PA1 REPRORT

R04725040 黃柏睿

Environment:

Python 3.5.0

VSM Implementation:

1. Pre Processing:

   一開始我先把所有的 inverted_file 掃過一遍，找出 term 跟對應到的 documents term 的部分把助教附的 22 中文 stopword 跟網路上找到的英文 stopword 都去掉。中文 bigram 只要有包含一個 stopword 就去掉。
2. Model:

   我採用 Tf-idf 跟 Cosine Similarity，TF 的部分用 Okapi 做 Normalize （b = 0.8 參考網路上的）
3. Query Vector

   用 concepts + title + narrative + question 做成大雜燴，把他們做 unigram 跟 bigram，再留下有出現在 inverted_file 裡面的 term，算出 tfidf。

Rocchio Feedback:

1. 一開始實作 Rocchio feedback 的時候，分數是比沒做還要低。後來發現是因為我一開始做的 top-k relevent documents 太少（10個），但這十個跟 query 不太相關，導致我的 rocchio 生出來的新 query 反而比原本的來爛。
2. 所以我稍微試了一下，改用 top-50 作為我的 relevant documents，分數就有明顯上升，也比沒用 Rocchio feedback 還要好。
3. 另外如果直接把 relavent documents 的 term 都加上來，新的 query vetor 會變得太大，同時也吃進很多不相關的東西。所以我只取 tfidf 是前 500 名的 term 作為新的 query vector。
4. Rocchio 參數的部分，我是直接參考 wiki 上面的，alpha = 1、beta = 0.8，而 non-relavent 的 document 我直接不管他。

檢討:

1. 一開始以為還蠻簡單的，就照著講義上的公式把 code 寫出來就好，沒想到還有很多地方需要注意跟實驗來找出最佳解
2. 要把 Load inverted_file 實在是太久了，而且很吃 Memory，後來我用 emacs 跟 ipython 做連到 Server 上編輯跟執行，節省了不少 coding 的時間
3. 最後在 Memory 上的控管似乎不是很理想（吃了 12GB），或許可以思考看看如何使用更少的資源來達成這次作業
4. 另外這次因為太晚開始寫，沒有跑什麼 Tune 參數的 Training 程式，分數就持平在 0.7 左右，有點可惜