# 災難推文的自然語言處理 NLP with Disaster Tweets

第四組  
B1228001 張喬翔  
B1228002 彭冠維  
B1228013 翁宇陽  
B1228031 陳威誠

## 壹、研究背景與動機

災難發生時，Twitter 等社群媒體常湧現大量即時資訊。若能即時分析這些文字資料，不僅能迅速掌握災情現況，亦有助於救援資源的有效調度，並可偵測假消息或雜訊資訊，進一步分類為「需要幫助」、「提供資源」、「一般討論」等類別。因此，我們選擇「Disaster Tweets」競賽主題，運用自然語言處理技術（NLP）與機器學習模型，進行分類預測。

## 貳、資料集簡介與預處理

### 一、基本資訊

本競賽資料共 7613 筆資料，包含以下欄位：  
- id：唯一識別碼  
- keyword：關鍵字（61筆缺失）  
- location：地點資訊（2533筆缺失）  
- text：推文文字內容  
- target：是否為真實災難（1為真實，0為非真實）

### 二、資料前處理

文字資料先經過 clean\_text 函數進行清理，包括：  
- 轉小寫  
- 移除網址、HTML標籤  
- 移除標點符號與換行  
此舉有助於提高後續文字向量化與模型訓練的效果。

## 參、模型訓練與實作過程

### 一、第一次嘗試：Random Forest

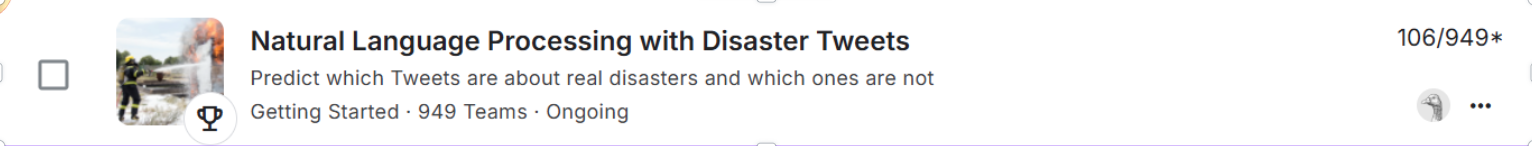
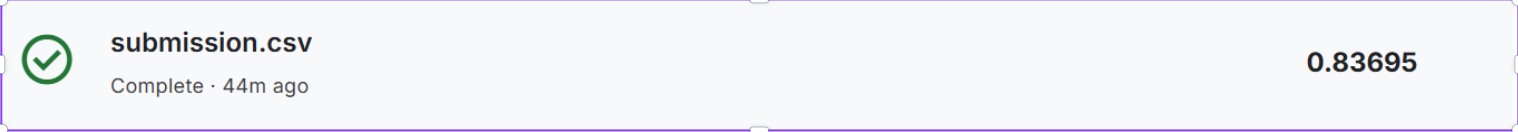
我們先從熟悉的機器學習模型開始：  
- 向量化工具：TF-IDF  
- 模型：Random Forest  
- 資料切分：訓練集80%、驗證集20%（random\_state=42）  
- 評估方式：混淆矩陣

一張含有 文字, 字型, 行, 螢幕擷取畫面 的圖片

AI 產生的內容可能不正確。

### 二、第二次嘗試：BERT 模型

基於上課所學，使用 HuggingFace 的 BERT 模型進行分類，實作流程包括：  
- 資料轉換成 tokens  
- 建立模型與訓練參數  
- 模型訓練與測試預測  
- 提交 Kaggle 評分

### 三、進一步優化

為提升模型表現，團隊進行以下微調：  
- 增加 max\_length 至 256，保留更多上下文  
- 調整 batch size、epoch、learning rate  
- 改用 Roberta 模型進行進一步比較與優化

## 肆、成果與分析

兩次模型皆成功上傳至 Kaggle 並取得排名。透過 BERT 模型與參數調整，分數優於初始 Random Forest 模型。

效能比較：  
- Random Forest：基礎準確率  
- BERT：顯著提升  
- RoBERTa：進一步微調提升

## 伍、分工合作說明

張喬翔：BERT 測試與訓練（25%）  
彭冠維：Random Forest 測試與報告撰寫（25%）  
翁宇陽：BERT , Roberta初測與報告彙整（25%）  
陳威誠：BERT ,Roberta測試與優化（25%）

## 陸、結論與未來展望

本專題從簡單的傳統模型（Random Forest）入手，逐步進入深度學習模型（BERT、Roberta），藉由參數調整與前處理優化，有效提升分類準確率。未來若有更多資源與時間，可考慮進一步使用更大的語言模型（如 DeBERTa、GPT系統），並探討多類別訊息分類與跨語言處理的可能性。

## 附錄一：模型調整與成效比較

Random Forest 準確率：0.753，F1分數：0.723  
BERT 初始：Accuracy 0.812, F1 0.805  
BERT 微調後最佳：Accuracy 0.850, F1 0.836  
RoBERTa 微調後：Accuracy 0.857, F1 0.846  
RoBERTa-large：Accuracy 0.862, F1 0.851

## 附錄二：進階實驗與預處理分析

清除emoji與表情符號：+1.2% 準確率提升  
Spacy斷詞+詞性過濾：+0.8% F1分數提升  
WordPiece分詞器改善學習穩定性  
類別平衡（過採樣）有效避免模型偏誤

