Natural Language Processing with Disaster Tweets

報告人:徐聖峰
日期: 2022/01/19

BrainStorming (問題脈絡釐清)

1. Kaggle 的比賽之中提供了哪些可用的資料？
    a. 關鍵字
    a. 地點
    b. 推文 (Tweet)
2. 著重在分析**推文**，但同時試著從關鍵字以及地點找出一些 insight
3. 有哪些可能遇到的問題
    a. 關鍵字和地點：NULL Values
    b. 推文：不知現有的分析工具是否能處理 User mention (@) 或者 Hashtag (#) 以及 URL
4. 預期結果
    - 根據 Kaggle 的 Leaderboard，希望至少能達到前10%

分析步驟

01 Data Exploration

- 快速讀過資料，並找出可行的分析方法

02 Data Analysis - Keyword, Location

- 對關鍵字以及地點進行Preprocessing
- 比較各種 Preprocessing 優劣
- 找出最佳的模型

03 Data Analysis - Tweet

- 閱讀 Tweet 分析相關技術
- 找出可用技術並實現
- 分析不同技術之間的差別

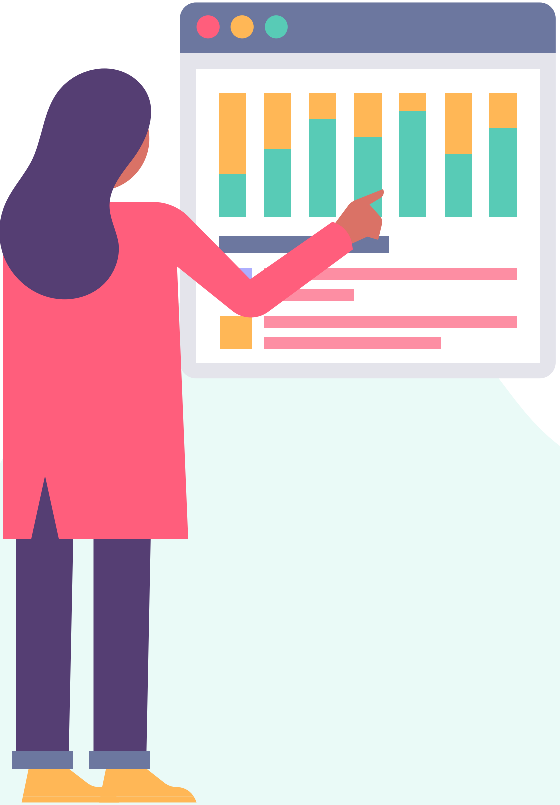
04 Further Improvement

- 反省並檢討，還有什麼可改進的地方？

01

Data Exploration

- Total Record: 7613
- Keyword: 7552 (Not NULL), N/A -> ‘empty string’
- Location: 5058 (Not NULL), N/A -> ‘empty string’
- Text(Tweet): 7613 (Not NULL), Need to process URL, User Mention, Hashtag
- Keyword + Location as feature -> Machine Learning Approach (Decision Tree, SVM)
- Tweet as feature -> Need to apply transfer learning (BERT)



02

Data Analysis - Keyword, Location

Data Preprocessing - Keyword

URL Encode Removal

- body%20bags → body bags
- oil%20spill → oil spill

Lemmatization

- annihilated → annihilate
- wrecked → wreck

Data Preprocessing - Location

Remove Nation/State (keep prefix)

- London, UK → London
- Vancouver, Canada → Vancouver
- Queensland, Australia → Queensland

Remove space and lowercase

- World wide → worldwide
- WORLD WIDE → worldwide
- New York → newyork
- Winston-Salem → winston-salem

