野村金融科技題目(一)

第一組

指導Mentor：黃俊哲(Tony)/陳景堯(Michael)

指導老師：石百達 老師　　張智星 老師

組長:余柏叡(台大財金碩一)

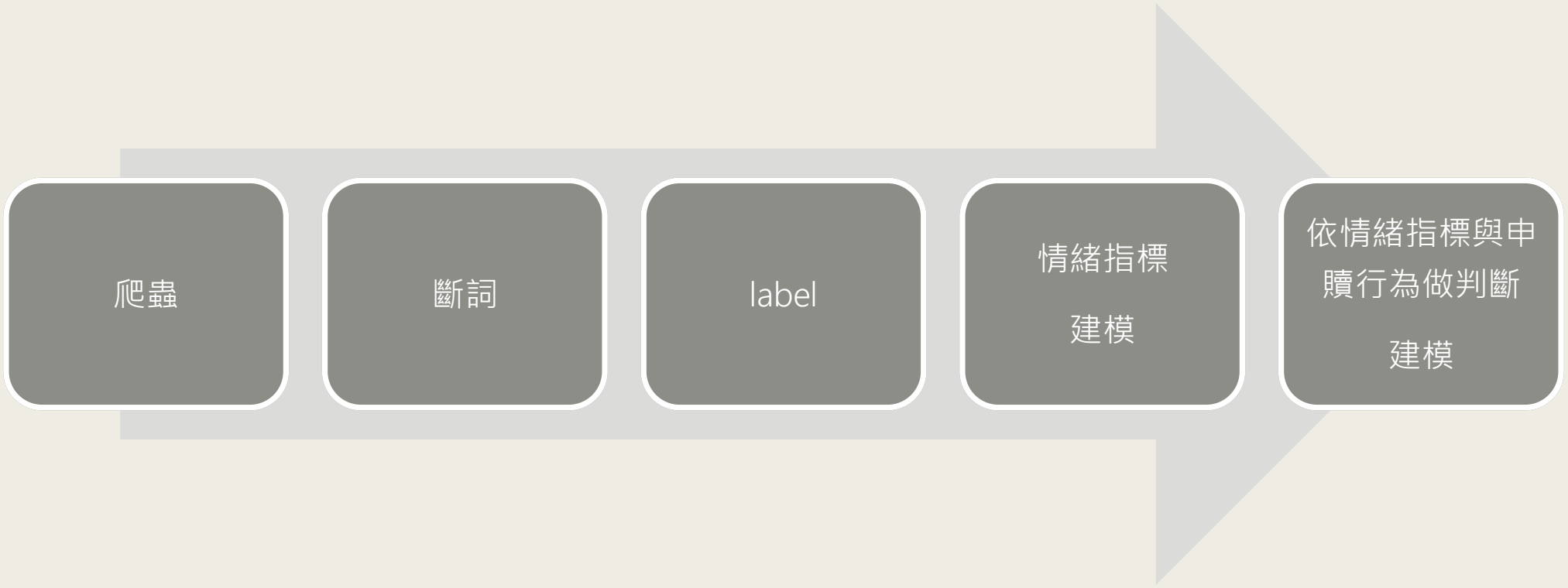
組員:趙翊茹(台大財金碩二)

張庭寧(台大工科碩一)

林庭樂(政大統計碩一)

黃俊穎(台大電子碩一)

專案流程



爬蟲

總數:約20-30萬筆(未過濾)

- PTT—Stock版(2007~)
- Mobile01(2006~)
- 財經M平方(2015~)
- Fundhot強基金論壇
- FB債券基金研究社
- FB綠角財經筆記
- FB李其展的外匯交易致勝兵法(2010~)

斷詞

✓ .Jieba(自行建立字典)

