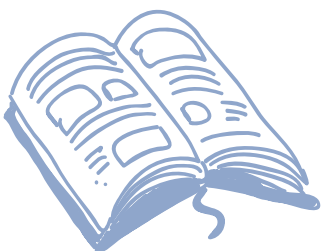


# 利用BERT自動判斷留言正負評

Dr. Ming-Chuan Chiu

Presenter: 洪聖博  
2022/01/07



# Outline

1. 問題定義與介紹
2. BERT模型架構介紹
3. 個案研究
4. 超參數優化
5. 結果
6. 結論

# Outline

1. 問題定義與介紹
2. BERT模型架構介紹
3. 個案研究
4. 超參數優化
5. 結果
6. 結論

## 問題定義

- 近年來網際網路迅速發展，除了能夠為商業服務增值，亦能提供景點的評論饋，以供觀光客參考與選擇，然而評論內容數大且繁雜，難以一一去檢閱以確保景點是否符合期望。
- 因此本研究目的即在為廣大觀光客提供景點評論的深度見解，以期能加速使用者尋找景點的過程。

# 5W1H

WHAT

- 判斷留言為正向或負向

WHO

- 利用Google評論尋找景點的使用者

WHEN

- 提供建議在使用者想要尋找景點的時候

WHERE

- 能在Google景點評論區進行判斷

WHY

- 避免字數過多的評論或過多的評論

HOW

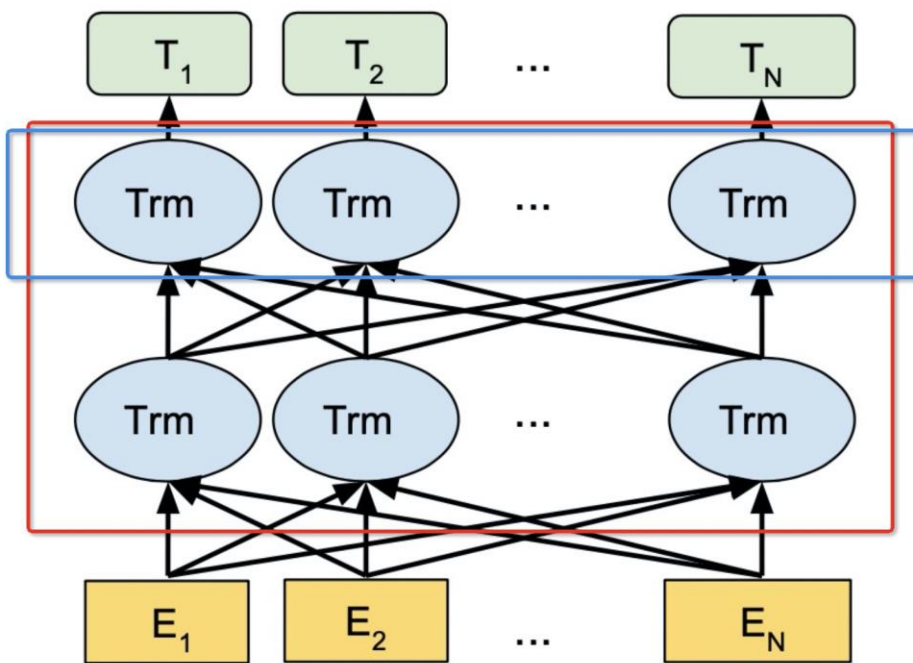
- 透過文本情感分析與文本分類等技術  
輔以網站開發，為使用者提供評價

# Outline

1. 問題定義與介紹
2. **BERT**模型架構介紹
3. 個案研究
4. 超參數優化
5. 結果
6. 結論

# MODEL

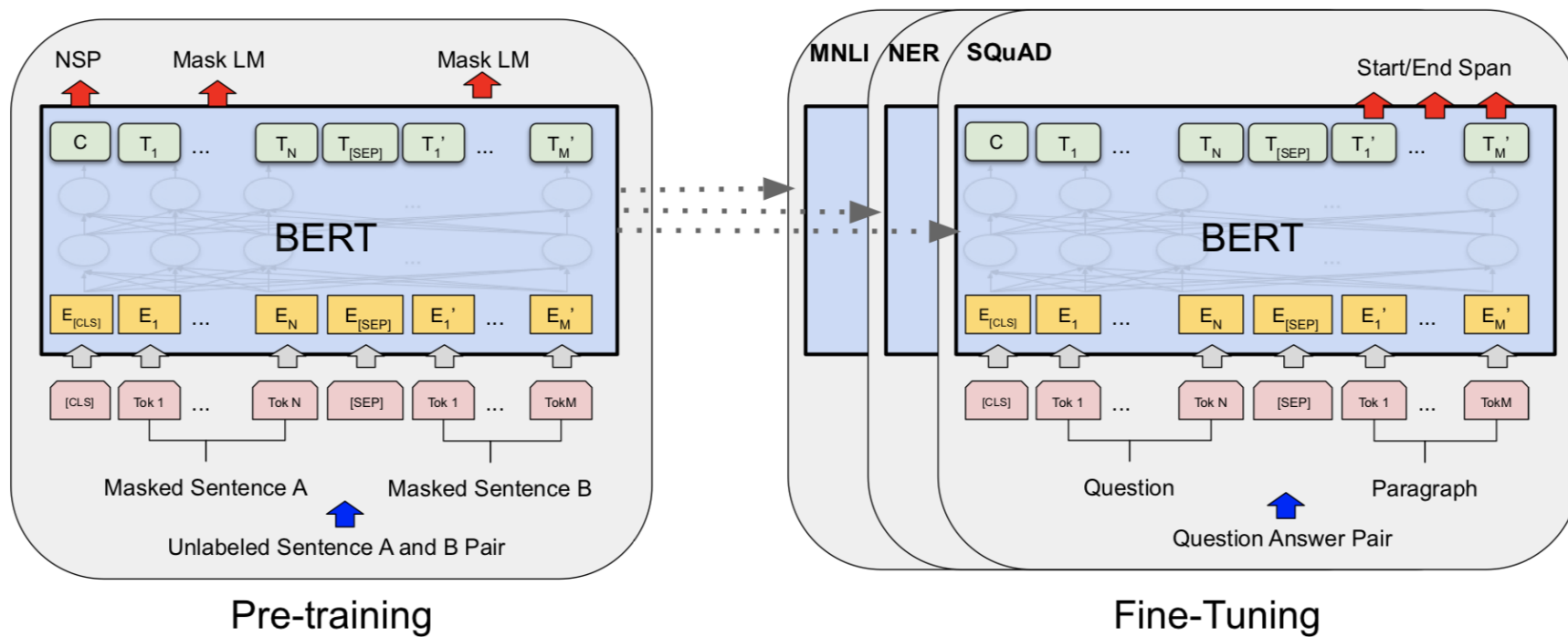
## BERT



- Bidirectional Encoder Representation Form Transformers
- 2018 年由 Google 提出
