# HW4 Report

R07725021 資管碩1 洪靖雯

## 1.執行環境

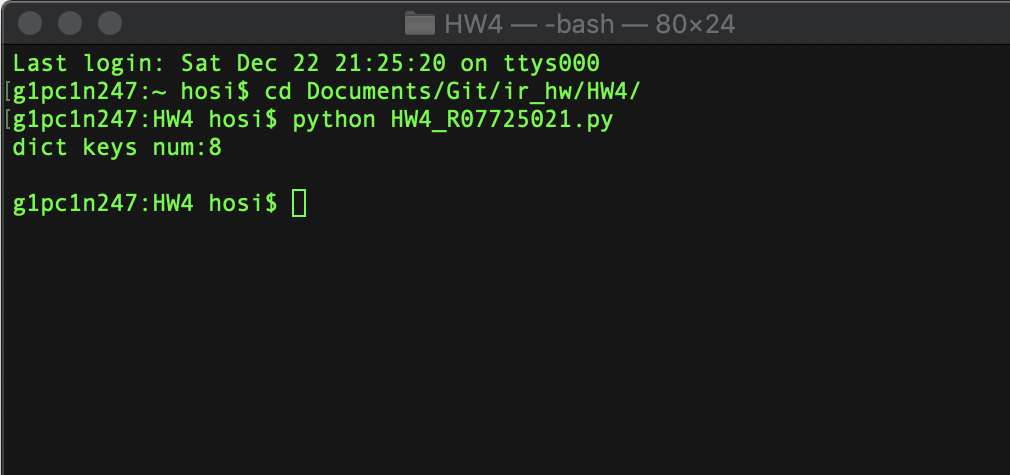
* Sublime
* Cmder

## 2.程式語言

* Python 3.6.6

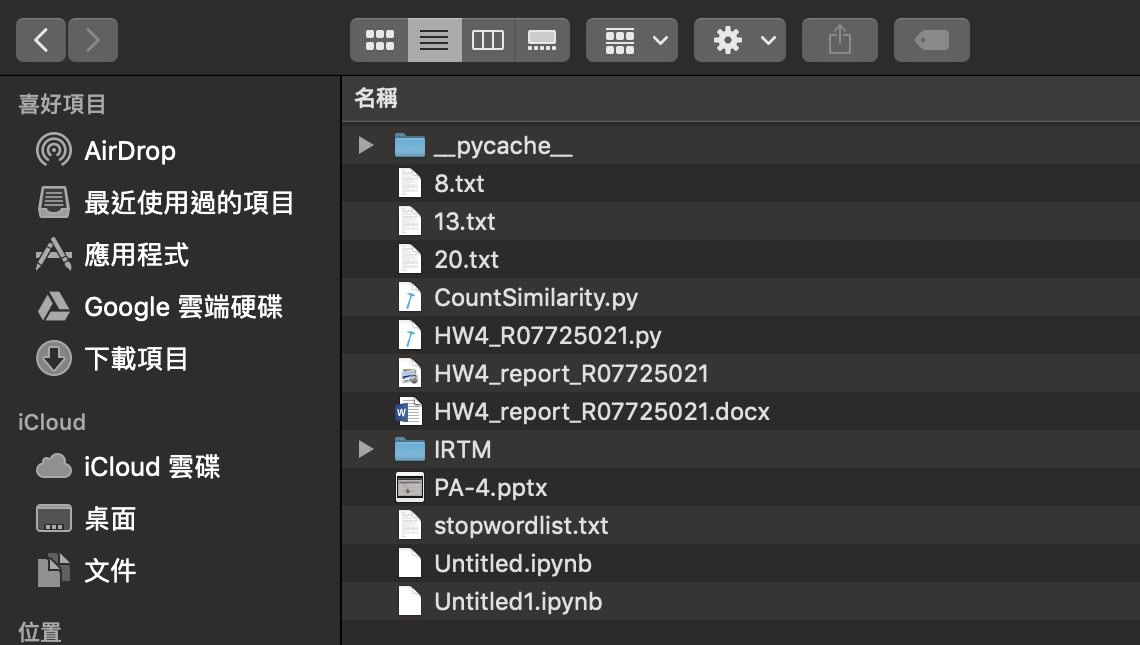
