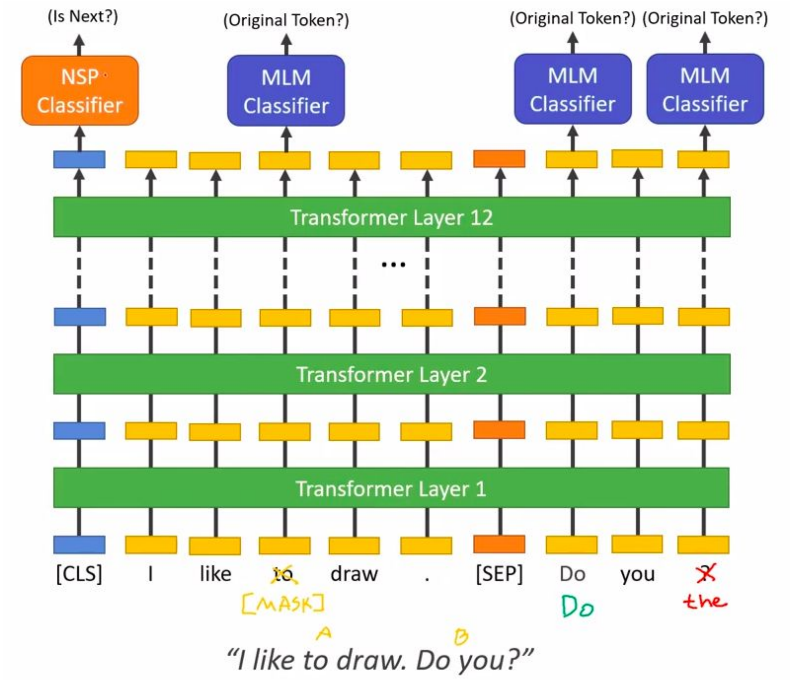
BERT - Add & Normalize

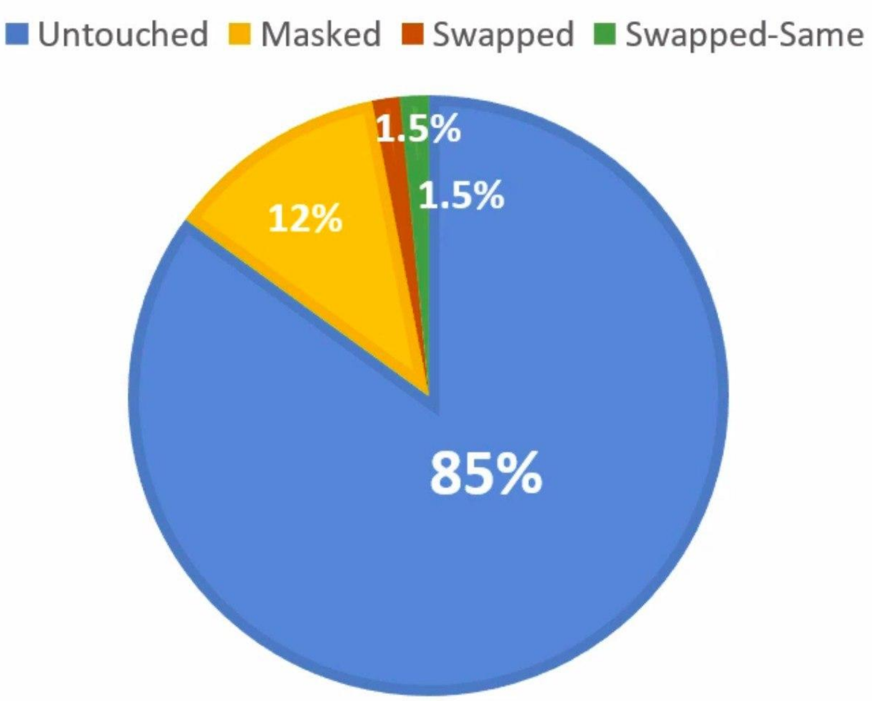


	f_0	f_1	f_2	f_3	f_4	mean (μ)	std (σ)
x_0 = When	0.98	1.28	0.41	0.27	0.41	0.67	0.44
x_1 = you	0.52	0.01	2.06	0.27	0.33	0.64	0.82
x_2 = play	2.22	0.27	0.10	0.41	2.06	1.01	1.04
x_3 = the	0.99	1.00	0.11	0.27	0.33	0.54	0.42
x_4 = game	0.52	0.01	0.33	2.06	0.52	0.69	0.79
x_5 = of	0.10	2.06	0.73	0.27	0.41	0.71	0.79
x_6 = thrones	0.33	0.01	0.13	0.27	1.28	0.40	0.51

