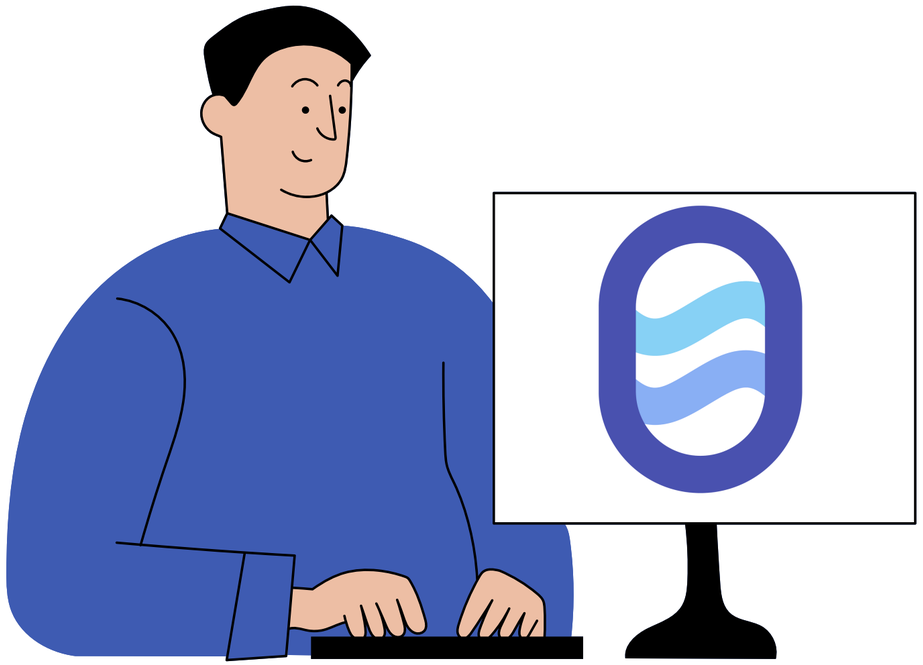
SENIOR PROJECT III

Opinion Zero

By: Maylin Catherine Cerf 6180039
Pith Laohavirojana 6180048

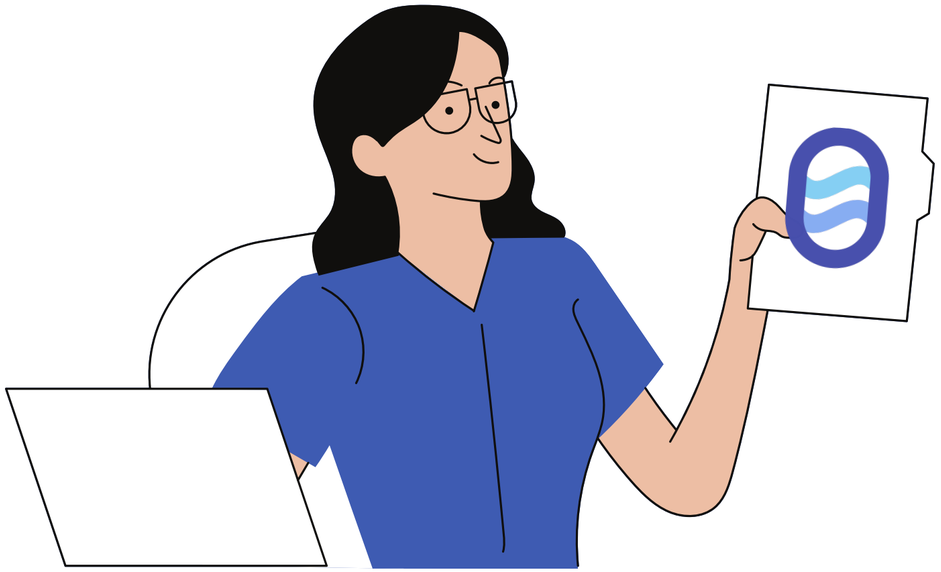
TABLE OF CONTENTS



Overview of our Application

Main Issue is Media Bias.

Overview of our Application



01 Detection criteria

Yellow

Neutral

Red

02 Giving an outcome Identifying as red shirt favour or yellow shirt favour or neutral

03 News will be accessible and displayed on the feed to the users.

04 Two important aspects:

- Mobile Application
- Bias Detection model

Data Collection

- 669 datasets
  - Red shirt: 201
  - Yellow shirt: 202
  - Neutral: 266

Text Classification Models

Recap: Previous Methods

01 Frequency-inverse document frequency

02 Word2Vec extension. Pretrained embedding layer.

03 Bidirectional Encoder Representations from Transformers

Percent Precision

58.16%

48.06%

80.30%

Issues of Previous Methods

TF-IDF

- Predict **correct as much as wrong**

```
[ ]  from sklearn.metrics import confusion_matrix
     confusion_matrix(y_true, y_pred)

     array([[61,  5, 19],
            [15,  6,  8],
            [ 9,  4, 15]])
```

FastText

- Predict **toward neutral**

```
from sklearn.metrics import confusion_mat
confusion_matrix(y_true, y_pred)

array([[83,  0,  2],
       [28,  0,  1],
       [24,  0,  4]])
```

BERT

- Best out of all, but **can improve**

```
from sklearn.metrics import confusion_matrix
confusion_matrix(y_true, y_pred)

array([[83,  0,  2],
       [15, 14,  0],
       [17,  0, 11]])
```

Other Issues...