## 3.執行方式

* 套件: nltk、bs4、defaultdict、numpy
* 編譯方式: 使用python HW4\_R07725021.py即可

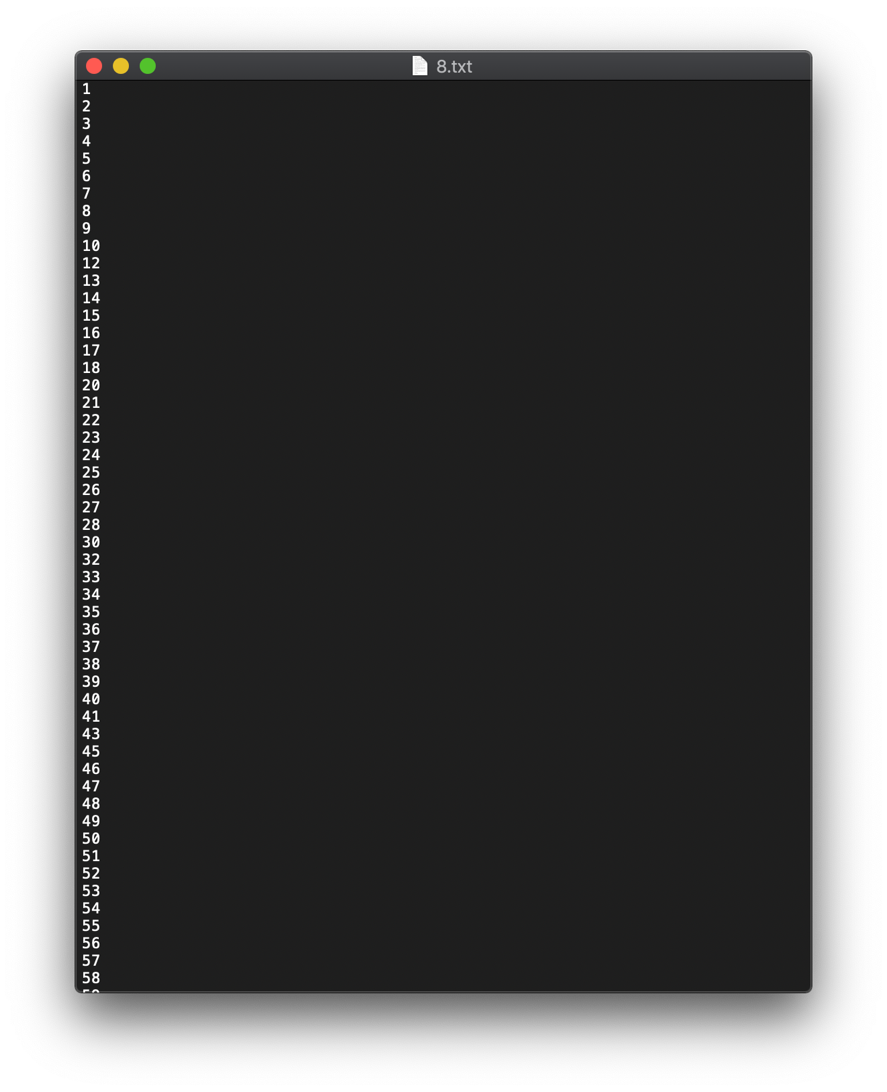


* 輸出結果

1. x.txt (總共有三個檔案，分別是群體數目為8、13、20個的結果)