$$x_i = \frac{x_i^d - \mu}{\sqrt{\sigma^2 + \epsilon}}$$

BERT - Pre-Training Tasks



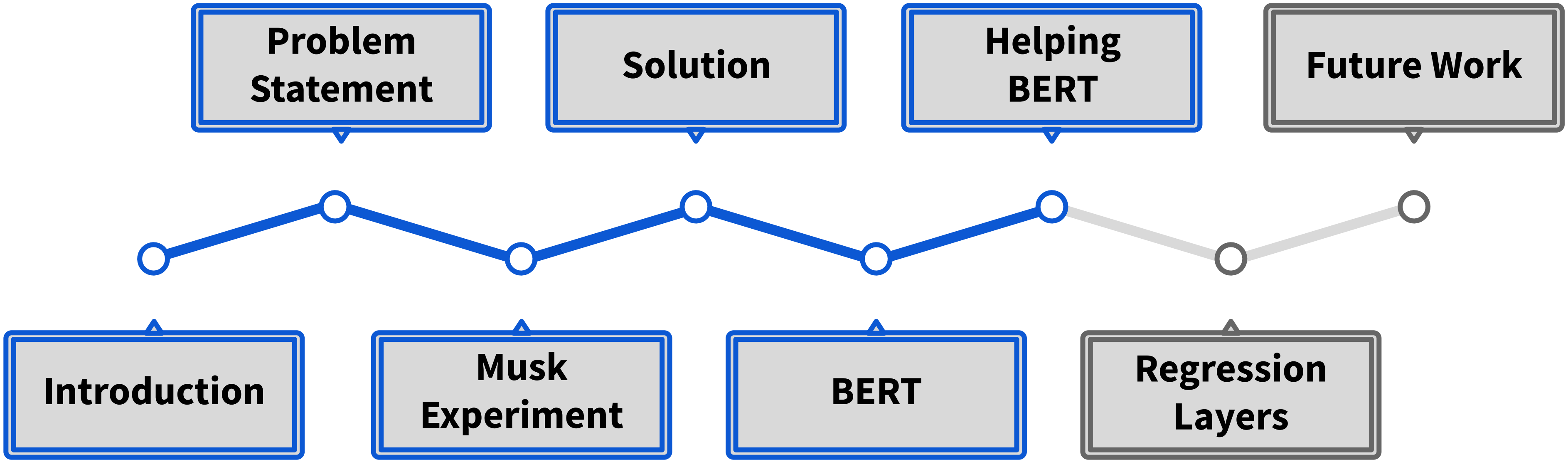


BERT - Shortcomings

BERT is very large → Embedding Layer with 24M weights (30K tokens each of 768 values)
→ Transformers (12) with 85M weights
→ slow fine-tuning
→ slow inferencing

Jargon → domain-specific language

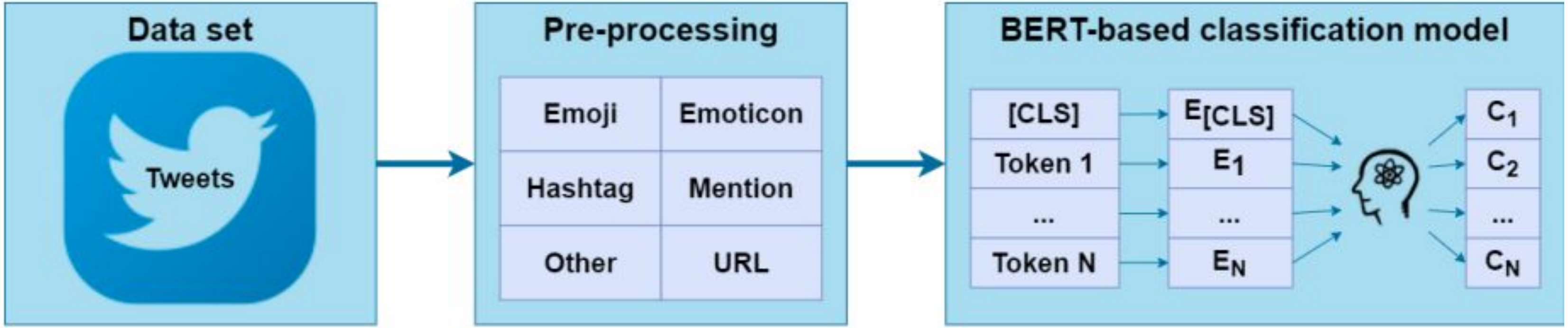
Model		Parameters	Layers	Hidden	Embedding
BERT	base	108M	12	768	768
	large	334M	24	1024	1024
	xlarge	1270M	24	2048	2048


Problem Statement
Solution
Helping BERT
Future Work
Introduction
Musk Experiment
BERT
Regression Layers

01	Help BERT with Informal Language	• Preprocessing Data
02	Slow Inferencing	• Dynamic Padding
03	Heavy Data Management	• Batch Division

Data Preprocessing → Tweet to BERT

Tweets are characterized by a **particular slang** and sometimes the important information are included
in this **very noisy and obscure** dataset, due to people's random and **creative use of social media**.



Data Preprocessing → Uninterpretable Words

According to the literature [4] we tried to preprocess the dataset with two particular aims:

REDUCE NOISE

- dates
- email
- money
- percentage
- url
- time
- phone
- number

THE “SPARKLY” TWITTER CREATIVITY

- user
- cashtag
- hashtag
- emoji

Data Preprocessing → Examples

- 30/02/2003 ➔ <date>
- $42 ➔ <money>
- 120% ➔ <percentage>
- 4:20PM ➔ <time>
- 351.8744170 ➔ <phone>
- 42 ➔ <number>
- https://da/S4m45aRu13z\n\nTrial ➔ <url>

