學號：R05942101 系級： 電信一 姓名：陳泓弦

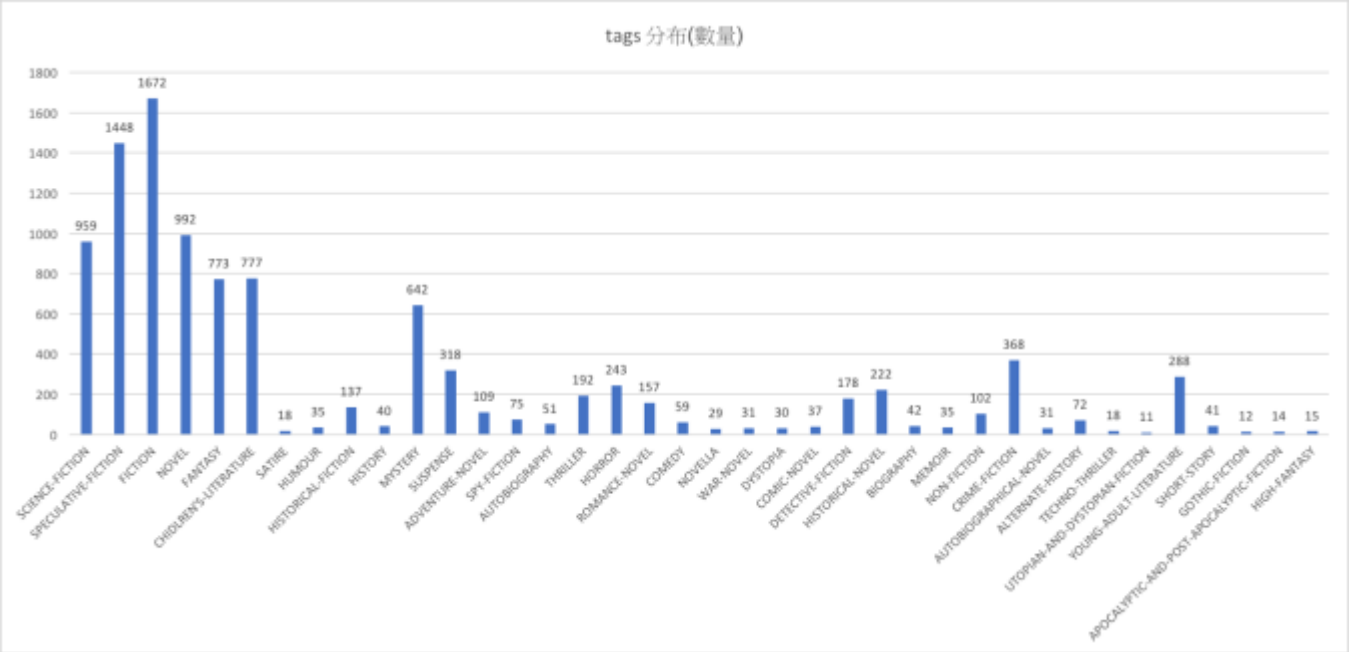
1. (1%)請問 softmax 適不適合作為本次作業的 output layer? 寫出你最後選擇的 output layer 並說明理由。

A: 全部小說分為 38 類，故 softmax 得到的 output 是一個 38 維的 vector，每個 element 代表屬於某一類的機率,但因為在這作業每筆資料都是 multi-label，所以沒辦法設一個 threshold 去找到最佳解，舉例來說，如果這筆資料屬於兩類，那 threshold 大概會設 0.5 附近，如果屬於三類，那 threshold 會想設 0.3 附近，不過我們無法得知此資料究竟屬於幾類，故 softmax 不太適合本次作業的 output layer。我選擇 sigmoid 作為 output layer，同樣也是 output 一個 38 維的 vector。sigmoid 是將每一類做二分法，屬於此類或不屬於此類，這樣的話就能設 threshold=0.5 去判斷是否屬於此類，但我設 threshold=0.3，原因在第三題會說明。

2. (1%)請設計實驗驗證上述推論。

A: 用同一個 model，predict 的 threshold 都設 0.3，在 output layer 分別用 sigmoid 和 softmax 去看他們的 performance，結果 sigmoid 在 kaggle 拿到 0.5，而 softmax 只拿到了 0.2，故在此作業中用 sigmoid 當作 output layer 會比較好。

3. (1%)請試著分析 tags 的分布情況(數量)。



從上圖可以發現大部份類別數量都蠻少，表示資料量不夠多。一般做 sigmoid 時是以 0.5 來做為 threshold，但因為訓練時某些類別的資料量不夠多，model 沒有足夠多的資訊學到此資料是否屬於某一類，所以 threshold 應該要設低一點，我最後是設 0.3，在 kaggle 上拿到 0.5165。

4. (1%)本次作業中使用何種方式得到 word embedding?請簡單描述做法。

A: 我使用了 glove 和 bag of words。glove 的概念是若兩個字 w_i 和 w_j 共同出現在同一篇文章的次數很頻繁，表示這兩個字有比較緊密的關係，glove 的做法是收集一堆文章，去找出一組向量 $V(w_i)$ 和 $V(w_j)$ 做內積得到的值，會接近他們共同出現在同一篇文章的次數。在作業中是直接拿網路上別人做好的 glove 來用。

bag of words 的概念類似 1-of-N encoding，餵入好幾篇文章，會先找出文章中總共有幾種不同的字，假設有 1000 種不同的字，那麼他會給每種字一個 index，然後用一個 1000 維的 vector 去表示每一個文章，若此文章包含了 index=1,10,100,900 的字，那麼在 vector 相對應的 index 會設為 1，其餘為 0。

5. (1%)試比較 bag of word 和 RNN 何者在本次作業中效果較好。

A: 在此作業中，我用 bag of word 得到了比較好的效果，原因我想或許是在這次作業的 train_data 和 test_data 總共有 51866 種字，若使用 glove，會有 12972 種字是找不到的，所以訓練出來的效果就蠻差的，在 kaggle 最好只拿到 0.43，如果能把缺的字補齊，那效果應該會比 bag of words 還要好很多，因為 glove 可以用 RNN 下去訓練，可以將文法和詞序也考慮進去，但用 bag of words 並不會考慮文法和詞的順序。