1. 8.txt結果示意圖



## 4.作業處理邏輯說明

本次作業有兩個檔案 - CountSimilarity.py及HW4\_R07725021.py，以下分別介紹：

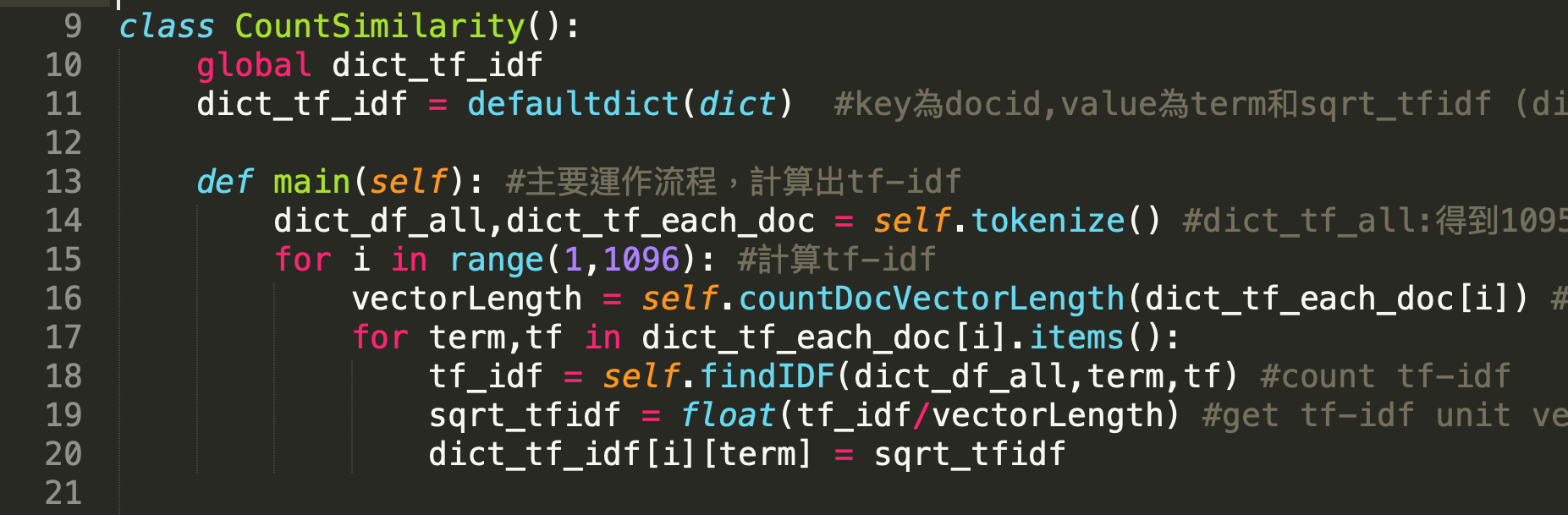
1. **CountSimilarity.py**

CountSimilarity.py其實是把HW2的程式寫成Class，供主程式呼叫。主要做的事情為將1095個document做tokenize，個別計算normalized tf-id vectors，最後輸入任兩文章docid，便可得到這兩個文章的cosine similarity。

* Import必要的套件



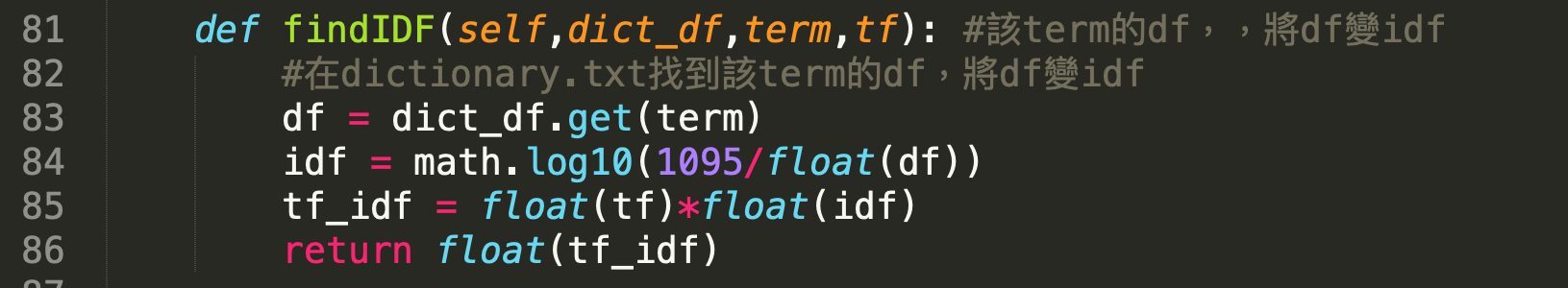
* 各文章tokenize，計算各文章各個term的normalized tf-idf vectors值，用nested dict方式儲存，key為**docid**, subkey為**term**, value為**tf-idf**值。
  + 主程式