資料量較多

財經用詞較多

斷詞速度較快

X .CKIP

一般用詞準確度較高但財經用詞略少

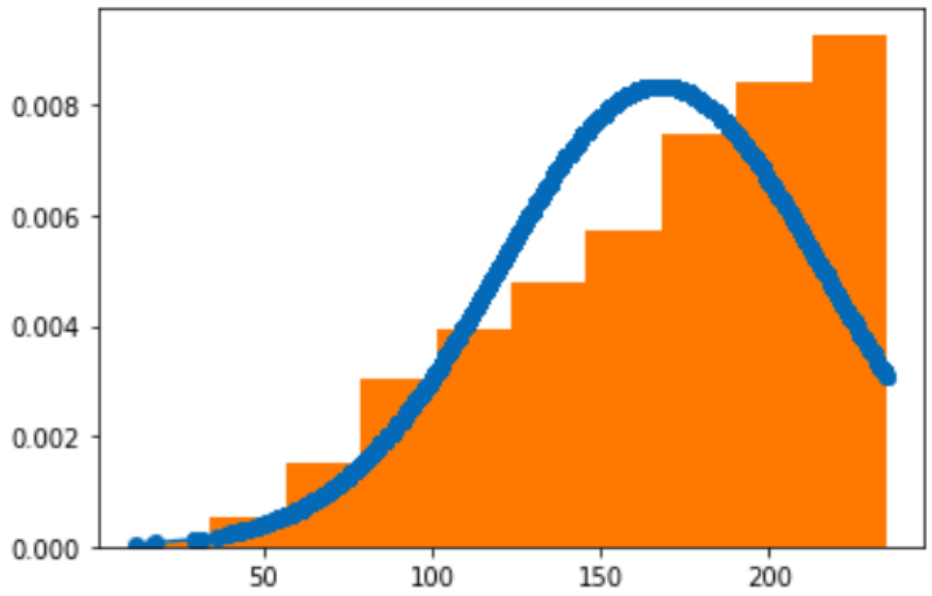
所需時間約為Jieba的8倍

情緒指標

- 資料來源：Ptt(2014-2020)+各式財經新聞內容
- Label_Pos:15032
- Label_Neg:5207

```
Max length is:  5161
30% cover length up to:  214
```

```
C:\Users\user\anaconda3\lib\site-packages\ipykernel_la
The 'normed' kwarg was deprecated in Matplotlib 2.1 ar
  if sys.path[0] == '':
```



文本字數長度分配圖

註：因人力有限使得Ptt內文標記數不夠大，遂尋找github上已標註完整之財經新聞文本

正向資料詞頻統計

```
In [12]: word_counts1

Out[12]: {'多': 467,
          '機制': 432,
          '年': 426,
          '分類': 397,
          '月': 392,
          '標的': 390,
          '退場': 384,
          '分析': 367,
          '進': 348,
          '買': 309,
          '台灣': 293,
          '公司': 277,
          '正文': 272,
          '營收': 271,
          '長期投資': 259,
          '股價': 238,
          '沒有': 236,
          '高': 216,
          '進場': 216,
          '說': 215,
          '停損': 212,
```

[多]一詞 正負向資料t-test (顯著不同)

```
In [17]: stats.ttest_ind(wordsfreq3,wordsfreq2)

Out[17]: Ttest_indResult(statistic=-5.832553912089536, pvalue=6.5479011073447766e-09)
```