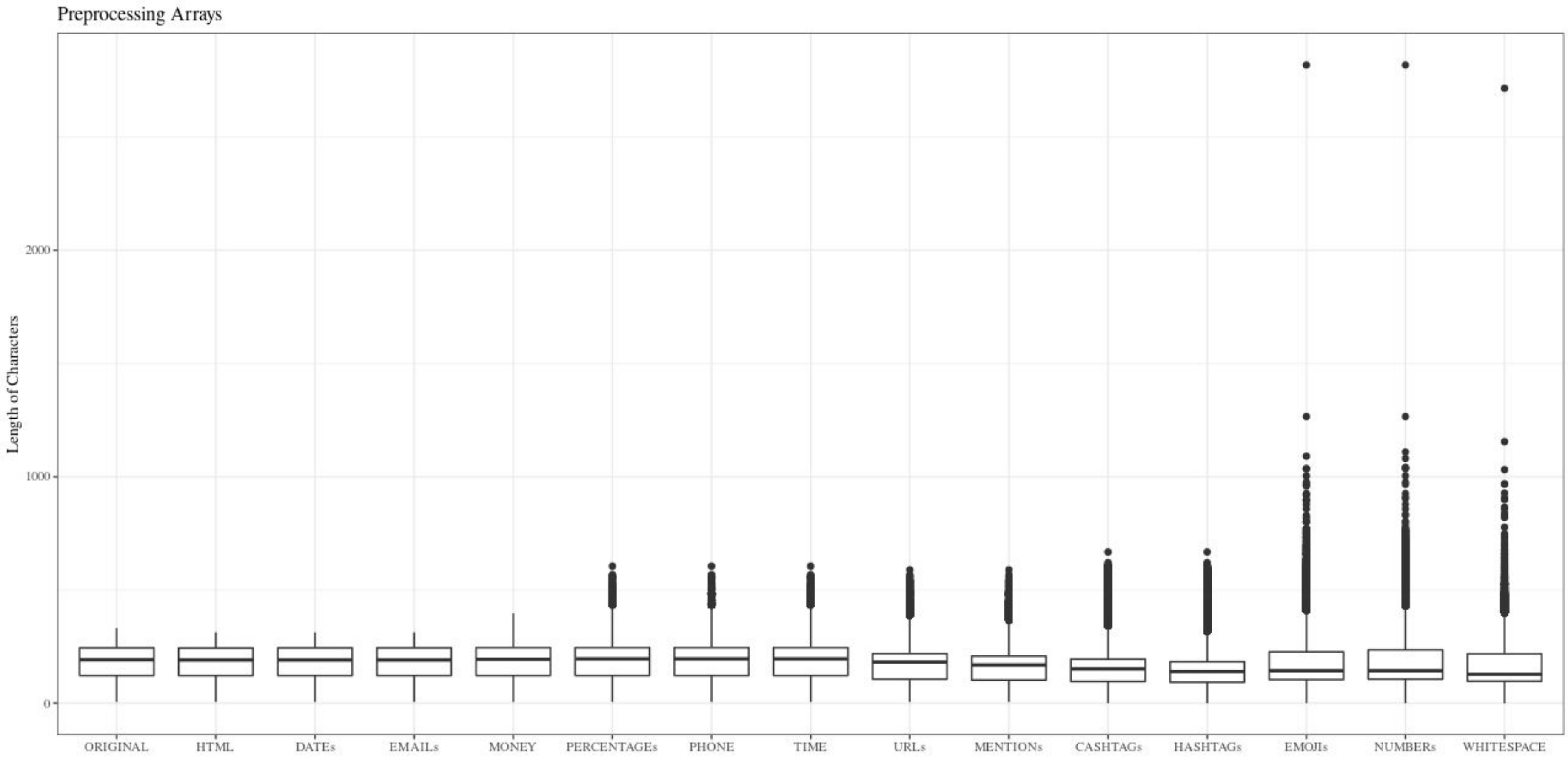
Data Preprocessing → Examples

- @mybeautifulaccount ➔ @user
- $BTC $BTCpizzamandolino ➔ <cashtag> <cashtag>
- #lamboformambo ➔ <lambo for mambo>
- 😂 ➔ Rolling on the Floor Laughing

Data preprocessing → Results

"- 💰 Buy now/pay later 🚀#BlackFriday\n- 🌎 World’s largest art lending platform coming to @Algorand.\n- Digital asset exchanges coming to $algo\n- #Algorand is interoperable 🤩 \n- Faster than $eth, better tech, green 🌱\nStart watching@ 5:42:00 ➡️ https://t.co/mtSxwVMUCT\n#AlgoAutumn"

"- **money bag** Buy now/pay later **rocket <Black Friday>** - **globe** showing Americas World’s largest art lending platform coming to . - Digital asset exchanges coming to **<cashtag>** - is interoperable star-struck - Faster than **<cashtag>** better tech, green seedling Start watching@ **<time>** right arrow **<url>**"


Preprocessing Arrays
Length of Characters
2000
1000
0
ORIGINAL
HTML
DATEs
EMAILs
MONEY
PERCENTAGEs
PHONE
TIME
URLs
MENTIONs
CASHTAGs
HASHTAGs
EMOJIs
NUMBERs
WHITESPACE

Fixed Padding Length

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	_Eh	_bien	_c	'	_est	_un	_bon	_indicateu	[PAD]	[PAD]	[PAD]	[PAD]	[PAD]	[PAD]
2	_Ouais	_je	_suis	_un	_coureur	[PAD]	[PAD]	[PAD]	[PAD]	[PAD]	[PAD]	[PAD]	[PAD]	[PAD]
3	_Ils	_ne	_sont	_pas	important	_	.	[PAD]	[PAD]	[PAD]	[PAD]	[PAD]	[PAD]	[PAD]
4	_Il	_y	_a	_de	nombreus	_condition	_qui	_ne	_sont	_pas	_visibles	_	.	[PAD]
5	_Chaque	_zone	_de	_l	'	_île	_offre	_quelque	_chose	_de	_différent	_	.	[PAD]
6	_Mais	_tu	_peux	_vivre	_avec	_eux	_	.	[PAD]	[PAD]	[PAD]	[PAD]	[PAD]	[PAD]
7	_Un	_grand	_homme	_	,	_dit	-	il	_	.	[PAD]	[PAD]	[PAD]	[PAD]
8	_Elle	_a	_été	_menée	_en	_silence	_	.	[PAD]	[PAD]	[PAD]	[PAD]	[PAD]	[PAD]
9	_Tu	er	_beaucou	_de	_fourmis	_de	_feu	[PAD]	[PAD]	[PAD]	[PAD]	[PAD]	[PAD]	[PAD]
10	_La	_question	_est	_de	_savoir	_si	_clin	ton	_a	_le	_cul	ot	_	.
11	_C	'	_est	_vrai	_	.	[PAD]	[PAD]	[PAD]	[PAD]	[PAD]	[PAD]	[PAD]	[PAD]
12	_Dans	_ce	_domaine	_	,	_seuls	_les	_sa	ther	i	_le	_savent	_	.

Batch Length: **14**

Batch Length: **14**

Batch Length: **14**

Total Tokens: **168**

Dynamic Padding

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	_Eh	_bien	_c	'	_est	_un	_bon	indicateu	[PAD]	[PAD]	[PAD]	[PAD]	[PAD]	
2	_Ouais	_je	_suis	_un	_coureur	[PAD]	[PAD]	[PAD]	[PAD]	[PAD]	[PAD]	[PAD]	[PAD]	
3	_Ils	_ne	_sont	_pas	important	_	.	[PAD]	[PAD]	[PAD]	[PAD]	[PAD]	[PAD]	
4	_Il	_y	_a	_de	ombreus	condition	_qui	_ne	_sont	_pas	_visibles	_	.	
5	_Chaque	_zone	_de	_l	'	_île	_offre	_quelque	_chose	_de	_différent	_	.	
6	_Mais	_tu	_peux	_vivre	_avec	_eux	_	.	[PAD]	[PAD]	[PAD]	[PAD]	[PAD]	
7	_Un	_grand	_homme	_	,	_dit	-	il	_	.	[PAD]	[PAD]	[PAD]	
8	_Elle	_a	_été	_menée	_en	_silence	_	.	[PAD]	[PAD]	[PAD]	[PAD]	[PAD]	
9	_Tu	er	beaucou	_de	_fourmis	_de	_feu	[PAD]	[PAD]	[PAD]	[PAD]	[PAD]	[PAD]	[PAD]
10	_La	_question	_est	_de	_savoir	_si	_clin	ton	_a	_le	_cul	ot	_	.
11	_C	'	_est	_vrai	_	.	[PAD]	[PAD]	[PAD]	[PAD]	[PAD]	[PAD]	[PAD]	[PAD]
12	_Dans	_ce	_domaine	_	,	_seuls	_les	_sa	ther	i	_le	_savent	_	.