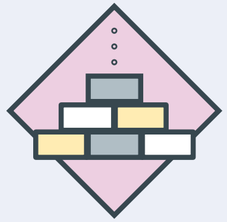
01. 473 is dataset is not Enough

02. More variety in learning, model layer and epoch

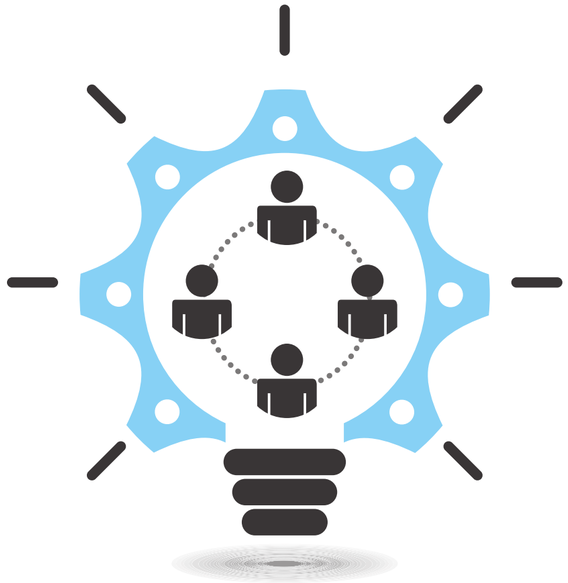
03. **Red**: 101 **Yellow**: 105 **Neutral**: 267

Not Enough Data

Neutral Class Domination



Model Structure



Solutions!

Let's begin.

Our Solutions

01 Collect more data and Rebalance the data

- 669 datasets to 599 datasets

202 news

196 news

201 news

02 Change Tokenizer

Monsoon-nlp /bert-base-thai → airesearch/wangchanberta-base-att-spm-uncased

03 Fine Tuning Bert with 4 Different Methods

Chosen Technique for Bert Fine tuning

Freeze Bert then attach layers

- all of the BERT layers are **frozen**, then the the retrain **process occurs by attaching one or more network layers** after the pre-trained BERT.
- the weight of the model will only be modified in the attached layers.

Fine Tuning Bert with 4 Different Methods

1. BERT with

Soft Max

2. BERT with

tanh

3. BERT with

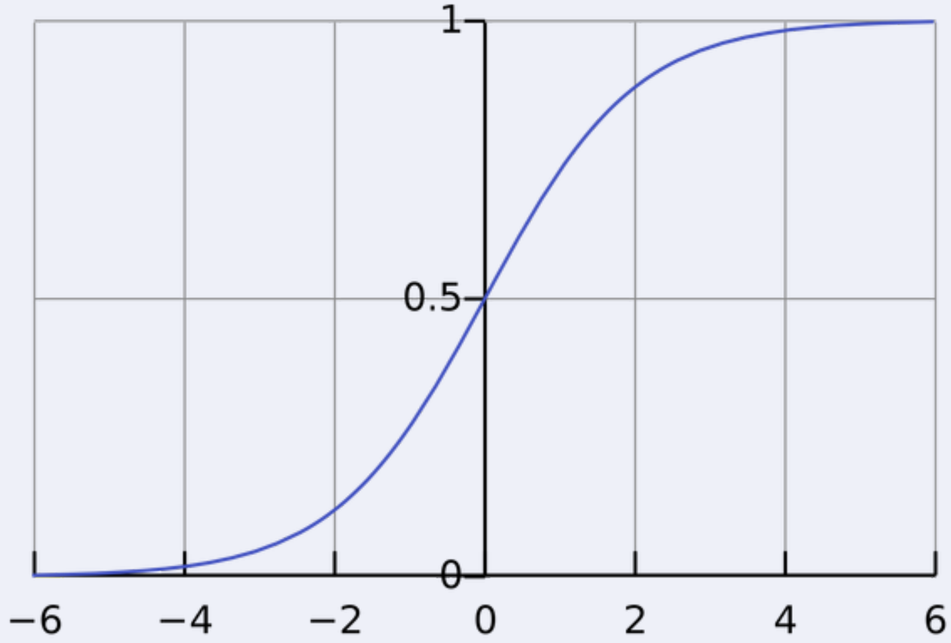
TRIPLET

4. BERT with

SIAMESE

BERT with Soft Max

What is SoftMax ?



01 Accepts the input value as a vector of k real values and convert it into a vector of k real values that add up to 1.

02 Output vector value can be explain into a probabilities, since Softmax Function alter it into a values between 0 and

Our Model with SoftMax

- Apply Softmax function as the last layer of activation function

- Return the probabilities of each possible class which are

Yellow

Neutral

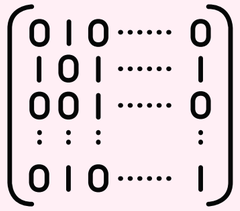
Red

Structure of Bert with Softmax

airesearch/wangchanberta-base-att-spm-uncased

```
{'input_ids': [5, 10, 17332, 41,
'attention_mask': [1, 1, 1, 1, 1, 1]}
```

PRETRAINED BERT



linear layers

Softmax

[0.8,0.5, 0.4]