負向資料詞頻統計

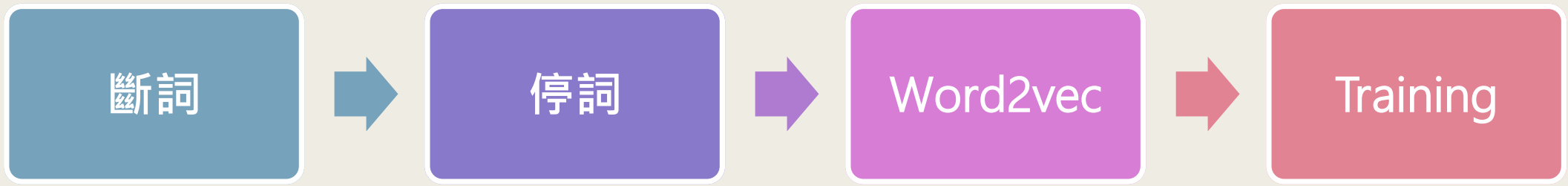
```
In [13]: words_counts2 = input_file_to_dict(words2)
         words_counts2
```

```
Out[13]: {'中國': 454,
          '美國': 443,
          '月': 371,
          '年': 300,
          '說': 282,
          '沒有': 246,
          '公司': 234,
          '多': 228,
          '台灣': 209,
          '市場': 207,
          '好': 191,
          '機制': 176,
          '沒': 174,
          '分類': 164,
          '分析': 163,
          '買': 163,
          '川普': 162,
          '日': 160,
          '股票': 159,
          '標的': 157,
          '華為': 153,
          '空': 146,
```

[空]一詞 正負向資料t-test(顯著不同)

```
In [20]: stats.ttest_ind(wordsfreq3,wordsfreq2)
```

```
Out[20]: Ttest_indResult(statistic=5.667026226950225, pvalue=1.7098837129127305e-08)
```


斷詞
停詞
Word2vec
Training

轉詞向量Word2vec

```
In [37]: sentence = word2vec.Text8Corpus("segmentation.txt")
```

```
In [88]: #model = word2vec.Word2Vec.load_word2vec_format("test200.model.bin")
         model = word2vec.Word2Vec(sentence, size = 300, window = 10, min_count = 5, workers = 4, sg = 1)
```

維度：300
窗口：10
最小長度：5
算法：skip-gram

```
In [95]: model.word_vec('買')
```

```
Out[95]: array([ 9.78308991e-02,  2.30139475e-02, -9.93087217e-02,  1.57668591e-01,
                -3.38991024e-02,  2.36200467e-02,  1.03772134e-01, -8.26957673e-02,
                -1.22993216e-02,  5.66803552e-02, -1.30692929e-01,  3.92189845e-02,
                -4.81891818e-02,  1.66966408e-01,  2.03718901e-01,  3.88465784e-02,
                 3.32480147e-02,  9.72835161e-03, -6.60578981e-02,  7.60258213e-02,
                -7.25131556e-02,  1.03556380e-01, -3.81813794e-02, -4.16989997e-02,
                 5.51574044e-02, -3.94193595e-03, -1.59075614e-02,  6.67173490e-02,
                 1.12929821e-01, -6.68295547e-02, -3.80626991e-02, -1.30721748e-01,
                -8.68625417e-02, -5.25530465e-02,  2.28675529e-02, -1.13974065e-02,
                -2.98771705e-03, -2.88649611e-02,  4.79868054e-02, -6.11756183e-02,
                -3.63447741e-02, -2.83617284e-02, -1.05639391e-01, -8.15387890e-02,
                 2.64951009e-02,  6.12511002e-02, -6.57710433e-02, -5.01344390e-02,
                 4.46354151e-02, -3.30061056e-02, -8.01272914e-02,  7.99070522e-02,
                 2.06784382e-02, -1.96486875e-01, -1.63560209e-04,  3.64097916e-02,
                -4.32146825e-02, -1.41897738e-01,  1.13388784e-01, -9.80242863e-02,
                -1.97601952e-02, -6.74527809e-02,  1.24254301e-02,  8.31347611e-03,
                -7.42098317e-02,  4.00250480e-02, -3.76466960e-02,  1.32907286e-01,
                 1.10818081e-01, -1.37906354e-02,  1.72840729e-01, -1.45135168e-02,
                -1.04816079e-01, -8.77505243e-02, -8.56106693e-04,  8.94538909e-02,
                 1.33312434e-01,  1.58688053e-01, -7.72194490e-02, -6.33240566e-02,
                -1.83733627e-02,  7.89523199e-02,  3.22571248e-02,  7.22382963e-02,
                 1.19852088e-01,  4.46136408e-02,  4.68834303e-02,  7.50342160e-02
```

