1. 執行環境

   Jupyter Notebook

2. 程式語言（請標明版本）

   Python 3.7

3. 執行方式（重要!!!!!!!）

   以 Jupyter Notebook 執行

4. 作業處理邏輯說明

   環境：

   - import 會用到的套件，以及設定好檔案路徑
   - 檔案順序我到最後一刻才發現有誤，導致原本文件跟新的 DocID 對應不起來，沒有加這一段的話，順序是: 1,1000,100,10,…而不是 1,2,3,4…，因此用這段來找出是檔案路徑末段包含數字的部分(natural key 函數所做)，轉成數字(atoi 函數所做)

   第一步：

   - 延續作業一，只不過 replace 的符號有變多，還有數字的部分
   - 從 replace, split, stemming, 製作 stopword list, 到 remove stopwords 基本上延續作業一，多加了一個 collections 套件中的 Counter 函數是用作計算詞頻(tf)，且可以拉出單獨的 term，結果呈現為{(termA, tf), (termB, tf)}
   - 若在整個 collections 中再使用一次 Counter 函數，則可以計算出 document frequency (df)
   - 看 dictionary 的長度
   - 製作一個 dict()，key 是 t_index, value 是(termA, df)
   - 寫成檔案

   第二步：

   - 寫一個把文件轉成 tfidf 的函數(transfer_doc_to_tfidf)，我想說可以先把計算過程都留下，trytry 看那一格可以看到所有計算過成的數字，包含 t_index, term, df, idf, tf, tf_idf, 正規化過的 tf_idf
   - 先用個 for 迴圈先把所有檔案寫起來，寫說 DocID 有多少 terms
   - 再用個 for 迴圈，呼叫 transfer_doc_to_tfidf，輸入路徑(direc[a])，此時函數會回傳剛 trytry 過的所有欄位跟值，但我只要 t_index, term, Norm_tf_idf，加在剛開剛開好的後面

   第三步：

   - 寫一個算 cosine similarity 的函數，輸入兩個檔案路徑，輸出相似度

5. 繳交格式 PDF 檔

執行方式的部分請輔以截圖說明

	t_index	term	df
1	1	wa	892
2	2	ha	753
3	3	said	729
4	4	hi	663
5	5	thi	626
6	6	say	593
7	7	one	590
8	8	state	570
9	9	two	569
10	10	presid	547
11	11	last	503
12	12	year	501
13	13	offici	496
14	14	would	493
15	15	peopl	483
16	16	time	478
17	17	also	477
18	18	day	460
19	19	first	433
20	20	kill	431
21	21	countri	411
22	22	new	401

dictionary 的結果

有兩個字的，我嘗試用 re.sub(r'¥b¥w{2}¥b', ' ', f)，想去掉，但不知道為何不成功，會再試試看

	t_index	term	df	idf	tf	tf_idf	Norm_tf_idf
51	291	milosev	134	0.912309	6	5.473856	0.300925
66	461	belgrad	100	1.039414	5	5.197071	0.285709
75	555	strike	85	1.109995	4	4.439981	0.244088
121	2918	maceda	11	1.998021	2	3.996043	0.219683
40	176	opposit	186	0.769901	5	3.849506	0.211627
102	1173	tomorrow	41	1.426630	2	2.853261	0.156858
93	947	jim	51	1.331844	2	2.663688	0.146436
85	759	regim	64	1.233234	2	2.466468	0.135594
84	748	nbc	65	1.226501	2	2.453002	0.134854
129	4839	blackout	4	2.437354	1	2.437354	0.133994
81	667	radio	72	1.182082	2	2.364163	0.129970
127	4358	twohour	5	2.340444	1	2.340444	0.128666
128	4359	tast	5	2.340444	1	2.340444	0.128666
78	656	pressur	74	1.170182	2	2.340365	0.128662
125	3990	poorli	6	2.261263	1	2.261263	0.124313
126	3991	shrink	6	2.261263	1	2.261263	0.124313
124	3989	stoppag	6	2.261263	1	2.261263	0.124313
74	551	serbia	86	1.104916	2	2.209831	0.121486
18	45	work	317	0.538355	4	2.153419	0.118384
123	3438	rid	8	2.136324	1	2.136324	0.117444
122	3437	wealth	8	2.136324	1	2.136324	0.117444

Trytry 的結果

有很多欄，但是可以看到計算過程，我想這樣比較好

心得：

雖然寫得沒日沒夜的但是過程蠻有趣的，作業二花得比想像中還要多時間，下次要更早點做作業！

另外，會盡快在作業三出來以前把第一步沒有清乾淨的東西處理好，雖然前面試了很多次但是都沒有成功，可能要求助一下，參考別人的作法。