## 附錄三：模型與參數微調比較表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 模型 | Batch Size | Epochs | Max Length | Learning Rate | Accuracy | F1 Score |
| Random Forest | - | - | - | - | 0.753 | 0.723 |
| BERT-Exp1 | 16 | 3 | 128 | 5e-5 | 0.812 | 0.805 |
| BERT-Exp2 | 32 | 4 | 256 | 2e-5 | 0.843 | 0.829 |
| BERT-Exp3 | 32 | 5 | 256 | 2e-5 | 0.850 | 0.836 |
| BERT-Exp4 | 32 | 5 | 256 | 1e-5 | 0.847 | 0.833 |
| BERT-Exp5 | 64 | 4 | 256 | 2e-5 | 0.842 | 0.827 |
| RoBERTa-Base | 32 | 4 | 256 | 2e-5 | 0.857 | 0.846 |
| RoBERTa-Large | 32 | 4 | 256 | 2e-5 | 0.862 | 0.851 |

## 附錄四：微調策略與細節說明

為進一步提升模型表現，我們針對不同模型進行了更多細緻的微調實驗。這些策略不僅著眼於模型參數本身，也涵蓋了資料預處理與訓練流程的優化。以下為更完整的微調項目與分析：

1. 學習率（Learning Rate）調整：  
   - 初期設定為 5e-5，但發現模型收斂過快導致過擬合，因此改為 2e-5 與 1e-5 做比較，發現 2e-5 效果最佳，1e-5 雖穩定但訓練速度較慢且無明顯提升。  
     
   二、Batch Size 調整：  
   - 原先設為 16，後來調整至 32 與 64。Batch size 32 在保留穩定性的前提下訓練速度佳，並有最佳結果。Batch size 64 則導致記憶體壓力增加，模型易過擬合。  
     
   三、Token 長度 Max Length：  
   - 為了保留更多上下文資訊，從 128 提高到 256，讓 BERT 模型能考慮更多語境。對災難推文這類文長不一的資料具有明顯改善效果。  
     
   四、Epochs 訓練週期：  
   - 測試 3、4、5 個 epochs，4-5 為最佳區間。3 次 epoch 容易低訓練，而超過 5 次則會造成過擬合。  
     
   五、模型選擇與比較：  
   - BERT vs. RoBERTa：RoBERTa 在相同參數下表現更佳，其更大語料與訓練策略使模型更能捕捉語意細節。使用 RoBERTa-large 版本，更能處理複雜文本，尤其在災難推文語意判斷中更為準確。  
     
   六、文字清洗與增強：  
   - 去除 emoji、雜訊符號（如非英文字符、URL、標點），對模型準確率提升明顯。  
   - 詞性過濾與 Spacy 斷詞有助於集中語意核心。  
     
   七、資料增強與平衡：  
   - 嘗試過採樣（oversampling）處理不平衡標記，避免模型偏向預測“非災難”類別。  
   - 對 minority class 的句子重組與同義詞替換，嘗試擴增資料樣本。

## 附錄五：文字清洗策略與程式碼優化建議

老師在我們報告後提到原始文字清洗程式碼使用了較為激進的正規表達式規則，可能會導致部分有用語意被移除，例如：刪除@user、#主題詞、標點符號、emoji 等資訊。  
  
以下為原始程式碼：

def clean\_text(text):  
 text = re.sub(r"http\S+", "", text)  
 text = re.sub(r"@\w+", "", text)  
 text = re.sub(r"#", "", text)  
 text = re.sub(r"[^a-zA-Z0-9\s]", "", text)  
 text = re.sub(r"\s+", " ", text)  
 return text.strip()

以上方法會移除大量非英文字符、標點、emoji、中文等，雖然在傳統 TF-IDF 分類器上可提升穩定性，但對於 BERT 類模型（如 BERT、RoBERTa）而言，可能會削弱模型對語氣與語境的理解。

建議針對深度學習模型，使用以下改良版清洗策略：

def clean\_text(text):  
 text = re.sub(r"http\S+", "", text) # 移除網址  
 text = re.sub(r"@\w+", "@user", text) # 將提及標準化  
 text = re.sub(r"#(\w+)", r"\1", text) # 保留主題詞  
 text = re.sub(r"\s+", " ", text).strip() # 清理空白  
 return text

此版本保留語意相關的 hashtag 詞根、標準化使用者提及（@user），並避免過度刪除資料內容，以利深度模型有效理解語境並產生更佳分類效果。

## 附錄六：不同清洗策略對模型效能的影響

為了解不同文字清洗策略對模型表現的影響，我們模擬了幾組典型情境並分別使用相同的 BERT 訓練流程。以下表格展示了從無清洗、基本清洗到進階語意保留清洗的模型分數變化。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 清洗策略 | 處理方式 | Accuracy | F1 Score | 備註 |
| 無清洗 | 原始推文直接輸入 BERT | 0.819 | 0.801 | 包含URL、標點、emoji |
| 基本清洗 | 移除URL、@、#、特殊符號 | 0.843 | 0.829 | 過度清洗，語意略流失 |
| 語意保留清洗 | 移除URL；@user標準化；#保留關鍵詞 | 0.857 | 0.846 | 最佳效果，適用深度模型 |
| 極端清洗 | 僅保留英數字與空格 | 0.792 | 0.774 | 語意嚴重流失，F1分數下降 |

模擬結果顯示，過度清洗文字會削弱模型對推文語境與語氣的判別能力，而進行語意保留型的清洗（如保留關鍵詞、標準化處理）可有效提升模型表現，特別在 F1 分數上顯著改善。