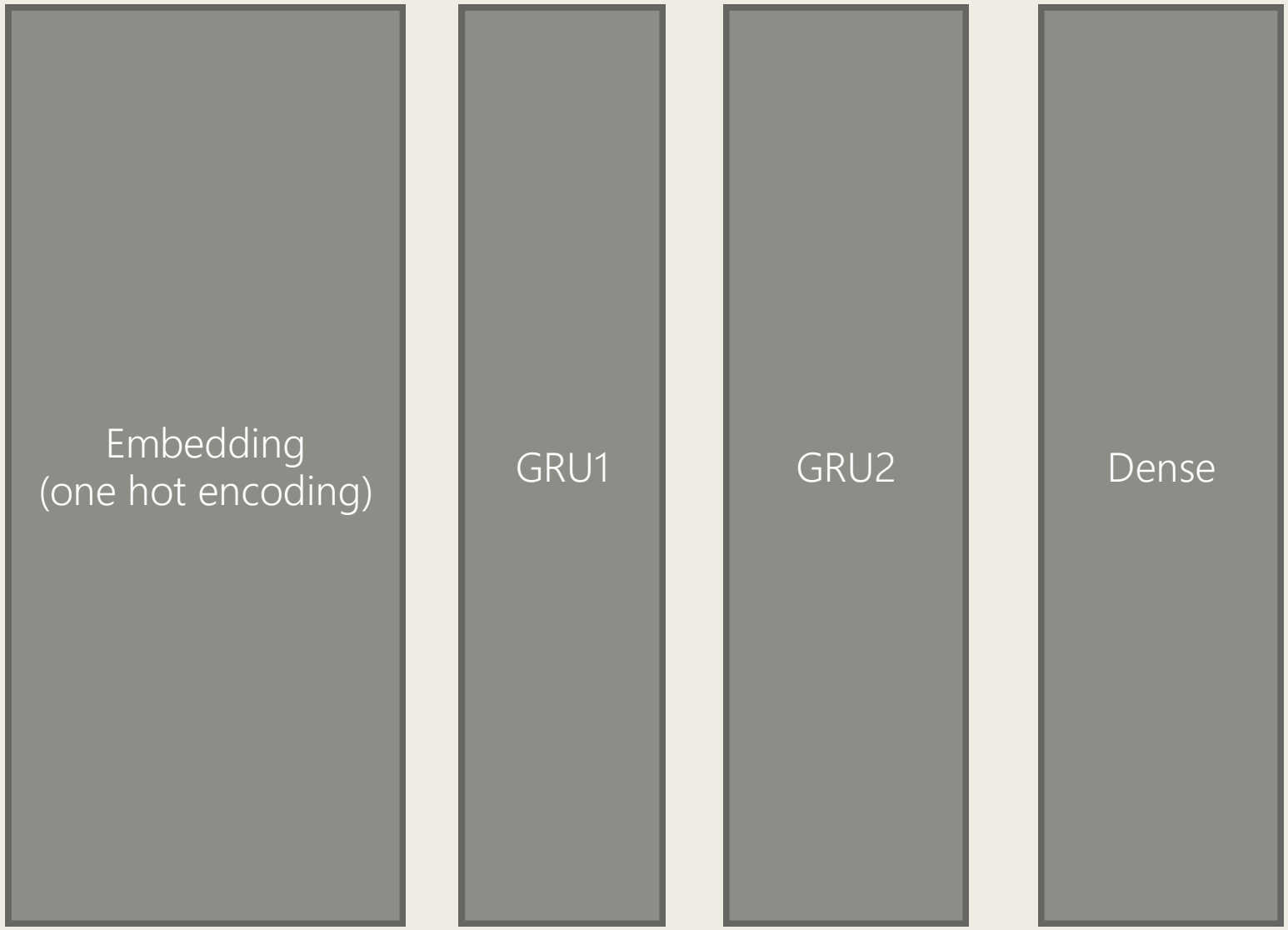
GRU - Accuracy

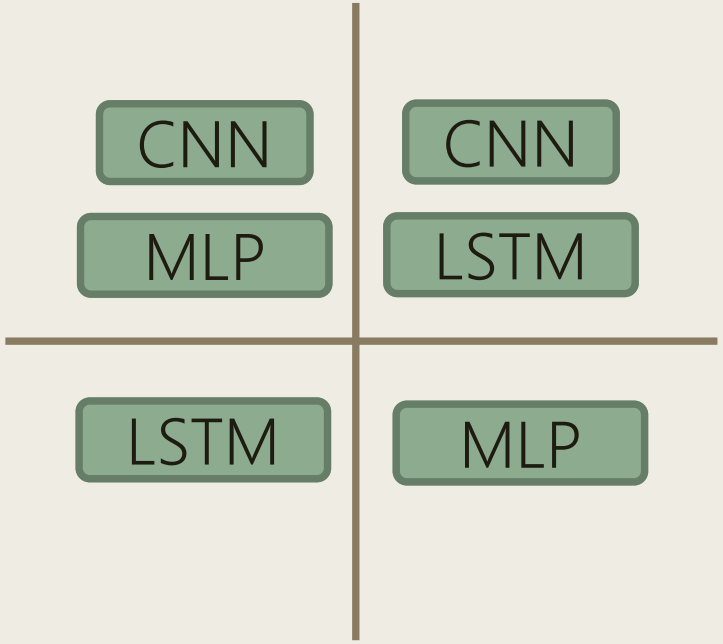
```
[20] model.fit(totalX, totalY, validation_split=0.1, batch_size=500, epochs=5, verbose=1, callbacks=[tbCall
```

