Rocchio Algorithm
資管碩一 M10909112 石家安

運用工具 & 資料前處理說明

1. 透過 nltk 的 word.tokenize function，將 Documents 與 Queries 進行斷詞。
2. 運用 snowball_stemmer function 將詞性還原。
3. 使用 nlt.corpus 中 stopwords 篩選掉不帶有資訊的字詞，ex: the, a, and…。

第一次檢索：rocchio_bm25.py、bm25_main.py

```
for wc in doc_dict:
    # IDF
    self.docidf[wc] = self.docidf.get(wc, 0.0)+1.0
    # TF
    doc_dict[wc] = 1+math.log(doc_dict.get(wc, 0.0),2)
```

1. Documents TF 計算：
   計算該文章每個字出現的次數，並運用 Log Normalization (1+log(tf))概念平滑 TF 參數，以完成該 doc 各文字的 TF 計算。
2. Query TF 計算：
   計算 query 各文字出現的次數則是運用最原始的 TF 計算方式。
3. IDF 計算:
   為最原始的 IDF 計算方式：log((N-ni+0.5)/(0.5+ni))

```
for w in self.queries[queryName]:
    if w in docTFtemp:
        ctd = 0.8*docTFtemp[w]/((1-b)+b*len/avglen)
        first = ((K1 + 1) * (ctd+delta)) / (K1+ctd*0.7)
    else:
        first = 0.0
    if w in self.queries[queryName]:
        second = ((K3+1) *self.queries[queryName][w]) / (K3 + self.queries[queryName][w])
    else:
        second = 0.0

    score += first*second*math.log10((30000-self.docidf[w]+0.5)/(self.docidf[w]+0.5))
    # score2 += first*second*math.log10((30000-self.docidf[w]+1)/(self.docidf[w]+1))

bm11 = K2*qalllen*abs(avglen-len)/(avglen+len)
score+=bm11
```

第一次檢索為 BM25L 結合 BM11，經由實驗參數設定結果如下：
K1 = 0.8、b = 0.7、K2 = 0.07、K3 = 0.2、delta = 0.2，
此外，由於「平均長度-長度」有可能為負值導致所求分數減少，因此我試套上絕對值後準確度的確會提升。再者，在實作 tfi' j 時前面有多 X 0.8，該參數亦是經由實驗而進行設定，而我認為該參數設定是要將 Doc Term Frequency 的值更平滑。

第 2 次檢索：rocchio_vsm.py

使用上述 bm25 之模型找出的每個 query 前 10 篇相關的 relevant doc，將上述所得之結果套用至第 2 次檢索，而我所使用的第 2 次檢索之模型為 vsm，套用的 tf 平滑方法同上，idf 為 log(N/(ni+1))。經由實驗結果，拿前 10 篇相關的 relevant doc，並以參數 α=1， β=0.2 為最好的實驗結果。

Rocchio Algorithm
資管碩一 M10909112 石家安

心得
其實從上次作業開始，難度就爆炸性上升，感覺自從開始教 PLSA 後，就有點跟不上進度了...，而且感覺後面幾次的作業關聯程度都很高，故由於上次作業的 EM 就已經沒做出來了，我自己後來也有再嘗試研究並理解，但因概念上就有點抽象了，還需將其轉換程式碼真的很困難，故這次作業我一樣沒有用到那些較進階的 model (ex: SMM、RM…etc)，只運用 Rocchio 方法。此外，由於我第一次檢索是用 bm25，第 2 次檢索是用 vsm，我也曾嘗試過將第 2 次檢索改成 bm25，但得出的準確率非常低，而因為準確率過低的關係，可得知我第 2 次 bm25 之檢索方式一定有寫錯之地方，雖然我目前還是不知道錯在哪裡，但慶幸的是有把 VSM 的做法做出來。 當然，從看大家連過 Baseline 都很難開始，外加幾位認識的資工同學也是很苦惱，甚至老師還把繳交作業的時間延長。就可以看出這次作業難度確實很高。但同時也看的出來有很多人在後來延長的第 3 周有成功做出來，大家的分數都突然變得好高! 雖然這時看到其實也是蠻挫折的，因為感覺有很多人在最後有做出來進階型的 model，又或是他們都有把 Rocchio 做得更準確，但我自從第一周做完 Rocchio 就一直停滯了，所以我很希望這次的作業老師可以找更多人上台分享他們怎麼做，希望藉由同學上台的分享可以讓我更了解。