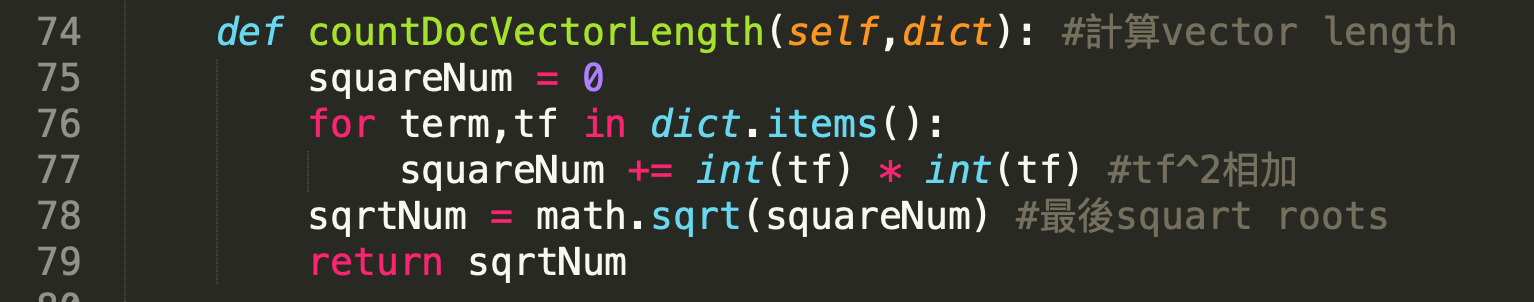


* Tokenize

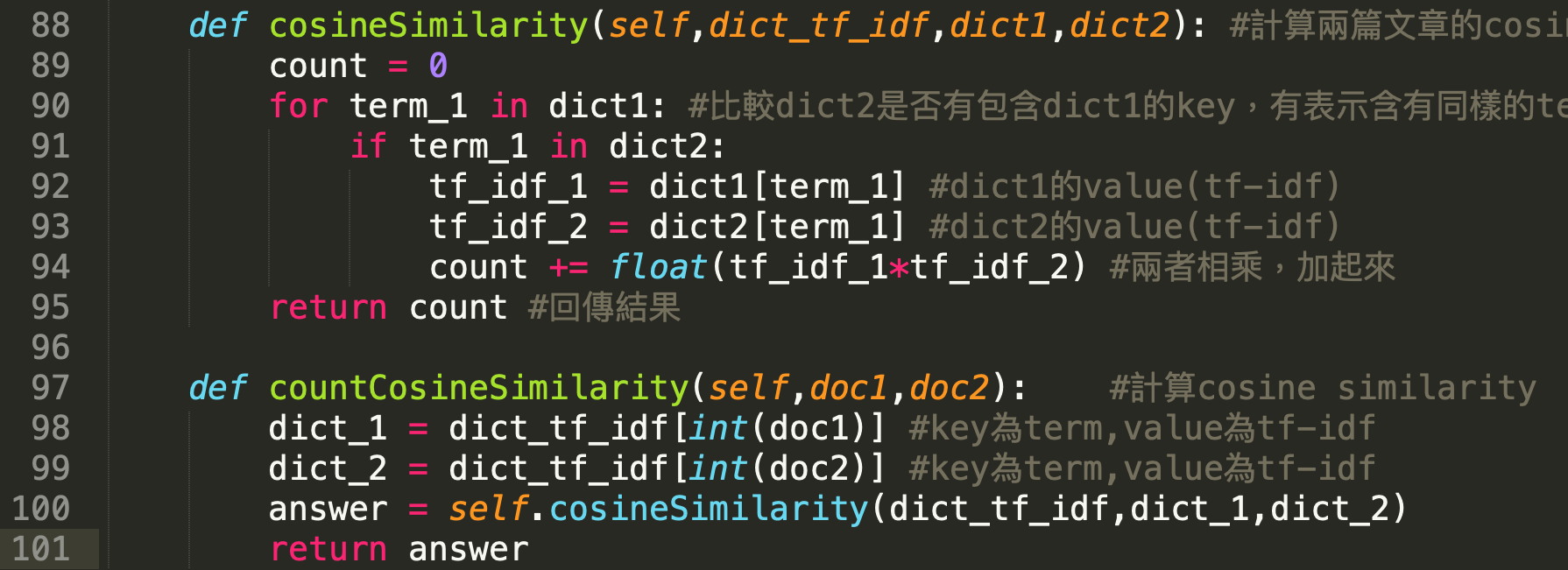


* 計算tf-idf，再計算vector length，得到normalized tf-idf vector





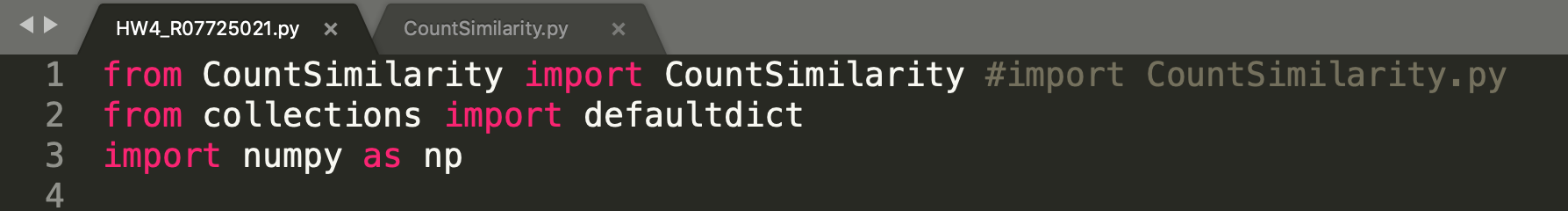
* 最後再給任兩個docid，計算這兩篇文章的cosine similarity並回傳值