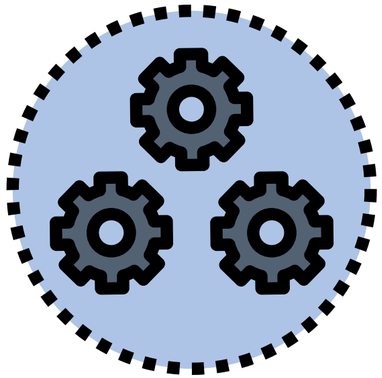
Neutral

BERT with tanh

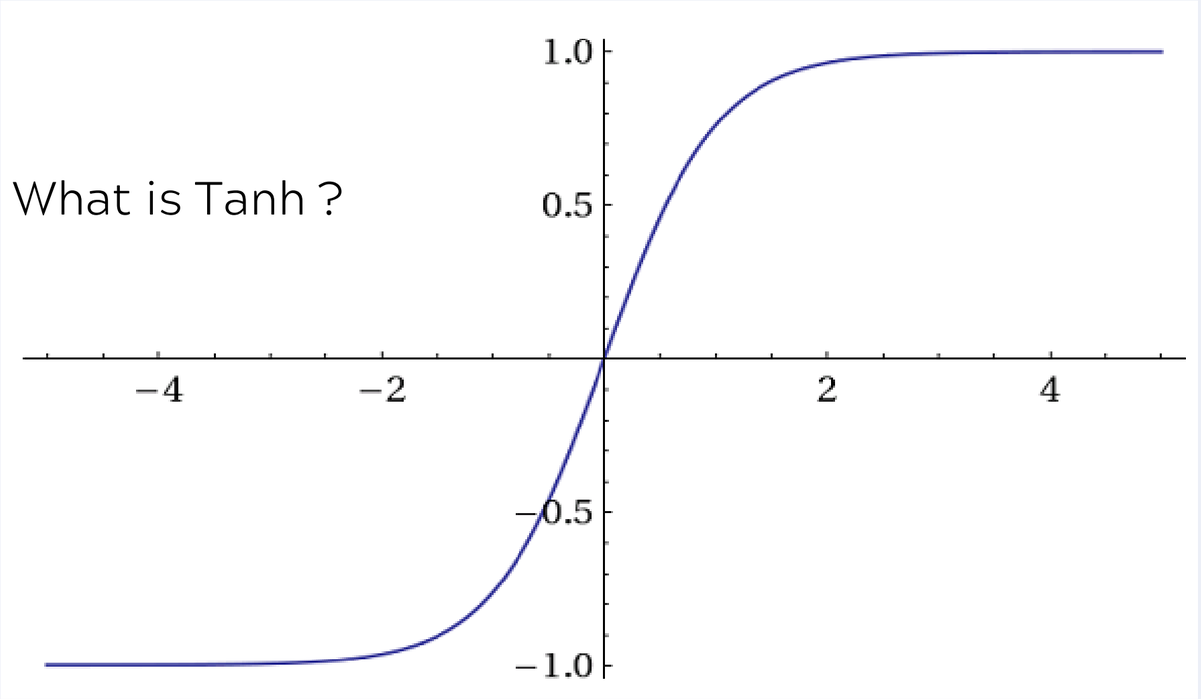
01 Accepts any real value as input and returns values ranging from -1 to 1

02 As the input value is larger, output will approach 1.

As the input value is small the output will approach -1.



What is Tanh Activation Function?

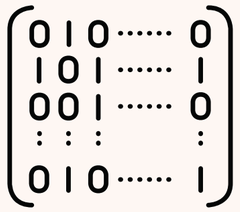


Structure of Bert with tanh

airesearch/wangchanberta-base-att-spm-uncased

```
{'input_ids': [5, 10, 17332, 41,
'attention_mask': [1, 1, 1, 1, 1, 1]}
```

PRETRAINED BERT



linear layers

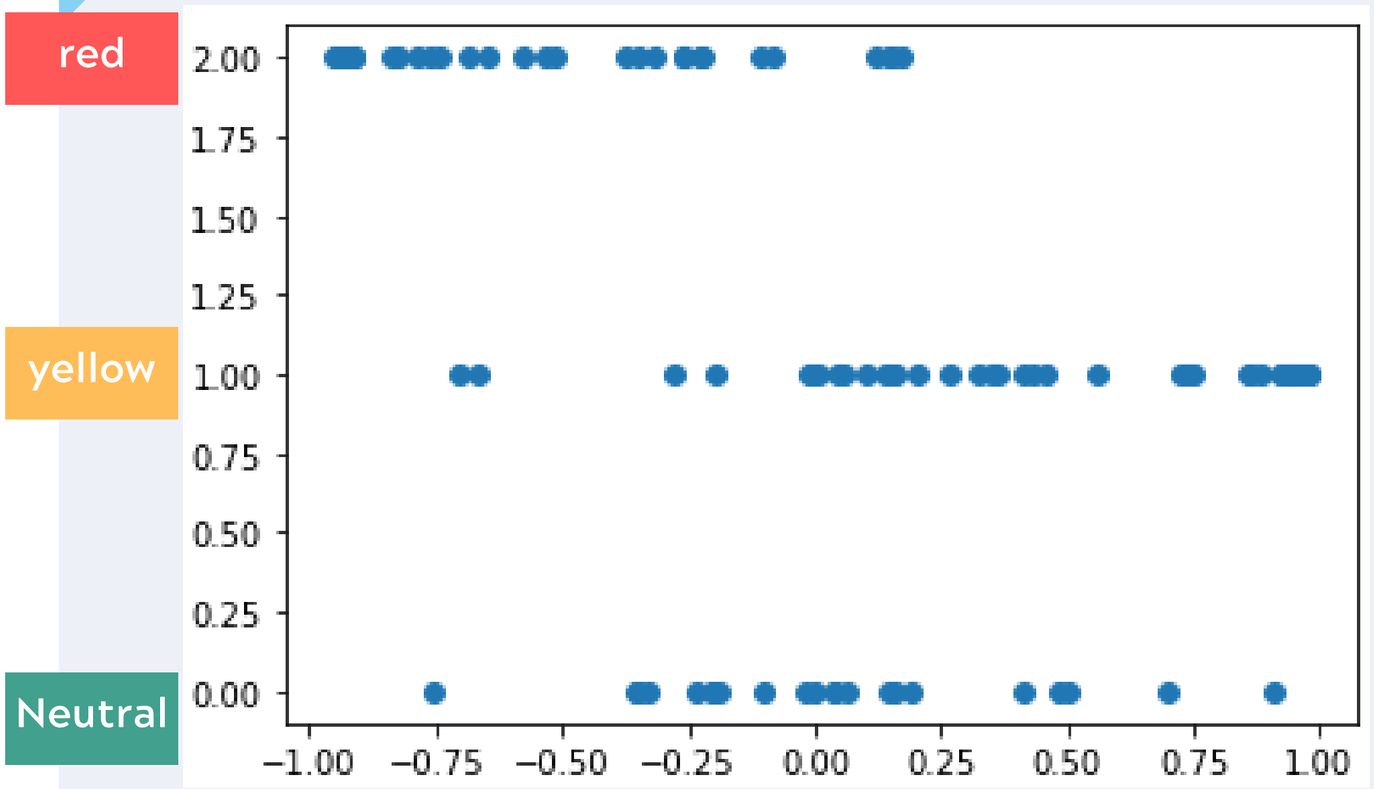
tanh

0.563

?

