資料檢索與文字探勘導論_HW2_Report

B06705030　　資管三　　蕭昀豪

1. 執行環境:

Jupyter Notebook

2. 程式語言 (請標明版本):

Python3.7.3

3. 執行方式:

。使用的非原生套件:

- punkt　　(原生套件 nltk 的衍生套件)
- stopwords　(原生套件 nltk 的衍生套件)

在 Source Code 中，我已附上安裝指令，如下圖

```
In [ ]: #download the necessary packages
        import nltk
        nltk.download('punkt')
        nltk.download('stopwords')
```

如果您們沒有 nltk，可依以下指令安裝

pip install --user -U nltk

4. 作業處理邏輯說明:

【程式架構】

延伸上星期所完成的 termMachine 物件，這次作業我實作出一新物件 Dictionary。它所包含的成員變數如下:

```
self.documentNum = 0
self._dictTreeRoot = None
self._dictionary = DICT_TREE()
self.source = [collection]
```

其中，documentNum 儲存了目前該 Dictionary 讀取的文件總數，以便順利計算出 idf。另一方面，以在網路上所找到的 AVL tree 程式碼為基礎，我新創一 DICT_TREE 類別繼承 AVL tree，並配合代表 root 的_dictTreeRoot，組合出適合本次應用的資料型態，儲存截取下的全部 term。最後，由 source 記錄下使用者輸進 Dictionary 的文件來源。值得一提的是，我在 source 可允許的資料型別上做了擴充，它容許的輸入包含以下型別。

- 資料夾的路徑: Dictionary 將遍歷該資料夾下所有檔案，抽取其中所有 terms
- 檔案路徑 : Dictionary 將抽取該檔案中所有 terms
- 字串 : 當輸入型別為 str，但不為資料夾或檔案路徑的字串時，會將該字串視為使用者自定義的 query，Dictionary 將抽取其中所有 terms
- 成員皆屬於以上三項類別的串列。

【程式邏輯】

為了實現讓 source 可以容許上述四種型別的輸入，我於 Dictionary 中實作一成員函式 whichMethod，該函式會根據傳入的來源型別，自動判讀該回傳哪一項對應的處理函式，我們僅須利用回傳函式，就能順利將來源檔案中的 terms 更新入 Dictionary 當中。

更新 term 進入 source 的過程中，最關鍵的函式屬 update，每一次當它被呼叫，它會立即將 term 串列更新至 DICT_TREE 型別的_dictionary 當中，並將 documentNum 加一。由於 AVL tree 的特性，這樣的更新方式能輕易維持 Dictionary 中各個 term 的排序。

5. cosine 函式處理邏輯:

我的程式包含處理單一文件的 termMachine，與處理複數文件的 Dictionary。首先，因為 tf 的運算僅涉及文件本身，我在 termMachine 擴充 tf_dict 的屬性。當我們存取 tf_dict，即可得到以 term 為 key，tf 為 value 的 dict。

接著我們在 Dictionary 中實作函式 cosine(Doc1, Doc2)。在這一步驟中，我認為我們字典應該事先建好並擁有足夠大的詞彙，所以並不在運算向量內積前驗證字典是否包含新文章的所有詞彙。我們先存取 Doc1 與 Doc2 的 tf_dict，算出對應的 tf_idf。最後進行向量內積並將結果化成單位向量回傳。

6. 擴充功能:

- 多型別來源:

  我在 source 可允許的資料型別上做了擴充，它容許的輸入包含以下型別。

  - 資料夾的路徑: Dictionary 將遍歷該資料夾下所有檔案，抽取其中所有 terms
  - 檔案路徑 : Dictionary 將抽取該檔案中所有 terms
  - 字串 : 當輸入型別為 str，但不為資料夾或檔案路徑的字串時，會將該字串視為使用者自定義的 query，Dictionary 將抽取其中所有 terms
  - 成員皆屬於以上三項類別的串列。

- 手動更新:

  Dictionary 無須在創建時即明確所有的 source，使用者可使用 Dictionary.addNewSource(newSource)讓 Dictionary 新增 newSource 中的 terms，同時期 documentNum 將隨之增加。其中 newSource 與宣告時一樣，支援四種型別的輸入。

- Dictionary 間的合併:

  兩個 Dictionary A 和 Dictionary B 可以使用+或+=進行合併。合併後的新 Dictionary 它所包含的 term 涵蓋 A 與 B 所有的 terms。當某一 term 僅屬於 A 時，它的 tf 與原先相同；當某一 term 屬於 A 和 B 時，它的 tf 與為該 term 在 A 與 B 的 tf 總和。

7. 注意事項:

   source 處理議題:

   該議題共有以下兩個面向:

   - 當使用 Dictionary.addNewSource(newSource)時，Dictionary 容許 newSource 與現有 source 重複。
   - 當兩辭典合併，新的 Dictionary 並不會檢驗某一 source 是否擁有相同來源，在 A 和 B 共有同一來源的情形下。新的 Dictionary 將重複計算該來源 term 的 tf，同時新 Dictionary 的 documentNum 也將重複計算重複 source 的次數。

不進行處理的原因在於，假如我在時間點 a 抽取了 Doc.txt 的 term，同時又在時間點 b(b > a)更新了 Doc.txt 的 term 進入字典中，我希望在使用者於 a ~ b 間對 Doc.txt 進行改動的情形下，我的 Dictionary 能將 Doc.txt 被改動前後包含的 term 都收錄進來。