Batch Length: **13**

Batch Length: **13**

Batch Length: **14**

Total Tokens: **160**

Uniform Length Batching (Our Approach)

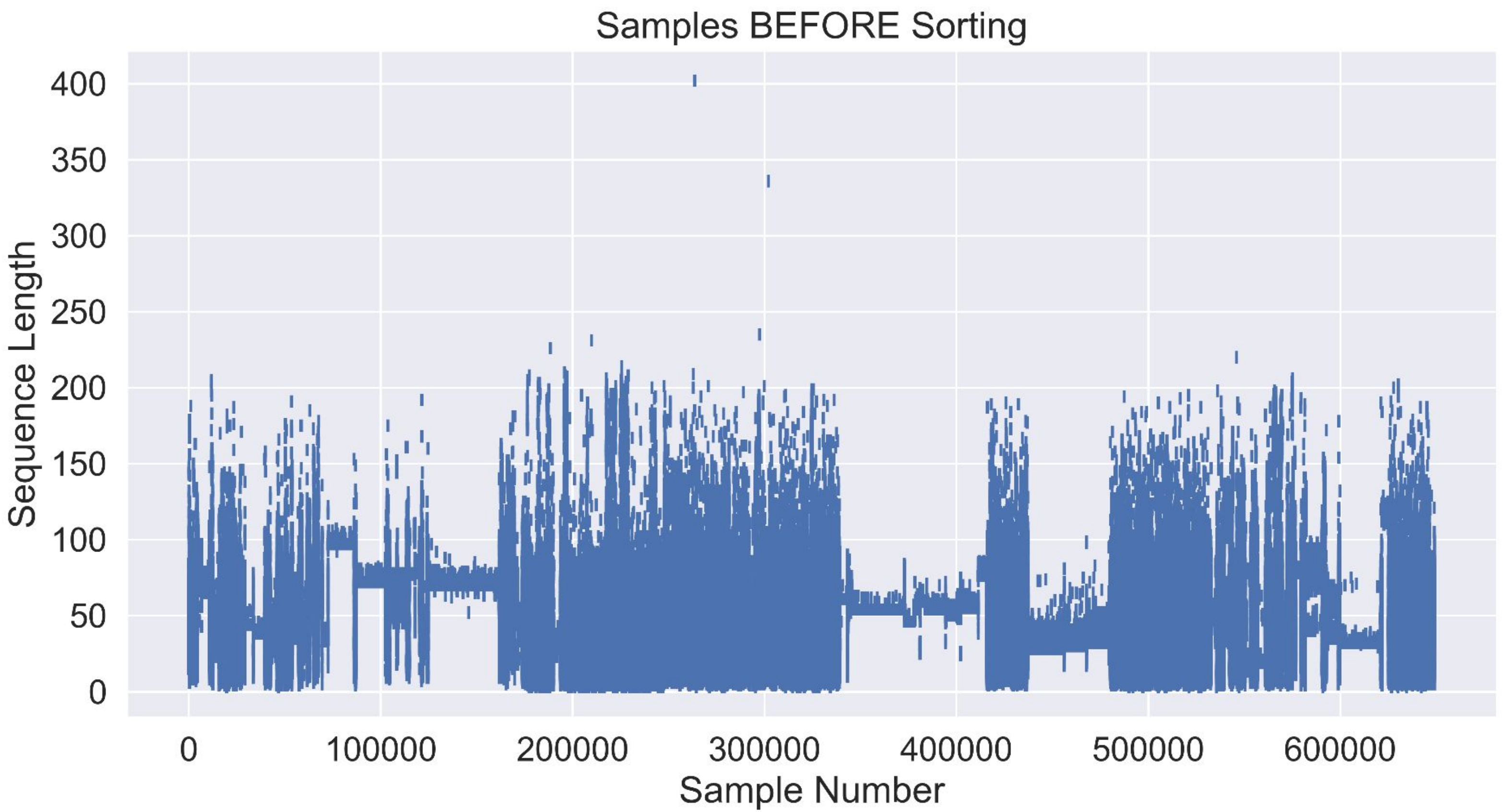
	1	2	3	4	5	6	7	8	9	10	11	12	13	14
2	_Ouais	_je	_suis	_un	_coureur	[PAD]	[PAD]							
11	_C	'	_est	_vrai	_	.	[PAD]							
3	_Ils	_ne	_sont	_pas	important	_	.							
9	_Tu	er	_beaucou	_de	_fourmis	_de	_feu							
1	_Eh	_bien	_c	'	_est	_un	_bon	_indicateu	[PAD]	[PAD]				
6	_Mais	_tu	_peux	_vivre	_avec	_eux	_	.	[PAD]	[PAD]				
8	_Elle	_a	_été	_menée	_en	_silence	_	.	[PAD]	[PAD]				
7	_Un	_grand	_homme	_	,	_dit	-	il	_	.				
5	_Chaque	_zone	_de	_l	'	_île	_offre	_quelque	_chose	_de	_différent	_	.	[PAD]
4	_Il	_y	_a	_de	ombreus	condition	_qui	_ne	_sont	_pas	_visibles	_	.	[PAD]
10	_La	_question	_est	_de	_savoir	_si	_clin	ton	_a	_le	_cul	ot	_	.
12	_Dans	_ce	_domaine	_	,	_seuls	_les	_sa	ther	i	_le	_savent	_	.