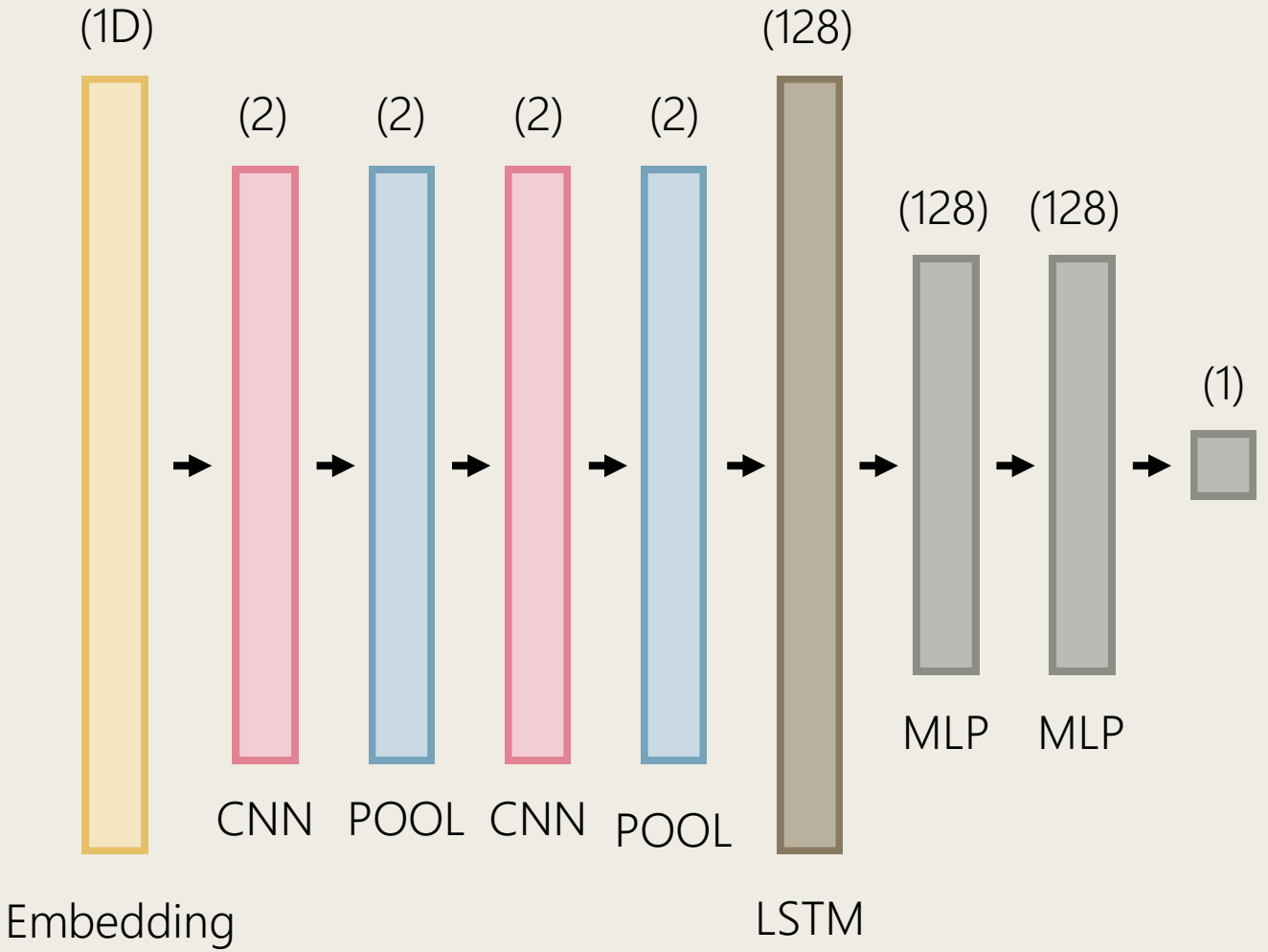
```
e on 2024 samples

==========] - 68s 4ms/step - loss: 0.2393 - accuracy: 0.9037 - val_loss: 0.4256 - val_accuracy: 0.8192

==========] - 63s 3ms/step - loss: 0.2187 - accuracy: 0.9141 - val_loss: 0.4867 - val_accuracy: 0.8276

==========] - 64s 3ms/step - loss: 0.2067 - accuracy: 0.9196 - val_loss: 0.4792 - val_accuracy: 0.8335

==========] - 64s 4ms/step - loss: 0.1982 - accuracy: 0.9253 - val_loss: 0.5640 - val_accuracy: 0.8330

==========] - 63s 3ms/step - loss: 0.1829 - accuracy: 0.9291 - val_loss: 0.5020 - val_accuracy: 0.8310
ory at 0x7f15a0fc2f98>
```