Is this red or neutral??

Why is *tanh* activation function doesn't work?



01 Tanh activation returns values ranging from -1 to 1 and while prediction can only be three classes.

02 Plotting return ranging output from -1 to 1 and try to find the possible clear cut for each cluster

03 Data points overlapping each other, impossible to find the clear cut for each clutter to determine what value belong to each of the classe

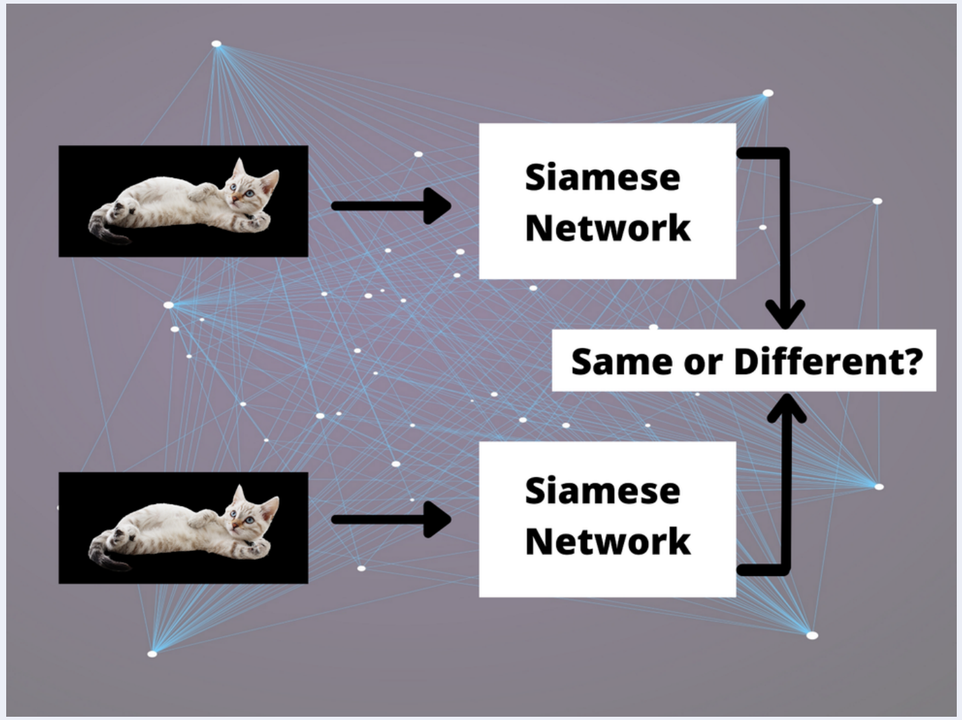
BERT with SIAMESE

01 Take a pair of Input and inside the network it consists of one or more identical networks.

02 Each model are required to have the same parameters as well as weight.

02 In each network, Euclidean distance of two features is calculated to identify the similarity of both features.

What is Siamese Network?



Our model with SIAMESE

Purpose 🎯

- Is not to find the similarity or difference between two inputs like Siamese model.

01 The two input are not the same kind.

02 The attached layer are two-headed alike pattern

03 Receive **title** as another parameter

TILTLE

Why TITLE is important?

01 We receive title as another parameter is because when the tokenizer Bert that being used, airesearch/wangchanberta-base-att-spm-uncased demand max-length parameter.

02 This parameter **limits** the length of each sequence up to only 416 tokens

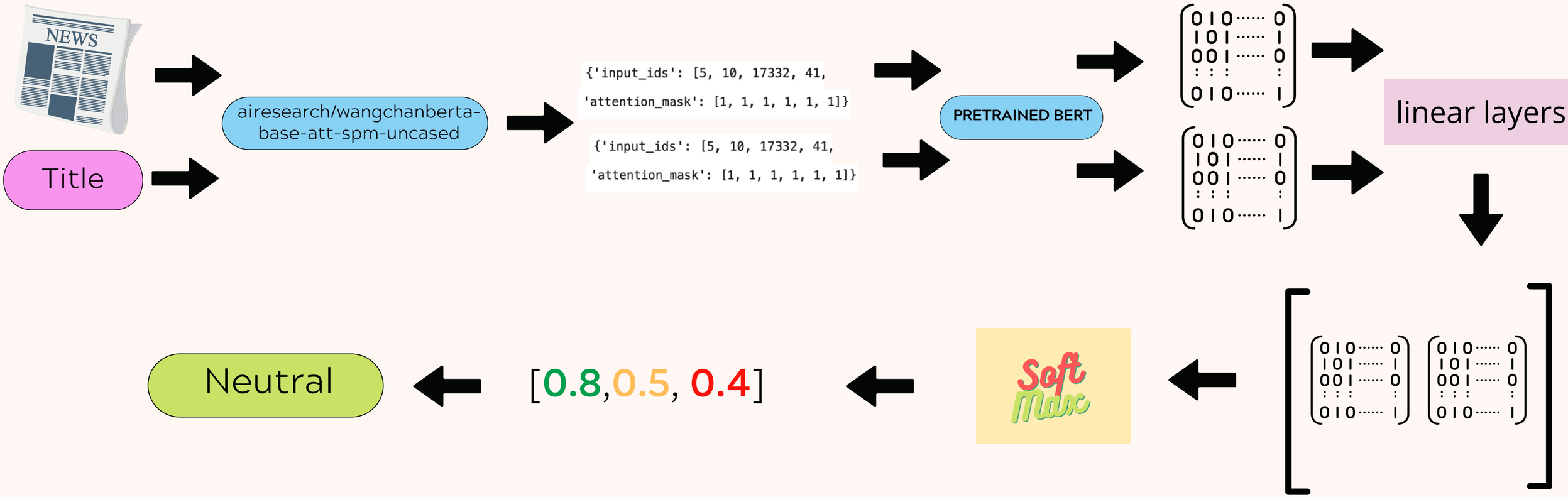
```
self.tokenizer(article, return_tensors="pt",truncation=True,padding=True, max_length=416)
```