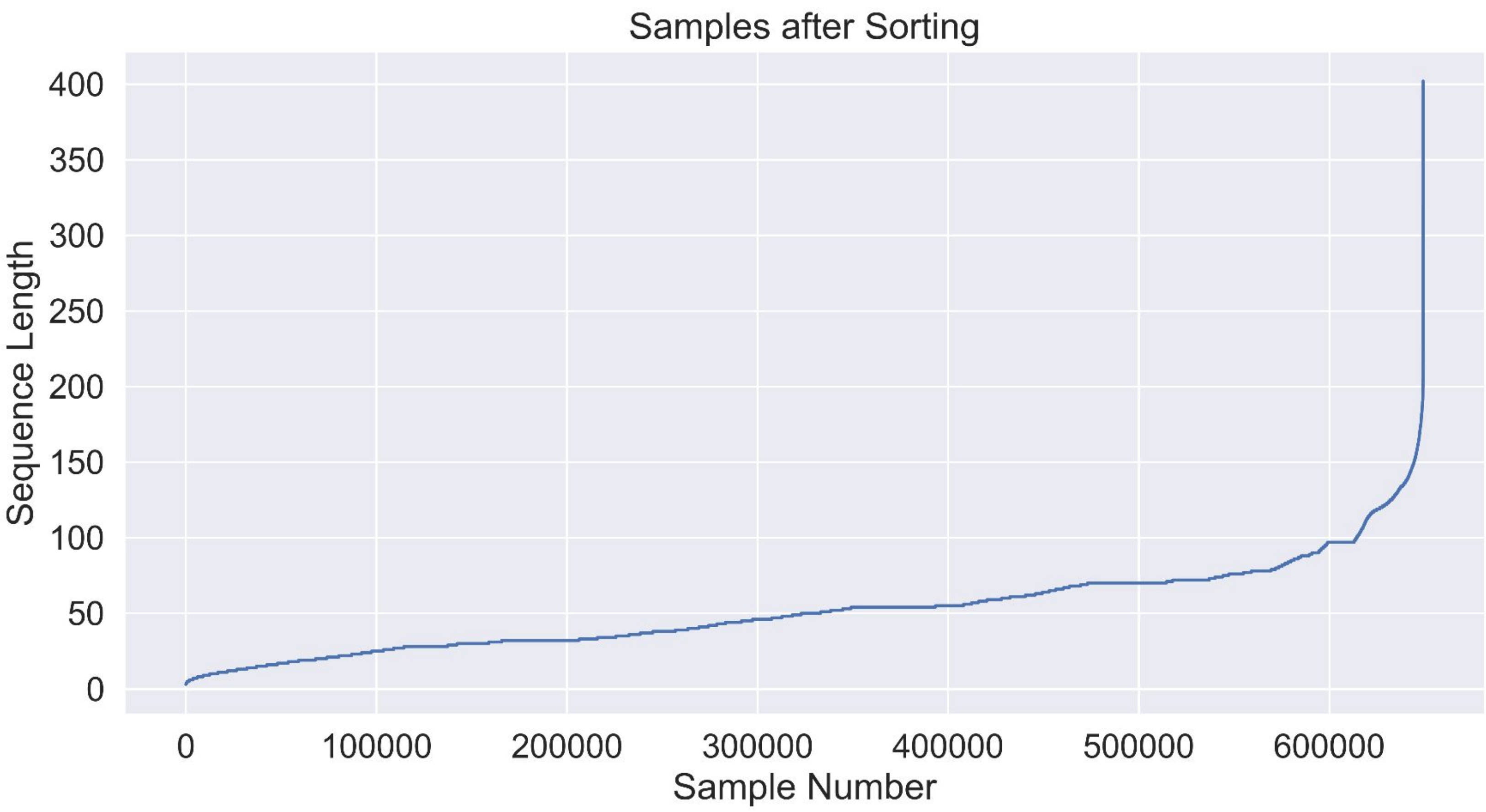
Batch Length: **7**

Batch Length: **10**

Batch Length: **14**

Total Tokens: **124**


Samples BEFORE Sorting
Sequence Length
400
350
300
250
200
150
100
50
0
0
100000
200000
300000
400000
500000
600000
Sample Number



Samples after Sorting
Sequence Length
400
350
300
250
200
150
100
50
0
0
100000
200000
300000
400000
500000
600000
Sample Number

Impact of PAD Token on Accuracy

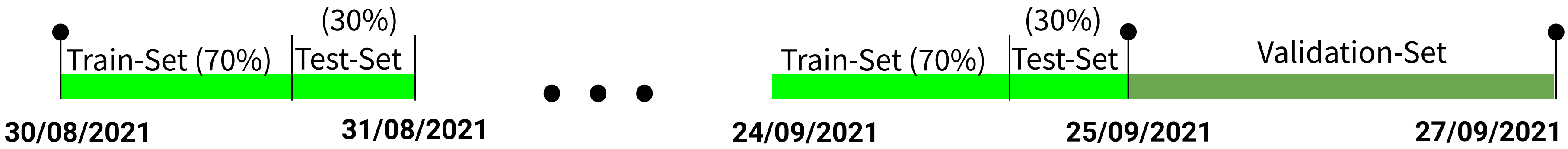
Padding Strategy	Model	Batch Size	Max Len	Test Accur.	GPU	Training Time per Epoch (mm:ss)
Smart Batching	BERT-base	16	400	0.935	Tesla K80	0:35:06
Fixed Padding	BERT-base	16	400	0.93	Tesla K80	0:53:14

Source: https://mccormickml.com/2020/07/29/smart-batching-tutorial/#s5-fine-tune-bert