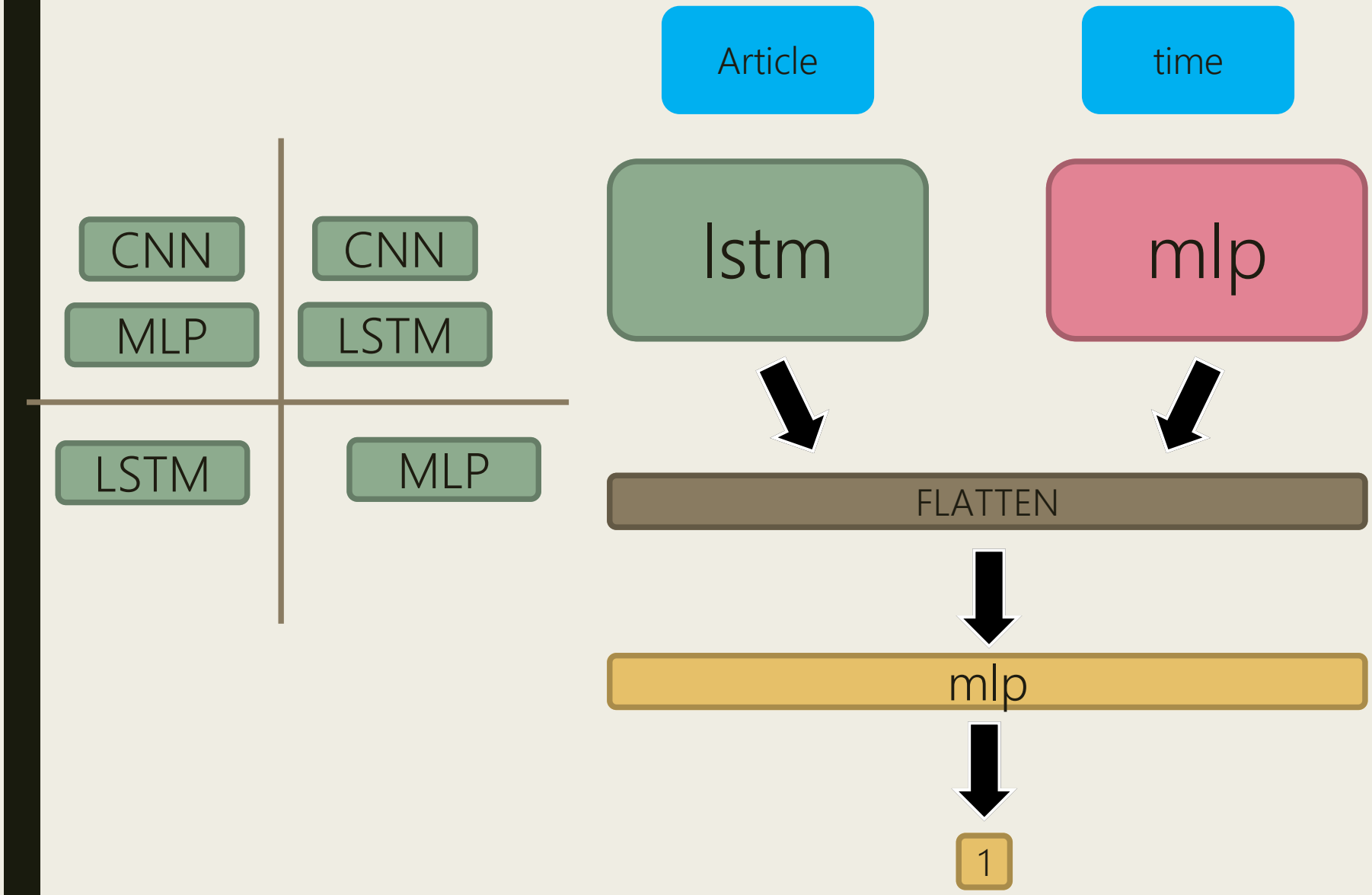
Model

```
Model: "sequential_7"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_7 (Embedding)      (None, 214, 256)          31046656
_________________________________________________________________
gru_13 (GRU)                 (None, 214, 256)          393984
_________________________________________________________________
gru_14 (GRU)                 (None, 256)               393984
_________________________________________________________________
dense_7 (Dense)              (None, 2)                 514
=================================================================
Total params: 31,835,138
Trainable params: 31,835,138
Non-trainable params: 0
```


Embedding
(one hot encoding)
GRU1
GRU2
Dense


CNN
CNN
MLP
LSTM
LSTM
MLP


(1D)
(2)
(2)
(2)
(2)
(128)
(128)
(128)
(1)
CNN
POOL
CNN
POOL
MLP
MLP
Embedding
LSTM


