IR HW2 Report

B05705025 資管四 陳漢威

1. 執行環境:

VSCode, terminal

2. 程式語言

Python3

3. 執行方式

程式會讀入在同一資料夾內路徑為”IRTM”的資料夾，並開啟裡面所有的檔案作為 df 計算的依據，程式最終會產出符合 json 格式的資料: “dictionary.txt”, “Doc1.txt”, “Doc2.txt” 於執行檔同一個目錄中。
再執行時最後會印出 Doc1.txt 與 Doc2.txt 的 Cosine Similarity
該次使用的套件有(編譯前須先 pip install):

- nltk (for Stemming and stopwords)
- re (for filtering the symbol)
- numpy (for vector operation)
- collections (for terms count)
- pattern (for lemmatization)

4. 作業處理邏輯說明:

讀入路徑中所有文字檔的同時會進行前處理與切詞、詞性還原等動作，並利用 Counter() 去計算每個 document 中 terms 集合的加總(即 document frequency)，並將該計算結果用 json.dump()存成 txt 檔(之後可以直接 load 為 dictionary class)。
計算完之後再依據 df 與總文章數去計算每個字詞的 idf，並存成 dictionary:{ term : idf}
最後，再依據每篇文章去計算它各個 terms 的 term frequency 並與其對應的 idf 相乘，獲得 tf-idf 的向量，每篇文章的向量會依據編號順序，各自存為 tf_idf(list)的 elements。
之後我會將前兩篇文章的向量分別 dump 成”Doc1.txt”與”Doc2.txt”。
最後，我會將兩個檔案的 file path 丟入我寫的 cosine_similarity(docX_path, docY_path) 函式去計算兩筆向量的內積與餘弦相似度，並印出再 terminal 中。
Output 結果如下:

```
s12 = cosine_similarity("Doc1.txt", 'Doc2.txt')...

Cosine Similarity of document 1 and 2: 0.2610757485999141
```