03 Not sufficient to represent all the content in the article

04 Significant part of the news may be left out

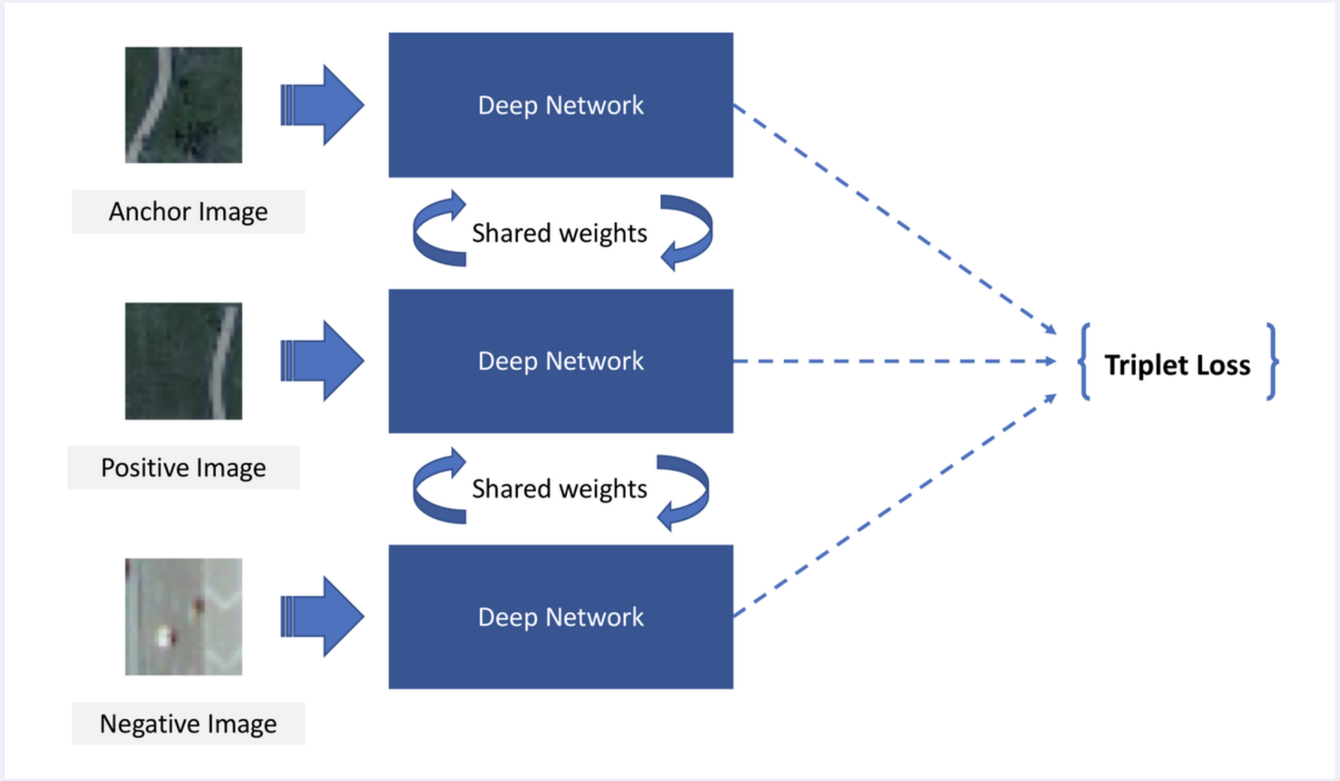
05 Title of the news usually sum up the main idea, lead to more accurate result

Structure of Bert with SIAMESE



Bert with TRIPLET

Structure of Triplet Network



01 What is Triplet Network?

- to learn useful representations by **distance comparisons t**o use in image classification tasks.