1. **HW4\_R07725021.py**

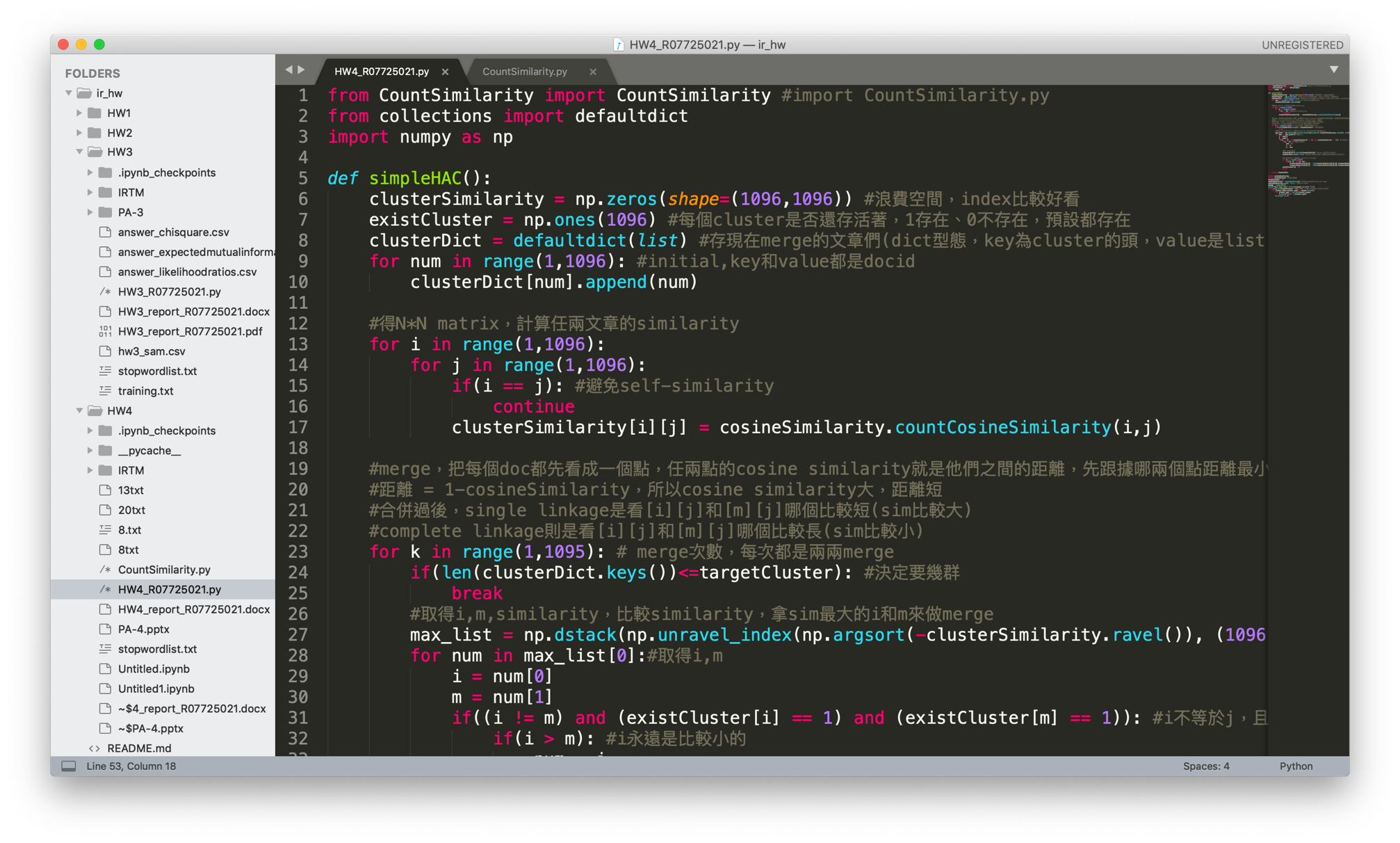
實作bottom up HAC，並輸出群數為8群、13群及20群的結果

* Import必要的套件，以及CountSimilarity.py Class

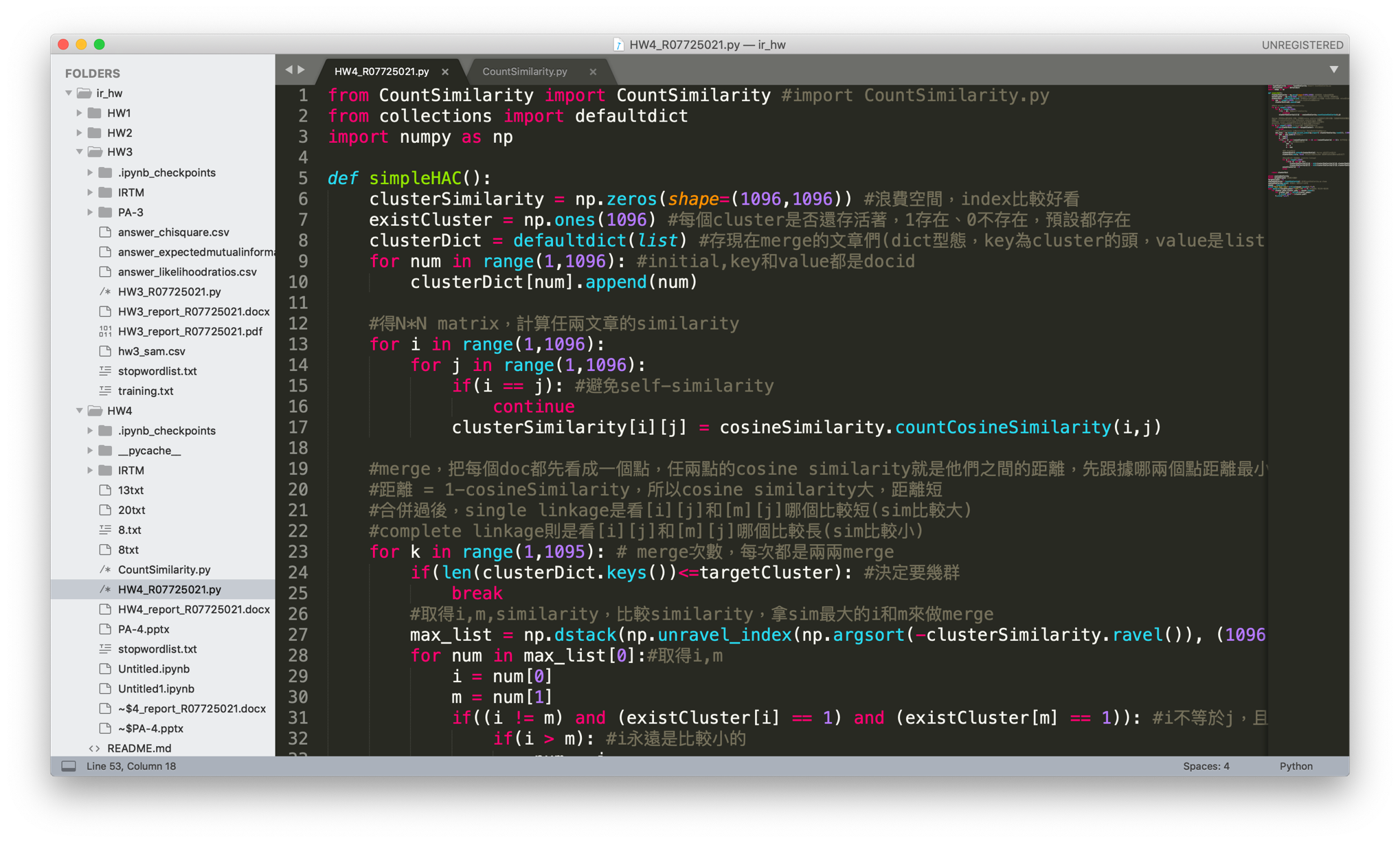


* HAC部分
  + 先將一些必要值、array做initialize

1. clusterSimilarity為numpy array，為2D ndarray，1096\*1096 matrics，因為index想要直接是docid，所以多了一維度浪費掉。
2. existCluster為現在docid是否還存在，1為存在，0表示已經merge掉。
3. clusterDict是儲存現在分群的狀態，預設是key和value都是docid。

