Information Retrieval – HW4

B10615046 柯元豪

實作環境

Jupyter notebook (Python 3)

使用的套件

numpy, pandas, pylab.random, math, nltk, Counter, numba, tqdm

資料前處理

用 nltk 的 PlaintextCorpusReader 去目錄下存取所有詞，並用 Counter 建成字典，最後篩選只有在 query 中出現且詞頻數大於 500 的詞。

函式解釋

1. initial_p() : p(wi|tk) 和 p(tk|dj) 初始化，用 random&normlize 來實現機率和為 1
2. get_pwd_pwbg(): 取得 p(wi|dj) 和 p(wi|BG)
3. get_pwt_ptd(): 給定 topic 與 it 數量，取得對應的 p(wi|tk) 和 p(tk|dj)
4. plsa(): 將最終 p(wi|tk)和 p(tk|dj)的結果與傳入的 alpha、beta 計算積分並排序

參數調整 (最終分數使用參數)

1. 總字數嘗試過 226-4126 字 （4126）
2. Topic 數嘗試過 5,8,16,32 （16）
3. Max_iter 固定 30 或 50 （30）
4. Alpha,Beta 0~1 之間皆嘗試過 （a=0.3, b=0.1）

運作原理

1. 給定 topic 數與過濾後的 word 和 document 生成初始化的 p(wi|tk) 和 p(tk|dj)
2. 給定 max_iter 反覆執行 em_step 讓 p(wi|tk) 和 p(tk|dj)收斂
3. 取得 p(wi|dj) 和 p(wi|BG)
4. 依序讀取 query，給定 alpha,beta 計算 query 中的詞與各文件的機率積
5. 將排序後的結果取 1000 篇匯出

心得

這次實作的困難點一開始是計算量過於龐大，後來先過濾到剩下 query 中的字過了 baseline 分數後，變開始擴大使用的總字數，但最終還是沒有嘗試最大字數，因為 4000 多字依賴 jit 算也要一點時間了，這次最大的收穫應該是節省了不少原本用 list 運算所浪費的時間，對 em step 的流程也更加的清楚了。