02 Key Features of Triplet Network

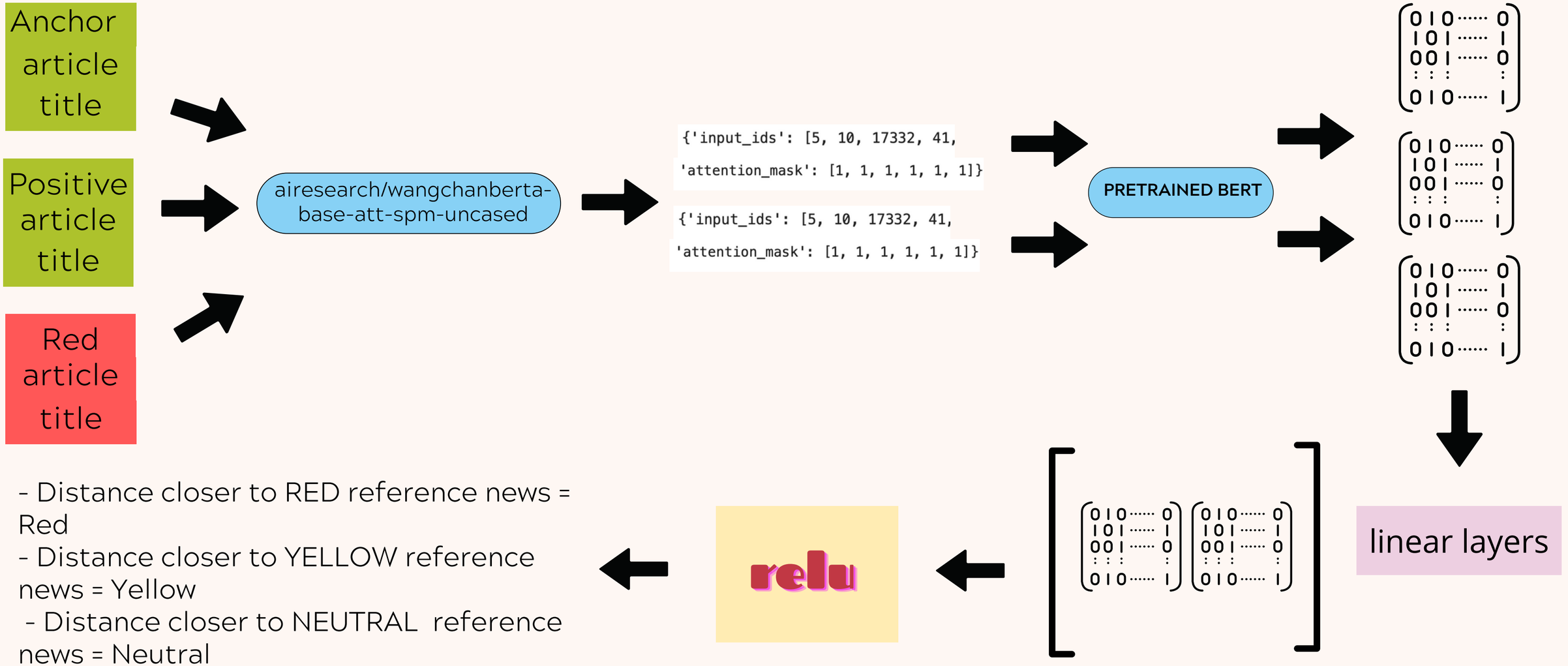
- 3 inputs parameter

Anchor (X)

Positve (X+)

Negative (X-)

Structure of Bert with TRIPLET



Result

Let's begin.

Result of BERT with Softmax

01 The performance of BERT with Softmax is moderately good.

Accuracy
77.77%

Recall
77.77%

Precision
77.81%

02 It performs best on speculating neutral class where it predict 10 wrong in the total of 53.

03 accomplish the same outcome when predicting yellow and red classes.

	Neutral	Yellow	Red
Neutral	43	3	7
Yellow	5	47	10
Red	5	10	50

Result of BERT with tanh

It is impossible to interpret result !!!

01 Can't find the exact threshold determine what value belong to each of the classes.

02 Interpretation is not executable.

Result of BERT with SIAMESE

01 The performance of BERT with Siamese is highly acceptable.

Accuracy
82.58 %

Recall
82.58%

Precision
83.03 %

02 It performs best on speculating red class where it predict 5 wrong in the total of 59.

03 Accomplish the same outcome when predicting yellow and neutral classes.

	Neutral	Yellow	Red
Neutral	41	6	7
Yellow	5	52	8
Red	1	4	54

Result of BERT with TRIPLET

01 The performance of BERT model with Triplet Network showed an appealing outcome

Accuracy
80.89 %

Recall
80.89%

Precision
82.04 %

02 Exceptionally predicts well on the neutral class where it predicts 48 correct out of 54

03 Does not perform well on predicting yellow, speculates 43 corrects and 22 wrongs. While preferable predict well on red, where it is 6 incorrect out of 59.

	Neutral	Yellow	Red
Neutral	48	2	4
Yellow	12	43	10
Red	3	3	53

Result Comparison

Let's begin.

Performance of BERT with Soft Max

01 The model performance is better than Model with only TF-IDF and Fasttex.

Precision
58.16%

Precision
48.06 %

<

Precision
77.81 %

02 The model outcome is beaten by BERT, Siamese Model as well as Triplet Network

Precision
80.30%

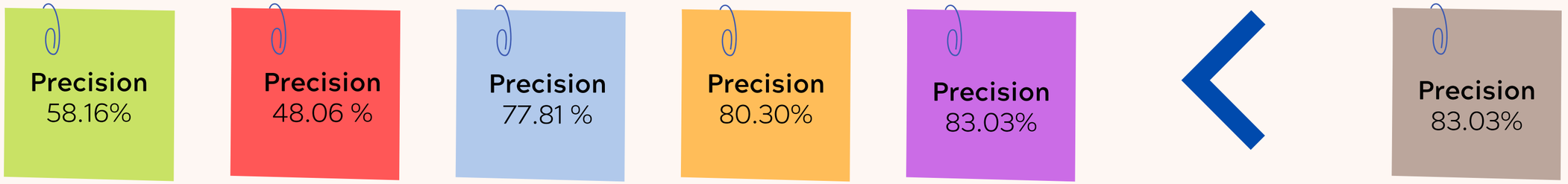
Precision
83.03%

Precision
83.03%

03 This model is not the final model

Performance of BERT with SIAMESE

01 It is the highest precision percentage out of all model that we have performed



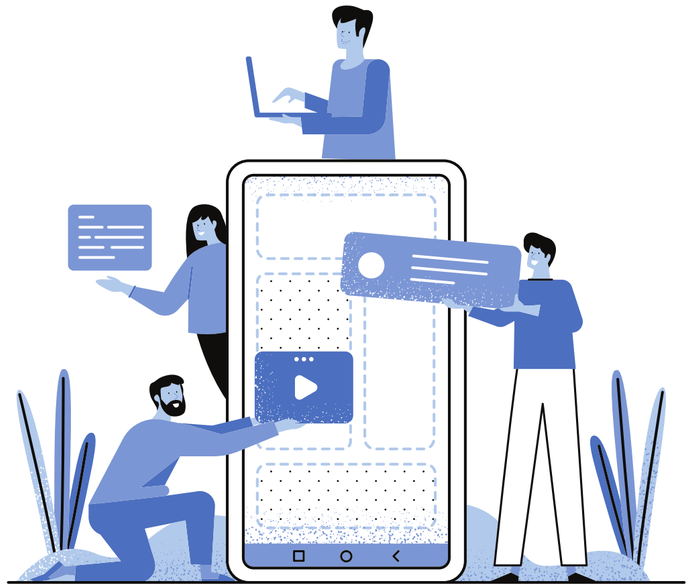
Performance of BERT with TRIPLET

01 Although the overall metrics score of those classifier yield one best result and one moderate result.

02 The prediction result is calculated by looking at the difference of distances, it is impractical to return the percentage of what is the likely chance this news article belong in each classes.