- 輸入一串句子  
給一串句子的embedding
- 會考慮上下文關係

# MODEL

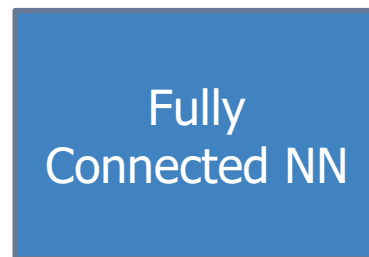
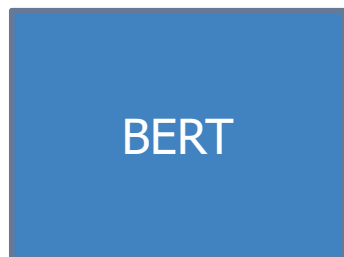
## Training Model





# 情感分析

鬼斧神工的天然美景  
不錯漂亮  
夏天有夠熱  
垃圾地方



1 (Good)

0 (Bad)

# Outline

1. 問題定義與介紹
2. BERT模型架構介紹
3. 個案研究
4. 超參數優化
5. 結果
6. 結論

## 資料獲取

- 利用Python網路爬蟲抓取Google景點評論
- 接著把分數轉換成好評論跟壞評論，4、5分為好評論，其餘為壞評論

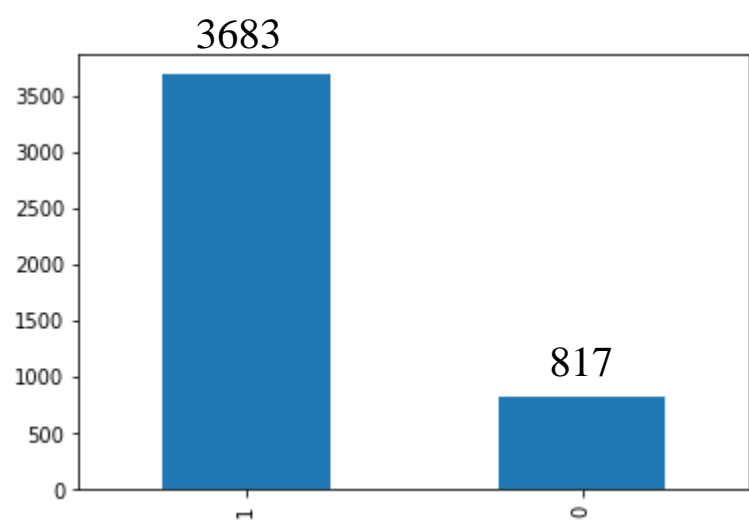
```
def convertScore(score):  
    if score >= 4:  
        return 'good'  
    else:  
        return 'bad'  
  
def Convert(google):  
    google['status'] = google['評等'].map(lambda e : convertScore(e))  
    google = google[google['status'].isin(['good', 'bad'])]  
    google['status'] = google['status'].replace({'good':1, 'bad': 0})  
    google = google.drop(columns = ['評等', '評論者'])  
    return google
```

