1102\_資料分析與學習基石 個人專題期末報告

黃振嘉 F74086048

資料集來源figure-eight [Data For Everyone’ website here](https://www.figure-eight.com/data-for-everyone/)

競賽首頁 [Natural Language Processing with Disaster Tweets | Kaggle](https://www.kaggle.com/competitions/nlp-getting-started)

1. 競賽簡介

這是一個由figure-eight這間公司製作的資料集，後來被放上Kaggle供大家使用。訓練的檔案包含10875筆資料，分別有keyword、location、text三個標題的內容。此資料集的特色是其中有出現hashtag(#)、at(@)，而且當中也有混雜一些次文化用語如表情符號與縮寫。這些內容與先前處理的標準語法會有一些出入，因此我認為這個資料具有相當的挑戰性。

競賽目的是要訓練出一個模型，用以分辨使用者的Tweets是否與災難相關。因為資料來源自推特的貼文，相較於文章或是新聞部分，內容有較多不符合文法與拼字的情形

1. 資料前處理

* “location”與”keyword”

在訓練與測試資料及當中，0.8%的“keyword”和33%”location”是同時缺失的，而且有許多不同種類的，數據相當混亂，因此我在前處理的部分便將它們刪除不在訓練中使用。



* URL、E-mail處理

一開始使用的是re.sub(r"http\S+", "", tweet)等函式處理網址,信箱地址。後來改用text\_hammer 來處理，包含remove\_emails()，remove\_html\_tags()，remove\_special\_chars()，remove\_accented\_chars()

* Emoji、Slang、Abbreviations、Punctuation(except’\’’)

這部分是tweets最大的特色，大多用戶都有使用表情符號以及各種次文化中出現的詞句。這部分暫時沒有函式能夠一手包辦。因此我使用網路資源，利用表列的方式呈現

* ‘#’ 和 ‘@’

因為這兩個符號再推文中，其其他符號相比有不同的意義，所以我有特別處理他們。前者hashtag比較多在使用上是跟文字敘述差步道很遠，甚至有些使用者的hashtag是一長串的，我僅將’#’去掉而已。那’@’後面接的則是帳戶名，大多對於判斷無助，我便將整個tag的部分都移除。

* Spell checker

我利用pyspellchecker模組進行前處理，使用的是Levenshtein距離演算法，來搜尋與原始單詞 2 的編輯距離內的排列組合。它會將所有排列（插入、刪除、替換和轉置）與詞頻列表中的已知詞進行比較，最後找出最有可能的字詞。這個模組很強大，然而他所後費的石將很長，所以我有將處過的文字另外輸出成檔案，每次訓練模型時就直接讀取即可。

1. 模型架構

tokenizer = AutoTokenizer.from\_pretrained('bert-large-uncased')

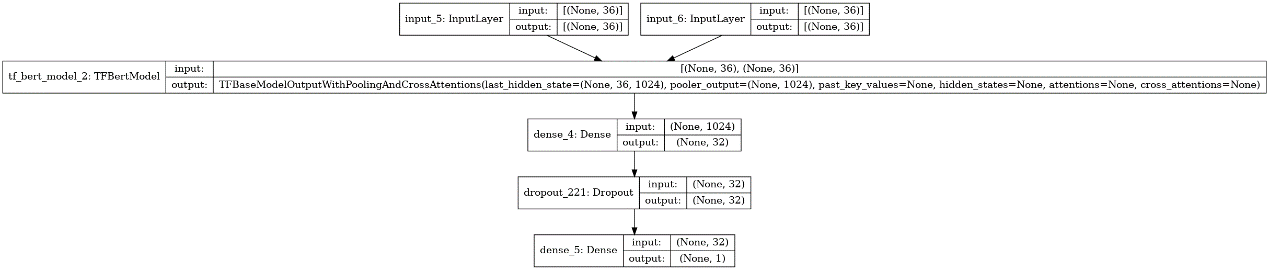
bert = TFBertModel.from\_pretrained('bert-large-uncased')

我使用的是bert-large-uncased的tokenizer與pretrained模型，下圖第一張是一剛開始是試的模型結構，但成效不夠好，後來看了很多別人的分析，經過參考後的模型是第二張圖。

一張含有 桌 的圖片

自動產生的描述

第二張圖的部分是最後採用的模型。



1. 心得

這次課程是我第一次接觸文字處理的模型，很開心有做出一點小成果。上次check point交的報告有設定目標，而我也有成功達成前百分之30，目前最佳成績為165/956。雖然說模型的一些超參數我還是只有大方向的概念，但是經過實作我更加熟悉這些內容了。

一張含有 文字 的圖片

自動產生的描述