The Final Model is

BERT with SIAMESE



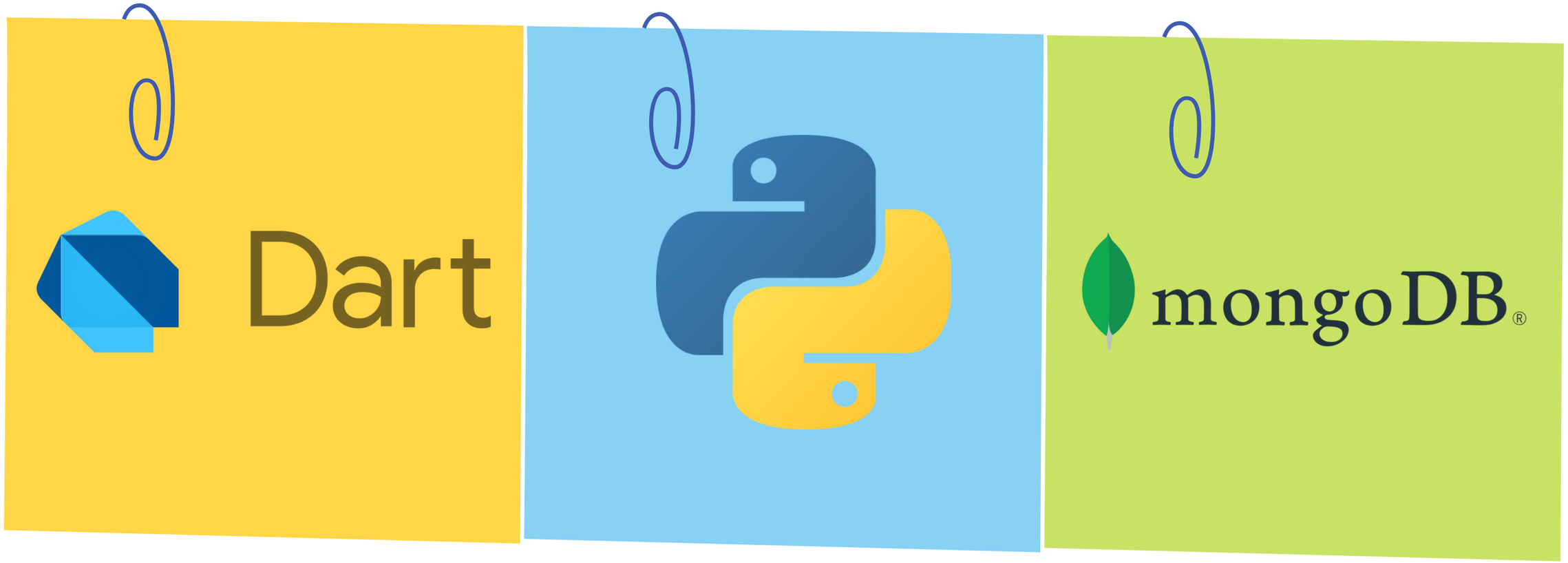
Mobile Application

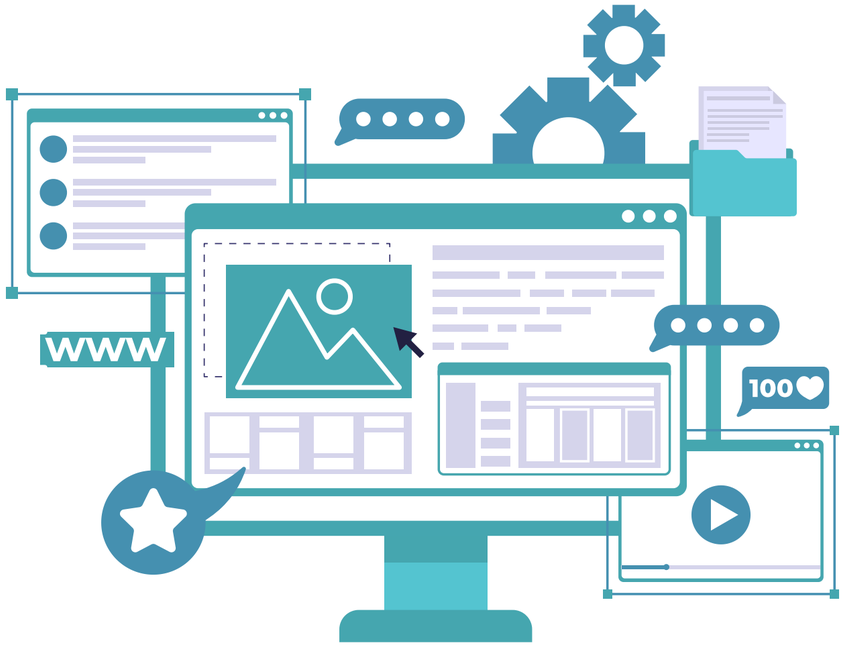
Let's begin.

Recap: Previous Methods

- Ionic and with the help of vue.js
- app development platform builds a cross-platform mobile application

What we actually used...