CNN
CNN
MLP
LSTM
LSTM
MLP
Article
time
lstm
mlp
FLATTEN
mlp
1

data

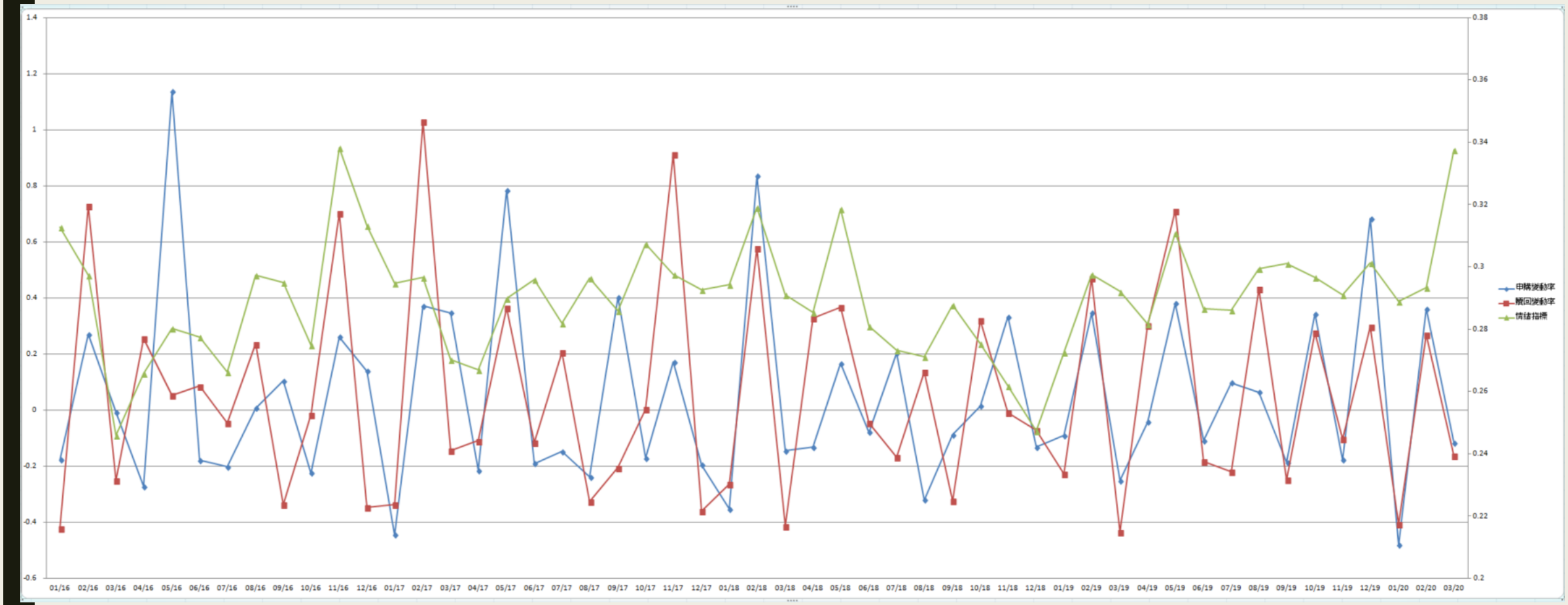
註：previous為尚未更新training data模型結果(資料筆數為1791)
+號表示測試準確度較未更新training data前提升，
表示加入新的新聞文本後，有提升辨識社群情緒準確度

- Total number of file is 1791 → 44025
- total data:41461
- Train num:37315
- Test num :4146
- maxSeqLength = 150
- Epochs = 30

Architecture	Test accuracy	Train accuracy
LSTM DENSE	0.628(+)	0.84336
CNN LSTM	0.5955(+)	0.8909
CNN MLP	0.5945(+)	0.9997(overfitting)
MLP	0.581(+)	0.9998(overfitting)
LSTM DENSE TIME (previous)	0.4804	0.6092
CNN LSTM time (previous)	0.5698	0.7239
CNN MLP time (previous)	0.4804	0.9336
MLP time (previous)	0.4916	0.9318

CNN_LSTM 情緒指標結果

紅：下月贖回變動率
藍：下月申購變動率
綠：當月情緒指標



預計未來完成及改進目標

- 建BERT模型，與LSTM、GRU進行比較
- 運用已建其他模型預測正負向情緒機率，建立每月社群情緒指標(搭配其他指標ex. 融資融券量)運用SVM等機器學習分類法或一般計量方法預測申購贖回
- 增加詞庫數
- 增強斷詞結果
- 增強情緒模型判斷準確度

目前困境

- 硬體運算能力不足
- 標註文本數需再提升
- 文本資料需清洗更完全
- 情緒模型準確度須提升