文字探勘初論 功課一 B10705054 劉知微

1. 執行環境：

Google colab

2.程式語言：

Python 3

3. 執行方式：

非原生套件:
-From google .colab import drive:讓 Google Drive 中的文件可以在 google.colab 中使用。
-from sklearn.feature_extraction.text importTfidfVectorize: 用於文本轉化為 tf-idf 的特徵矩陣，。
-from sklearn.metrics.pairwise import cosine_similarity：用於計算文本之間的余弦相似性。
-import numpy as np:-
原生套件:
-引入 python 標準庫 os，用於操作檔案和資料夾，方便讀取或寫入檔案。
執行輸出:
-點擊 google colab 中的執行階段，點全部執行或者按 ctrl+F9 便可執行程式碼，作業要求的 Cosine Similarity 會輸出在 Command Line，TF-IDF vectors 則會存取進雲端中的 TF-IDF-Vectors"。

4. 作業處理邏輯說明：

1)第一個儲存格
-建立一個空的 List documents, 將 folder 中的文件依照順序，從 1~1095，並且將文本內容添加至 documents 中。
-fidf_vectorizer = TfidfVectorizer(lowercase=True, stop_words='english')：創建一個 TfidfVectorizer 物件，用於將文檔轉換為 TF-IDf 特徵矩陣。lowercase=True 表示將文本轉換為小寫，stop_words='english'表示在 TF-IDF 計算中，排除英文常見詞
-tfidf_matrix 從取透過 fidf_vectorizer 物件轉換為 TF-IDF 特徵矩陣的 documents 其中每行代表一個文檔，每列代表一個詞語的 TF-IDF 值。
2)第二個儲存格
-os.makedirs(output_folder, exist_ok=True)：這行代碼用於創建輸出資料夾

-利用 file.write()還有迴圈，將每一個 TF-IDF 向量寫入對應文件（1.vec 和 2.vec…）並儲存至資料夾

3）第三個儲存格

-使用 np.loadtxt 函數從指定路徑讀取兩個向量檔（1.vec 和 2.vec）。

-使用 reshape 函數，將其轉換為形狀為(1, -1)的矩陣，以確保矩陣是 2D 的。

-cosine_similarity 函數來計算兩個矩陣（這裡是 matr1 和 matr2）之間的餘弦相似度。

最後計算出 Cosine Similarity between Document 1 and Document 2:

0.1999136150897865