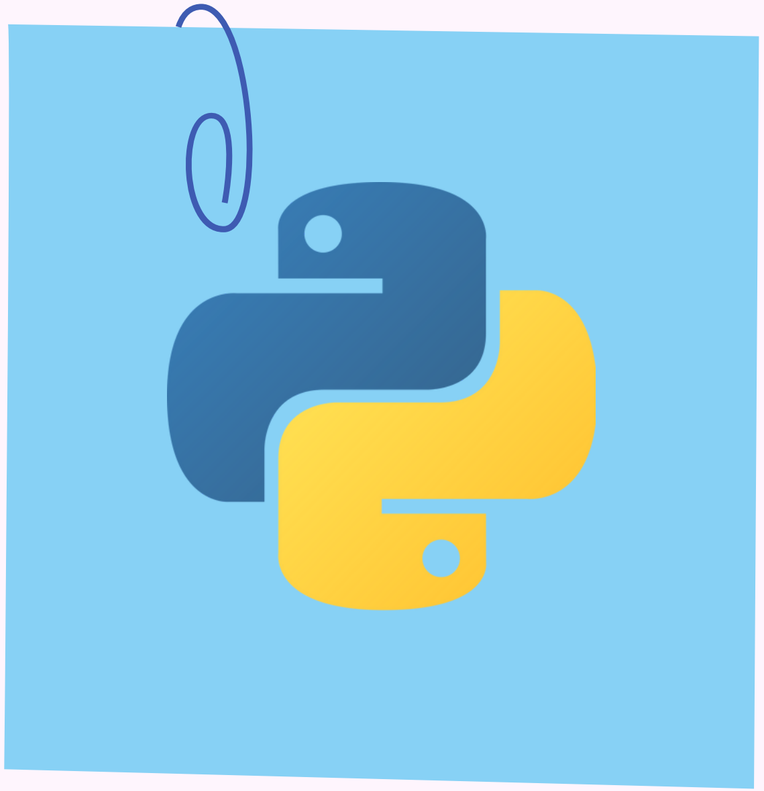
Backend

Let's begin.

Methods of Backend

Web Sever

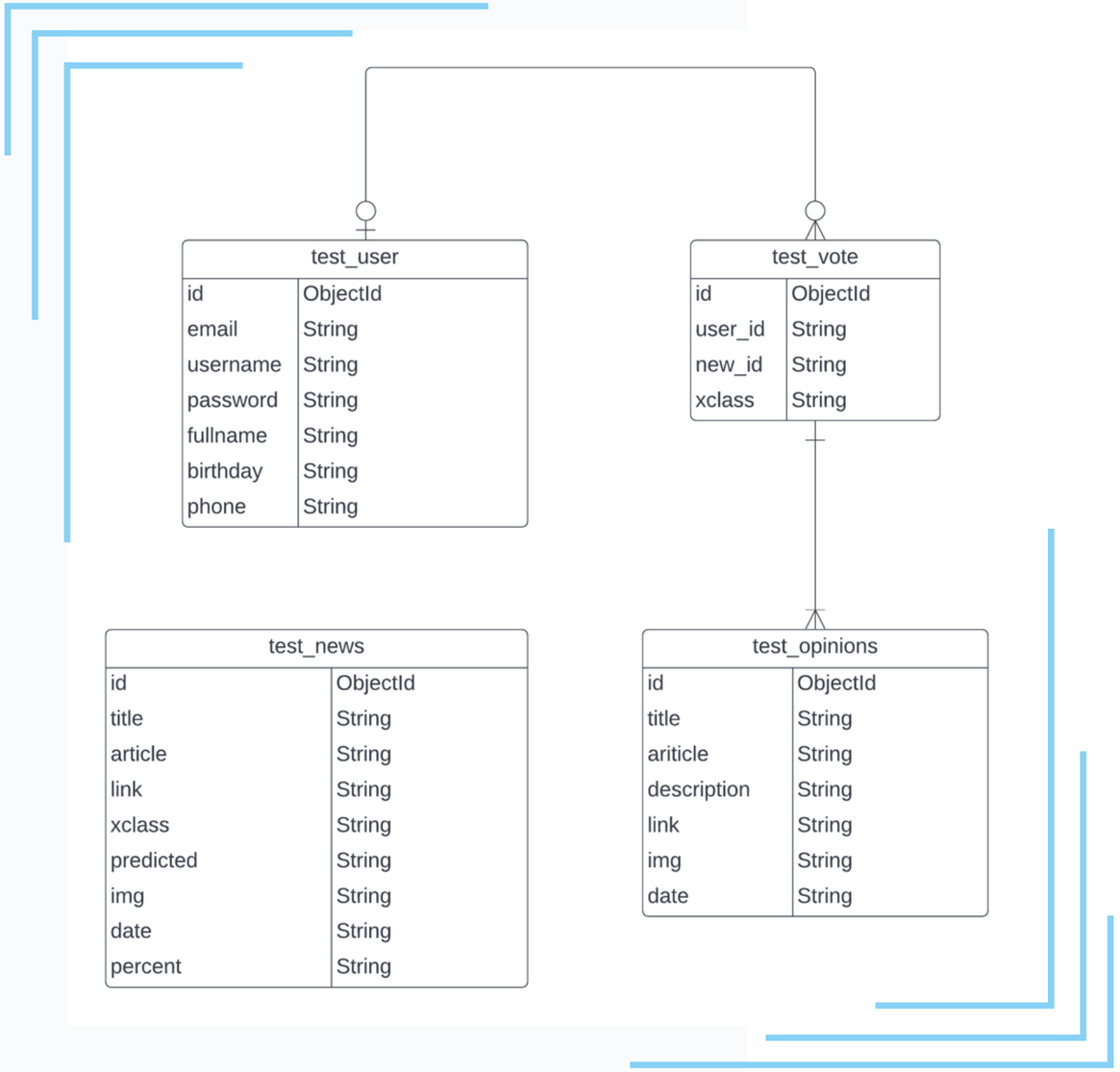
Python



API

FastAPI

Database: ER Diagram



- test_user and test_vote
  - (optional) one to (optional) many
- test_vote and test_user
  - (optional) many to (optional) one
- test_vote and test_opinions
  - one to many
- test_opinions and test_vote
  - many to one

APIS

users

PUT `/users/add` Adduser

auth

PUT `/auth/` Validator

articles

PUT `/articles/retrieve` Retrievenews

opinion

PUT `/opinion/retrieve` Retrieveopinionnews

news

PUT `/news/addopinion` Addopinion

Demo of Finished Prototype

That's a wrap!

Thank you for participating.