# 資料前處理

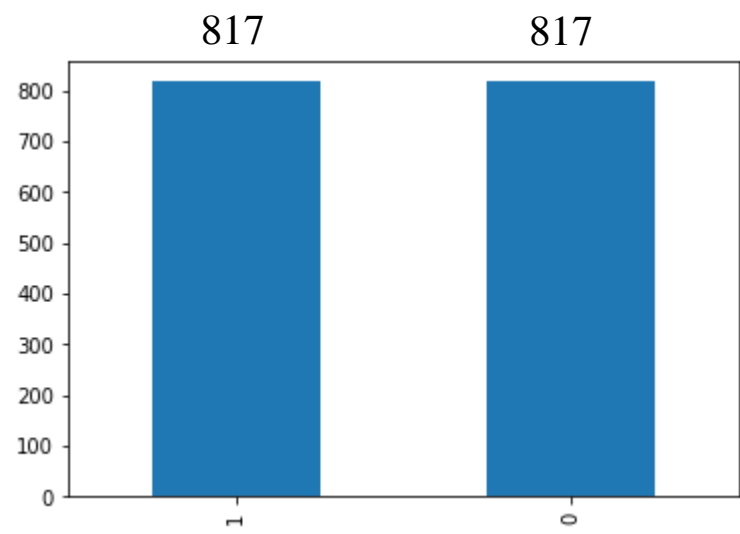
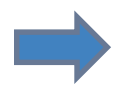
- 資料平衡

由於評論正負面資料量差距過大，可能會導致訓練失敗

```
▶ bad_df = google[google['status'] ==0]  
bad_df.shape  
good_df = google[google['status'] ==1]  
good_df = good_df.sample(n=817)  
google = pd.concat([good_df,bad_df],ignore_index=True)  
google.shape
```



處理前



處理後

# 研究資料

```
▶ from sklearn.model_selection import train_test_split  
   train_df, test_df = train_test_split(google, test_size = 0.2, random_state = 927)
```

- 數據集:
  - 13個觀光景點
  - 1634則評論
- 訓練集(80%):
  - 1307則評論
- 測試集(20%):
  - 307則評論

# Outline

1. 問題定義與介紹
2. BERT模型架構介紹
3. 個案研究
4. 超參數優化
5. 結果
6. 結論

# 實驗設置

- 超參數設計
- 以三因子二水準進行實驗
- 實驗次數為次 $2^3$ 次
- 本實驗以全因子實驗進行

	Level 1	Level 2
Activation Function	Relu	Sigmoid
Dropout	0.2	0.4
Pooling	0	X

# 超參數優化

	Activation Function	Dropout	Pooling	Accuracy
1	Relu	0.2	O	0.9572
2	Relu	0.2	X	0.9174
3	Relu	0.4	O	0.9543
4	Relu	0.4	X	0.9633
5	Sigmoid	0.2	O	0.9572
6	Sigmoid	0.2	X	0.9572
7	Sigmoid	0.4	O	0.9205
8	<b>Sigmoid</b>	<b>0.4</b>	<b>X</b>	<b>0.9694</b>



# Outline

1. 問題定義與介紹
2. BERT模型架構介紹
3. 個案研究
4. 超參數優化
5. 結果
6. 結論

# 結果

- 最終測試結果

	Activation Function	Dropout	Pooling	Accuracy
最佳參數組合	Sigmoid	0.4	X	0.9694

- 各類別準確度

	Accuracy
Good	0.9598
Bad	0.979

# Outline

1. 問題定義與介紹
2. BERT模型架構介紹
3. 個案研究
4. 超參數優化
5. 結果
6. 結論

## 結論(1/2)

- **貢獻**

本實驗成功訓練文本情緒分析模型，並試圖驗證實驗的有效性及模型性能，最終取得高達96%的準確度。

透過本次實驗的結果能讓使用者更快速了解該景點的評價。

- **限制**

由於本實驗所收集的留言僅含十三個景點，可能造成評論不夠多元。

## 結論(2/2)

- 未來展望

由於數據集不足，希望能增加更多資料來源例如:FB、IG等更多社群媒體。

希望能透過BERT模型，分類更多的類別，不僅僅只有好或壞。

**Thank You for Your Listening**