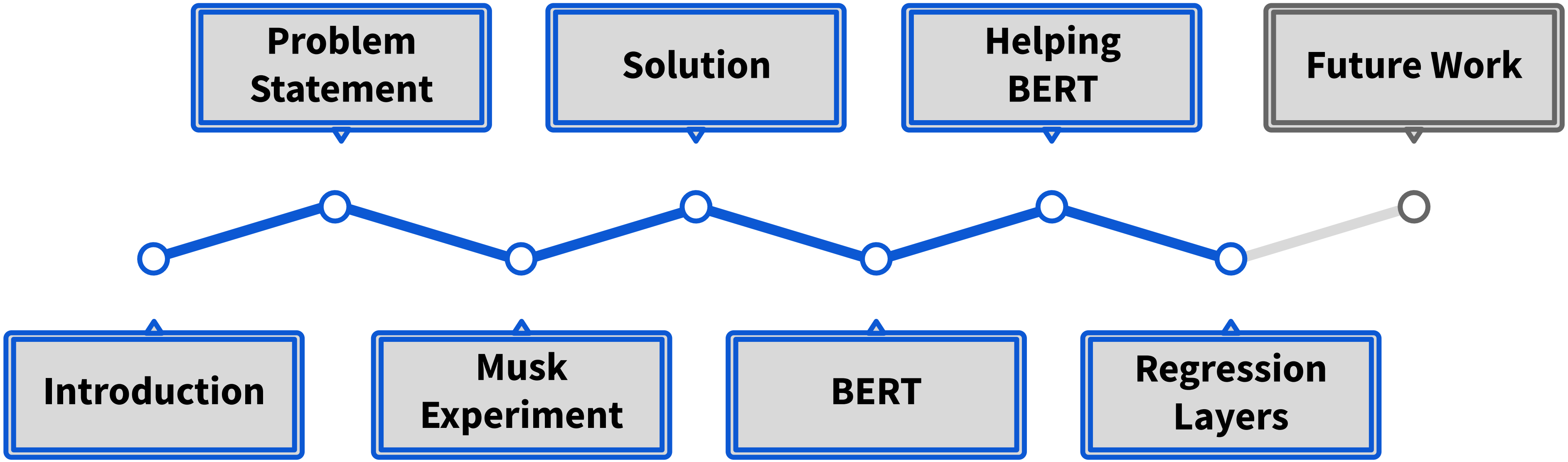
Our Data in Numbers

Coin	Tweets
ADA	59918
AVAX	18049
DOGE	35976
DOT	22158
ETH	291569
LTC	11856
SOL	183511
UNI	5695
XRP	20651

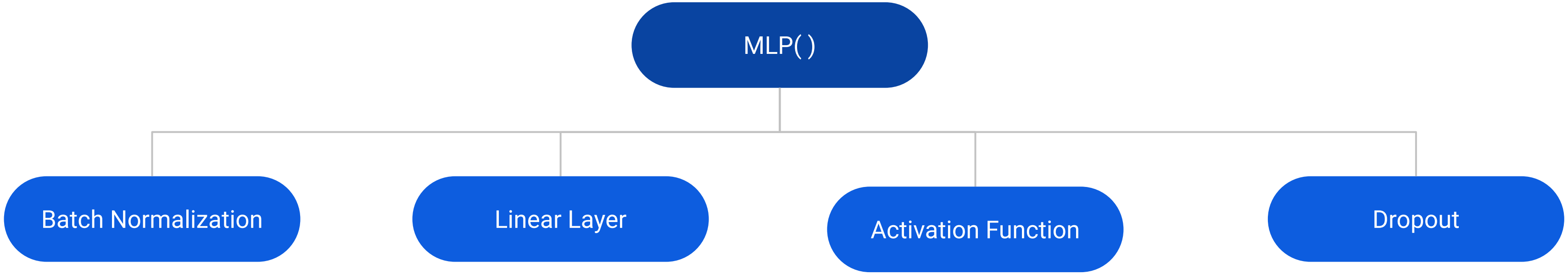
Train, Test, Validation Split



Data Batch Division


Problem Statement
Solution
Helping BERT
Future Work
Introduction
Musk Experiment
BERT
Regression Layers