Data Modeling - Keyword + Location

Model Selection

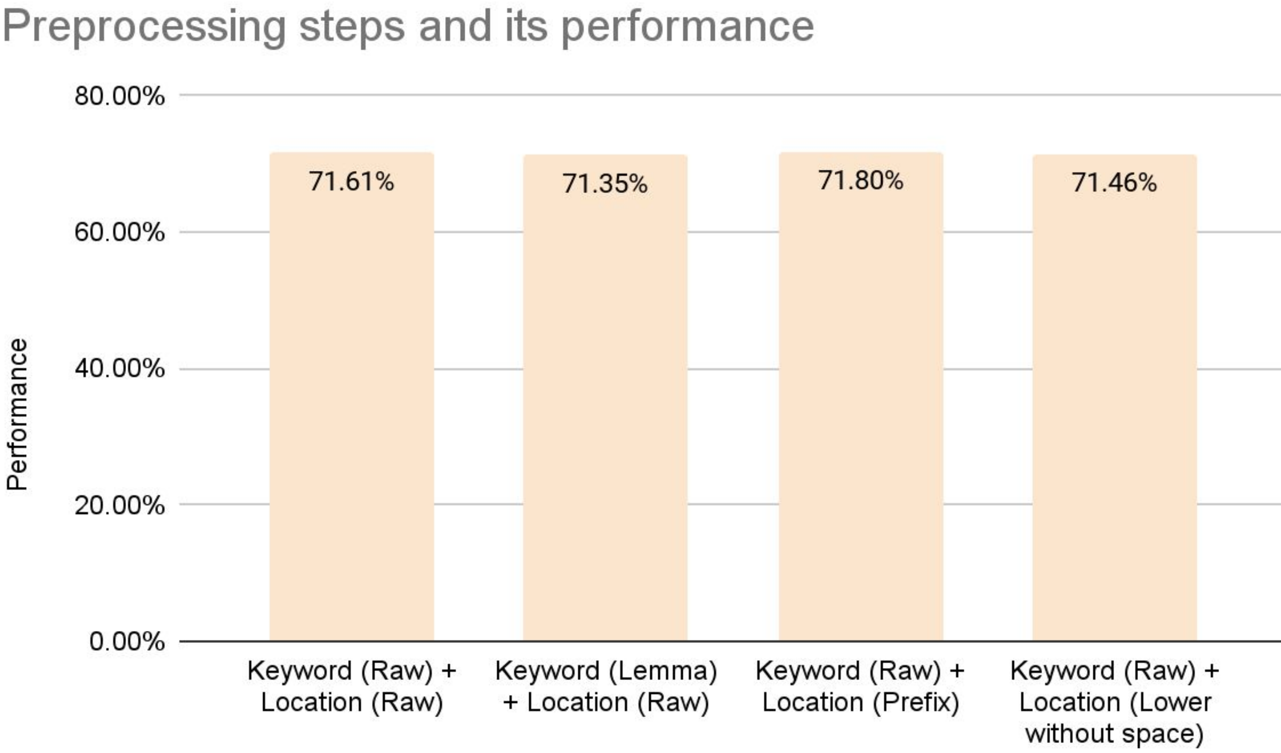
Decision Tree is fast and suitable for prototyping

Preprocessing

Raw keyword and location prefix have best performance

Summary

Also tried SVM and gradient boosting, and SVM have highest validation accuracy



Data Modeling - SVM (Keyword, Location)

- 實驗 Notebook (包含 Decision Tree, SVM): Colab
- Kaggle Submission Notebook (SVM): Kaggle
- Validation Accuracy: 73.46%
- Test Accuracy on Kaggle: 72.11%
- Leaderboard Ranking: 788/868

Data preprocessing is not always boosting the performance

—**Someone Famous**

03

Data Analysis - Text (Tweet)

Tweet Analysis - Baseline

Model:

Simple Transformer in Keras API

Validation Accuracy:

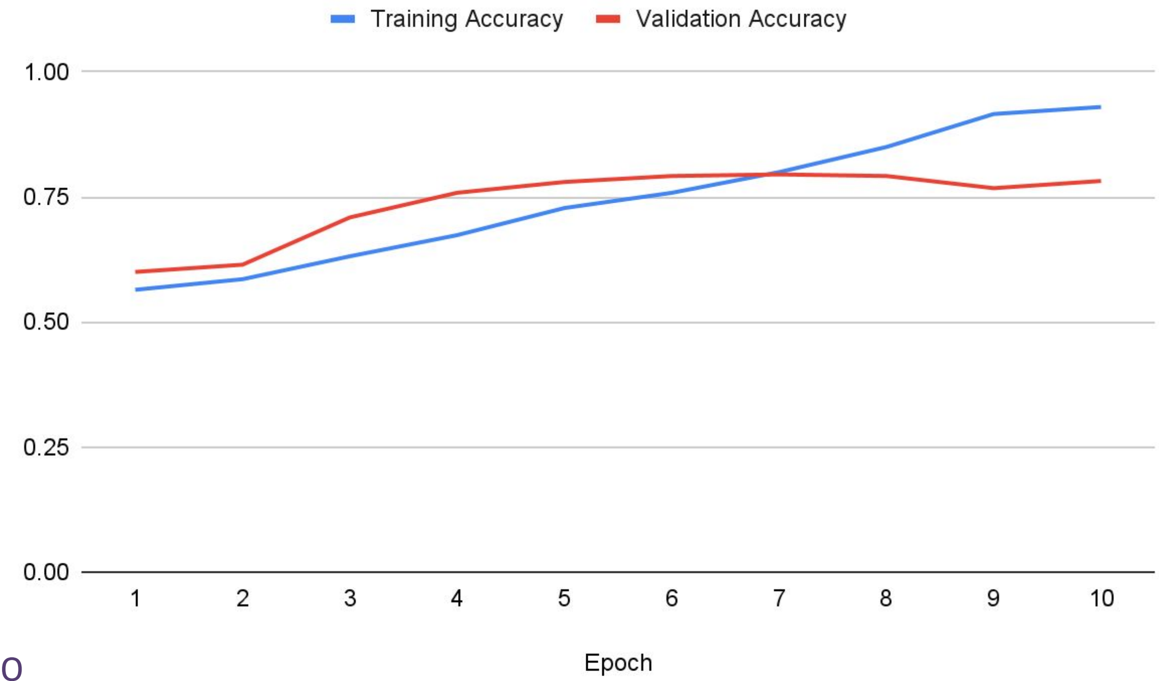
0.7919%

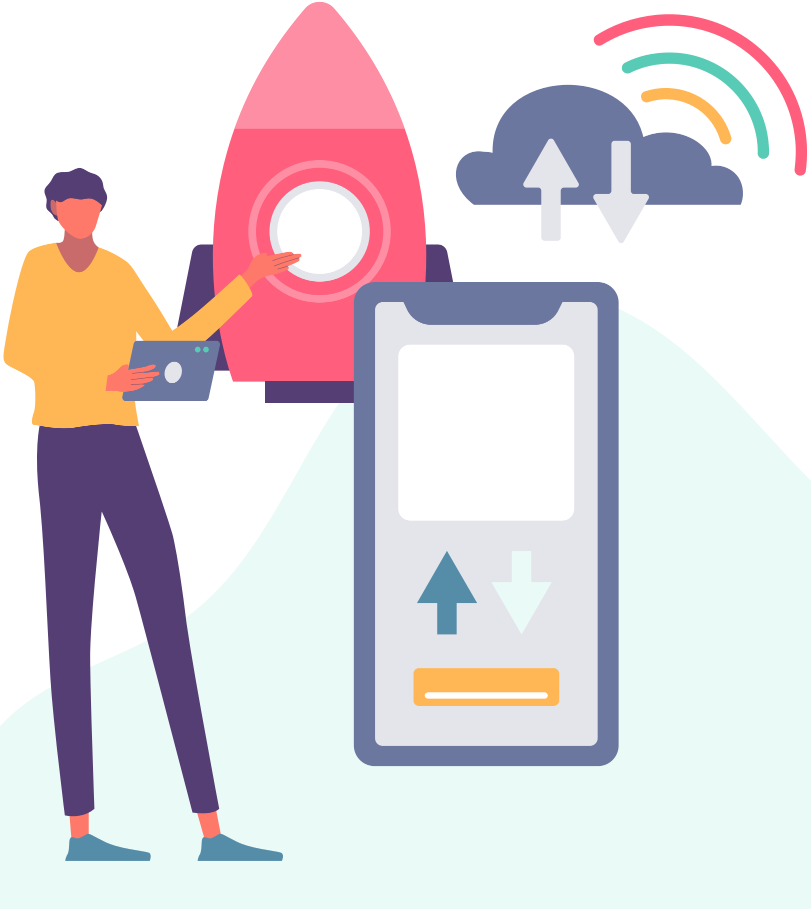
Status:

Overfitting

Explanation:

Without transfer learning, it's hard to train a text classifier with small dataset





What to improve?

1. Review more advanced approaches on tweet analysis

2. Transfer learning might be great

Paper Research

TWEETEVAL:

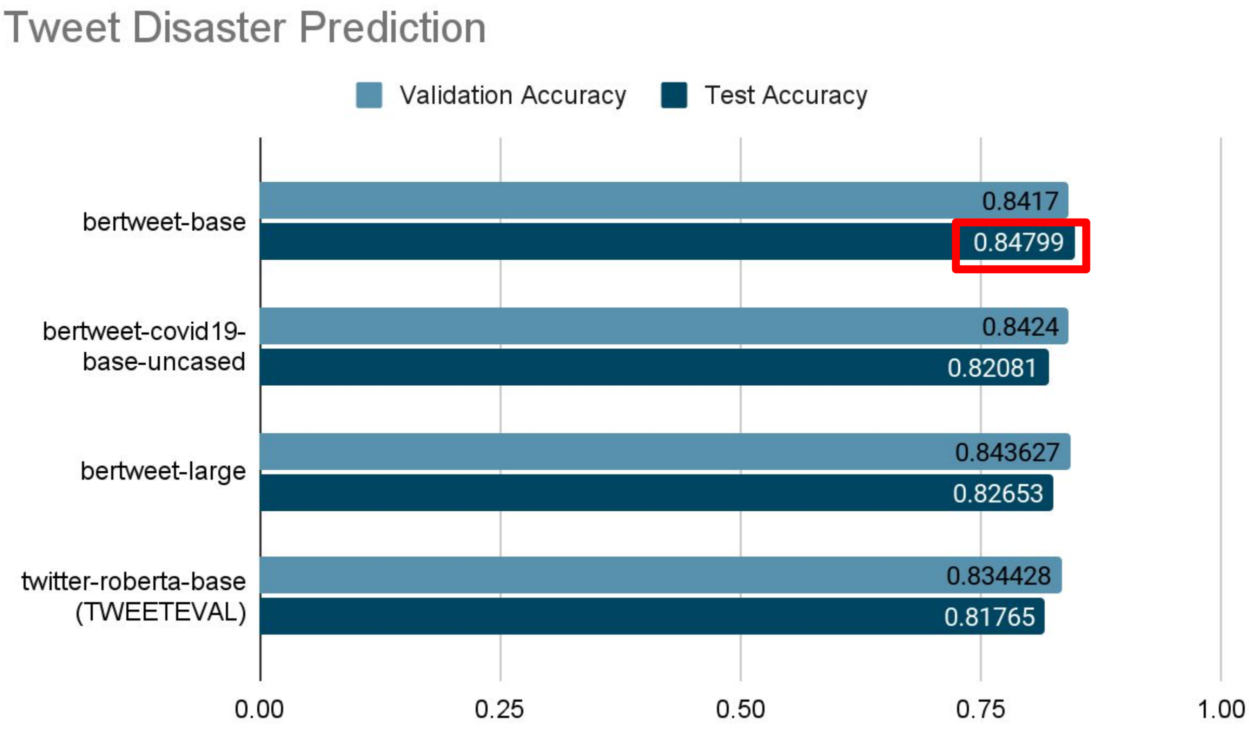
- Use the pre-trained weight from RoBERTa and further train on 60M Tweets

BERTweet:

- Having same architecture as BERT base
- Using RoBERTa pre-training procedure to pre-trained on 850M Tweets
  - Dynamic Masking
  - Remove Next Sentence Prediction Task

Fun fact: RoBERTa Prediction, Twitter-xlm-roberta-prediction

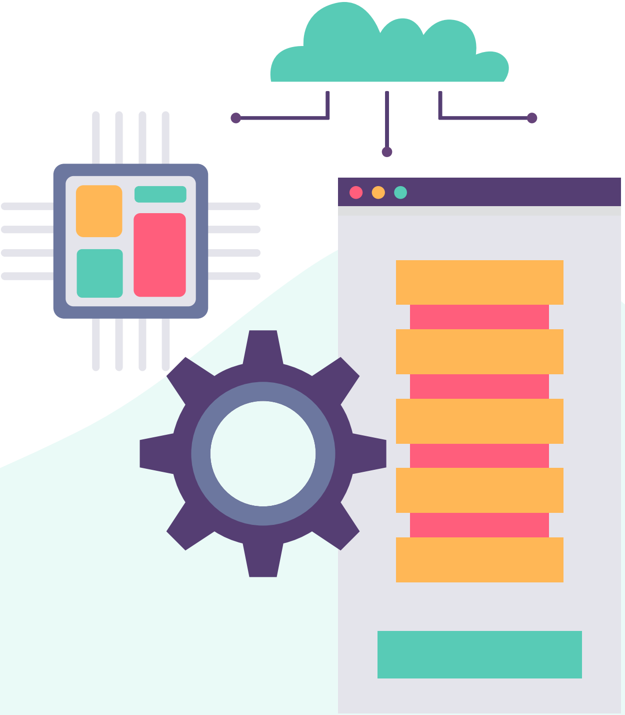
Experiment Results



Experiment Notebook (Colab)

Best Model - BERTweet

- **Model:** vinai/bertweet-base
- **Validation Accuracy:** 84.17%
- **Test Accuracy:** 84.799%
- **Leaderboard Ranking:** 48/818 (6%) → Beat my expectations
- **Further Improvement:** Pre-trained tweet on new models such as
  - DEBERTA: https://arxiv.org/pdf/2006.03654.pdf
  - ALBERT: https://arxiv.org/pdf/1909.11942.pdf



Thank you