資訊檢索語文字探勘導論 PA2

1. 執行環境：Jupyter Notebook
2. 程式語言：Python 3.10.9
3. 執行方式

   （1）在 terminal 使用 pip install nltk 和 re 套件

   （2）使用 Jupyter Notebook 執行 pa2.ipynb

4. 處理邏輯

   （1）讀入所有 data 資料夾中的所有 txt 檔

```
files = listdir(FILE_PATH)
files.sort(key=lambda x: int(x[:-4]))
doc_set = list()

for file in files:
    with open(FILE_PATH + file, "r") as f:
        document_id = str(file)[:-4]
        document = f.read()
        doc_set.append([document_id, document])
```

   （2）先寫出「前處理」、「計算 tf、df 數量」、「計算 tf 向量」、「計算 tf-idf 向量」和「計算 cosine similarity」的函數

   （3）利用「計算 tf、df 數量」的函數計算出各文章的 tf、所有的 df，並將 df 寫入 dictionary.txt

```
tf_list, df_list, t_index = count_tf_df(doc_set)

with open("dictionary.txt", "w") as f:
    f.write("t_index\tterm\tdf\n")
    for term in df_list:
        index = t_index[term]
        key = term
        df = df_list[term]
        f.write(f"{index}\t{key}\t{df}\n")
```

（4）計算 tf 向量後，利用其結果計算 tf-idf 向量，並分別寫入 DocID.txt

```
tf_vector = tf_vec(tf_list, t_index)
tf_idf_vector = tf_idf_vec(tf_vector, df_list, t_index)

for vector in tf_idf_vector:
    doc_id, vec_list = vector
    terms_num = np.count_nonzero(vec_list)
    with open(f"./output/{doc_id}.txt", "w") as f:
        f.write(f"{terms_num}\n")
        f.write("t_index\ttf-idf\n")
        for i in range(len(vec_list)):
            if vec_list[i] != 0:
                f.write(f"{i}\t{vec_list[i]}\n")
```

（5）計算 doc 1 和 doc 2 的 cosine similarity

```
    print("cosine similarity of doc 1 and doc 2 is ", cosine("1", "2"))
  ✓ 0.0s
cosine similarity of doc 1 and doc 2 is  0.19986585359571019
```

結果為 0.19986585359571019