Multi Layer Perceptron

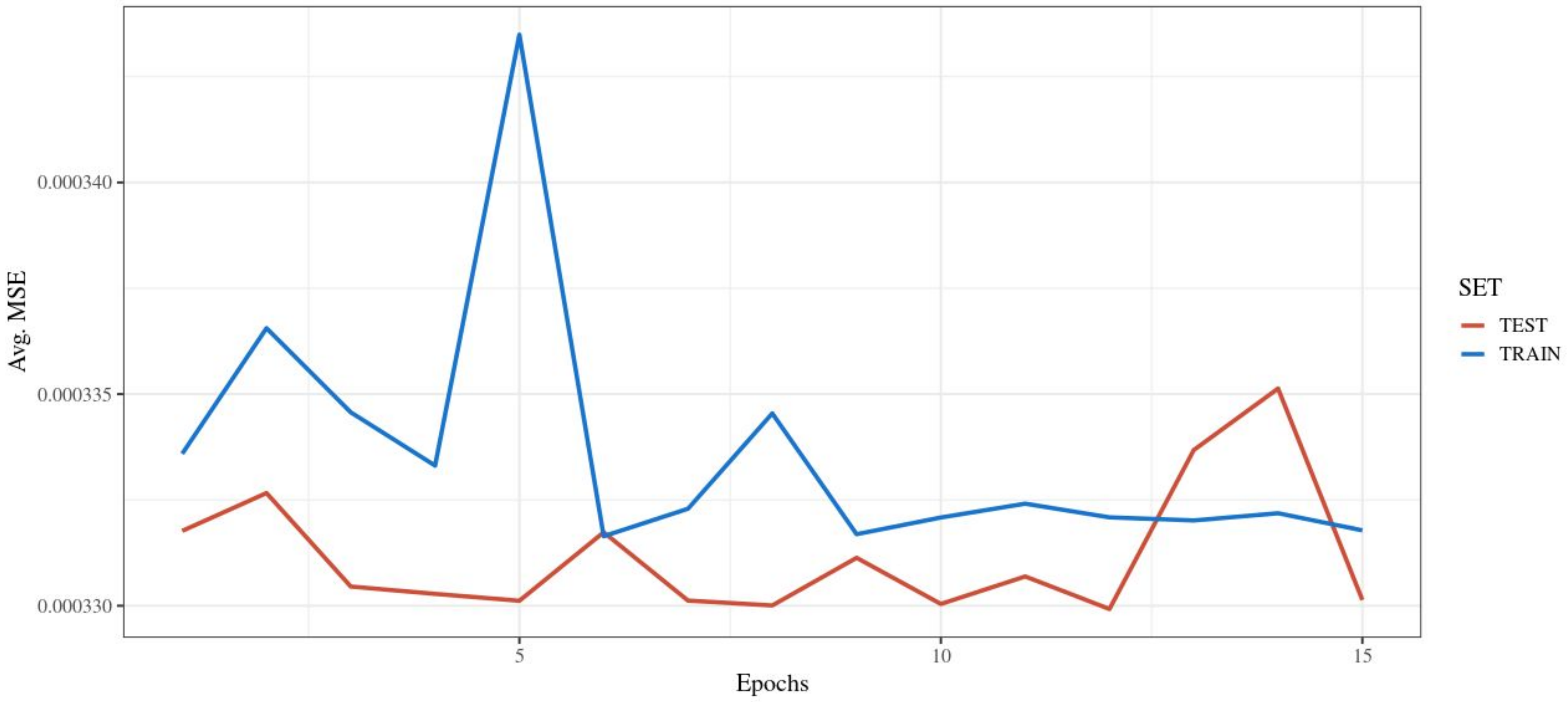


First Attempt

```python
hyperparams = pd.DataFrame(data={"n_in_1": [768, 768, 768],
                                 "n_out_1": [512, 151, 768],
                                 "batchnorm_1": [False, False, False],
                                 "activ_1": [nn.ReLU, nn.ReLU, nn.ReLU],
                                 "dropout_p_1": [None, None, None],

                                 "n_in_2": [512, 151, 768],
                                 "n_out_2": [341, 341, 512],
                                 "batchnorm_2": [True, True, True],
                                 "activ_2": [nn.LeakyReLU, nn.LeakyReLU, nn.LeakyReLU],
                                 "dropout_p_2": [0.4, 0.5, 0.6],

                                 "n_in_3": [341, 341, 512],
                                 "n_out_3": [227, 512, 341],
                                 "batchnorm_3": [True, True, True],
                                 "activ_3": [nn.LeakyReLU, nn.LeakyReLU, nn.LeakyReLU],
                                 "dropout_p_3": [0.4, 0.5, 0.6],

                                 "n_in_4": [227, 512, 341],
                                 "n_out_4": [151, 768, 227],
                                 "batchnorm_4": [True, True, True],
                                 "activ_4": [nn.LeakyReLU, nn.LeakyReLU, nn.LeakyReLU],
                                 "dropout_p_4": [0.4, 0.5, 0.6],

                                 "n_in_5": [151, 768, 227],
                                 "n_out_5": [1, 1, 1],
                                 "batchnorm_5": [True, True, True],
                                 "activ_5": [nn.LeakyReLU, nn.LeakyReLU, nn.LeakyReLU],
                                 "dropout_p_5": [None, None, None],

                                 "learning_rate": [0.0001, 0.001, 0.01]})
hyperparams
```

