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。