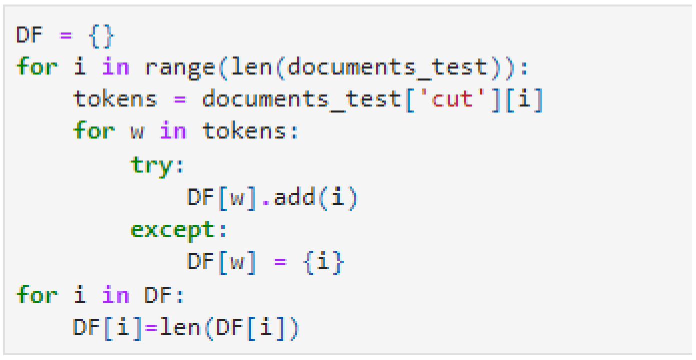
1. 把 document 的檔案讀取進 dataframe
2. 計算 DF(單字出現在那些 document)



```
DF = {}
for i in range(len(documents_test)):
    tokens = documents_test['cut'][i]
    for w in tokens:
        try:
            DF[w].add(i)
        except:
            DF[w] = {i}
for i in DF:
    DF[i]=len(DF[i])
```

3. 把 count 的檔案讀取進 dataframe
4. 逐一比對並排序(compare function)

   def compare2(s1_cut,s2_cut,DF,co):

   s1_cut:query 分詞過後

   s2_cut:document 分詞過後

   DF:單字出現在那些 document

   co:計數達一定量釋放記憶體

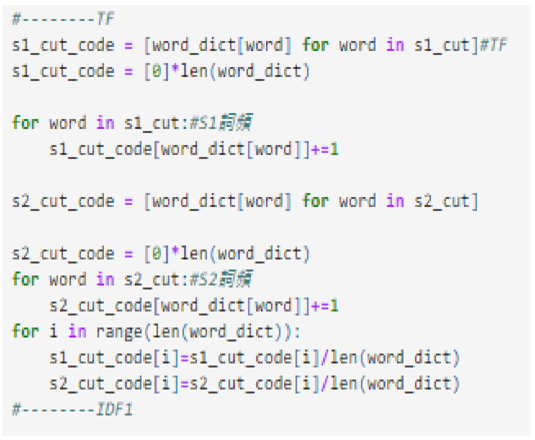
   4.1: s1,s2 union(取兩個比較的文字集合建立 TF 計算的 dict)

   4.2:s1,s2TF 計算

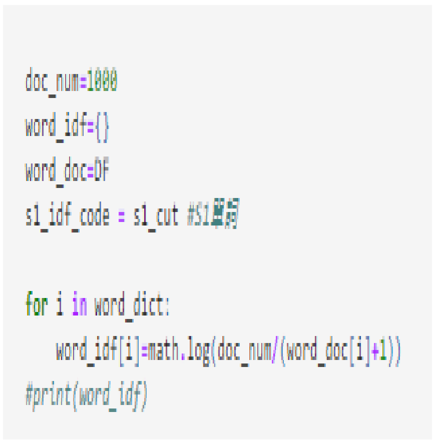
   4.3:s1,s2IDF 計算

   4.4:TF*IDF

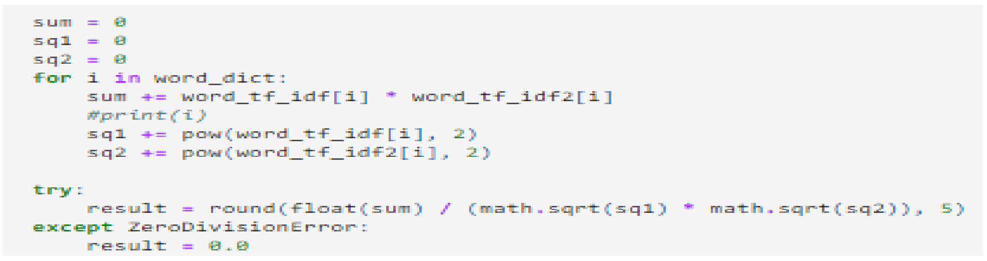
   4.5:計算 cosine



```
#--------TF
s1_cut_code = [word_dict[word] for word in s1_cut]#TF
s1_cut_code = [0]*len(word_dict)

for word in s1_cut:#S1詞頻
    s1_cut_code[word_dict[word]]+=1

s2_cut_code = [word_dict[word] for word in s2_cut]

s2_cut_code = [0]*len(word_dict)
for word in s2_cut:#S2詞頻
    s2_cut_code[word_dict[word]]+=1
for i in range(len(word_dict)):
    s1_cut_code[i]=s1_cut_code[i]/len(word_dict)
    s2_cut_code[i]=s2_cut_code[i]/len(word_dict)
#--------IDF1
```



```
doc_num=1000
word_idf={}
word_doc=DF
s1_idf_code = s1_cut #S1單詞

for i in word_dict:
    word_idf[i]=math.log(doc_num/(word_doc[i]+1))
#print(word_idf)
```



```
sum = 0
sq1 = 0
sq2 = 0
for i in word_dict:
    sum += word_tf_idf[i] * word_tf_idf2[i]
    #print(i)
    sq1 += pow(word_tf_idf[i], 2)
    sq2 += pow(word_tf_idf2[i], 2)

try:
    result = round(float(sum) / (math.sqrt(sq1) * math.sqrt(sq2)), 5)
except ZeroDivisionError:
    result = 0.0
```