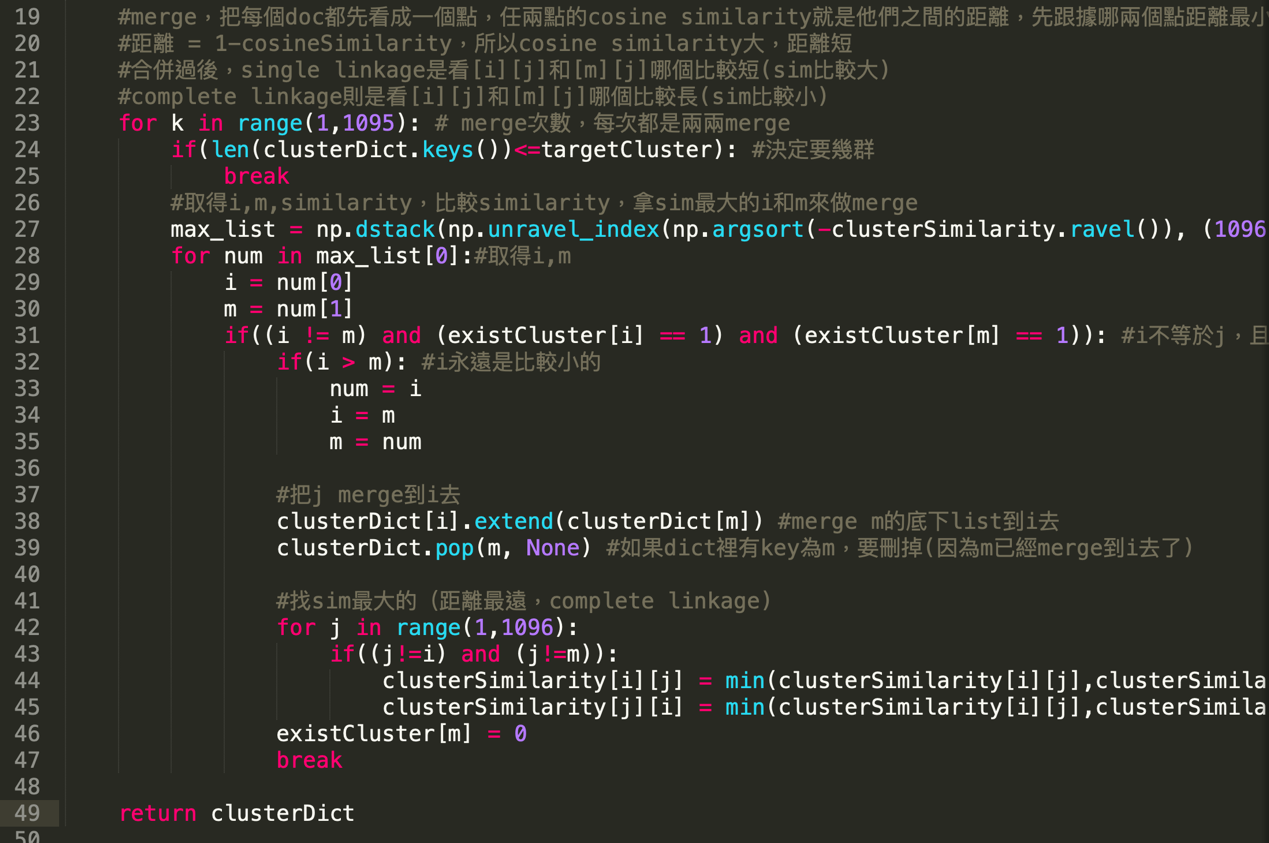


* 再從CountSimilarity.py那得到1095篇文章任兩篇的cosine similarity



* 接著是做merge的動作

1. k為merge次數，每次為兩兩node做merge，總共merge N-1次。
2. targetCluster為我們預計要分幾群，是看clusterDict的keys剩多少，一開始為1095個keys，慢慢減少成我們要的群數(bottom up)
3. 將上面的clusterSimilarity用numpy array裡的argsort得到這個matrics中similarity最大的indices，用max\_list儲存，並從大到小做排序。
4. i和m各為每個indices中的index[0]和index[1]
5. 判斷i是否不等於m，且doc i和doc m是否都還活著
6. 將m merge到i中，並將m在clusterDict資料刪除
7. 接著更新其他與i有關的j的similarity。這裡是用**complete linkage**的方式去做。最一開始每個document都是一個點，所以任兩個document的cosine similarity便是他們間的距離。而complete link看兩個cluster中最遠的點間的距離，所以更新i與j的間的距離，便是比較i對j與m對j間的complete link誰比較短，就取比較短的距離！(min)
8. 最後把m的existCluster設為0
9. 全部運算完後回傳clusterDict為答案



* 將答案依據目標群數，輸出成x.txt (8,13,20群)