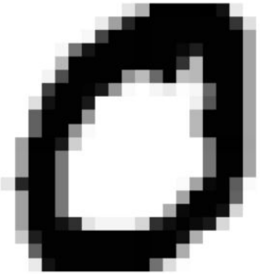
Training Parameters

Optimizer	→	ADAM
Loss Function	→	MSE
Batch Size	→	64
Early Stopping	→	patience=4

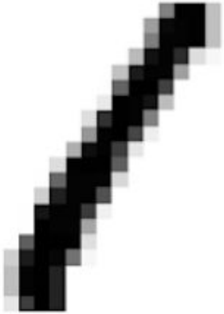

0.000340
0.000335
0.000330
Avg. MSE
5
10
15
Epochs
SET
TEST
TRAIN

Second Attempt

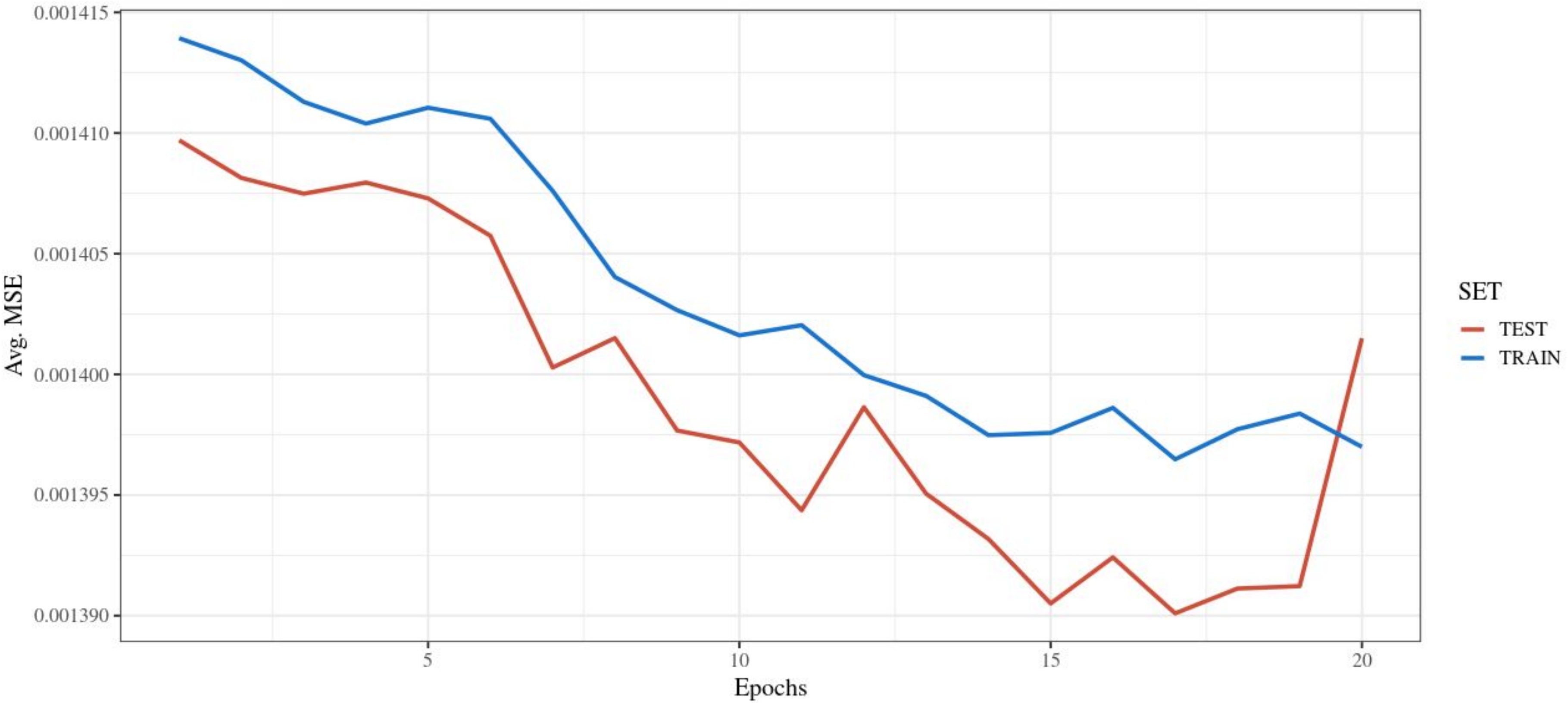
Michael Nielsen’s Tip

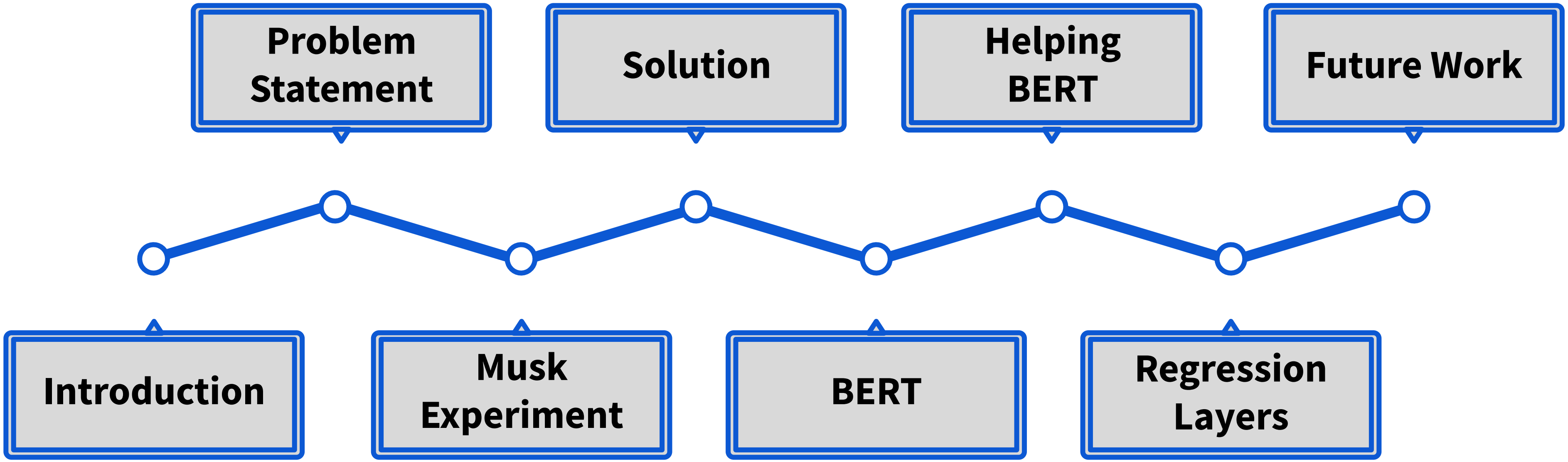


1st Quartile



4th Quartile


0.001415
0.001410
0.001405
0.001400
0.001395
0.001390
Avg. MSE
5
10
15
20
Epochs
SET
TEST
TRAIN


Introduction
Problem Statement
Musk Experiment
Solution
BERT
Helping BERT
Regression Layers
Future Work

Future Work: No Results is a Result :)

- No Correlation Discovered:
    - Propose a different preprocessing or work on a pre-trained model on Tweets.

- Handle Twitter Bots (~80%):
    - -- Krypto-Invaders - -\n\n💸 - from 0.002 $ETH\n👾 - only 300. Ever!! - 1:1 NFTs\n🕹️ - Playable Game \n\nCollect here:\n🌐 👉 https://t.co/HcGFC6euWB\n\n#pixelnft #galxy #NFTcollectibles #nftgaming #nftgame https://t.co/LyvZQ79HgE

- Try on a different lag

References

[1]Jacob Devlin Ming-Wei Chang Kenton Lee Kristina Toutanova Google AI Language. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding
[2] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In Advances in Neural Information Processing Systems, pages 6000–6010.
[3]Stephane Lathuili ´ ere, Pablo Mesejo, Xavier Alameda-Pineda, ` Member IEEE, and Radu Horaud. 2020. A Comprehensive Analysis of Deep Regression
[4]Marco Pota, Mirko Ventura, Rosario Catelli and Massimo Esposi. 2021. An Effective BERT-Based Pipeline for Twitter Sentiment Analysis: A Case Study in Italian
[5]Kostadin Yotov, Emil Hadzhikolev, Stanka Hadzhikoleva. 2020. Determining the Number of Neurons in Artificial Neural Networks for Approximation, Trained with Algorithms Using the Jacobi Matrix
[6]STEFANO CRESCI, FABRIZIO LILLO, DANIELE REGOLI, SERENA TARDELLI and MAURIZIO TESCONI. 2018. Cashtag piggybacking: uncovering spam and bot activity in stock microblogs on Twi□tter
[7]Chris McCormick, Smart Batching Tutorial - Speed Up BERT Training:
https://mccormickml.com/2020/07/29/smart-batching-tutorial/#s5-fine-tune-bert
[8] Anthony Galtier. 2021. Fine-tuning BERT for a regression task: is a description enough to predict a property's list price?
[9] Chris McCormick YouTube channel: https://www.youtube.com/c/ChrisMcCormickAI/videos
[10] Hedu Math of Intelligence YouTube channel: https://www.youtube.com/c/HeduMathematicsofIntelligence/videos
[11] Neural Network and Deep Learning, free online book: http://neuralnetworksanddeeplearning.com/index.html