IR HW2 r07922003 劉濬慶

VSM

Step1 -> 讀 query 中每篇的 concepts，只有分 bigram

ex: 流浪狗、流浪犬、動物保護、動保法

奇數切詞位移量=1，流浪狗 -> 流浪、浪狗

偶數切詞位移量=2，動物保護 -> 動物、保護

Step2 -> 從 query 中的 trem 去對應到每個 document 去

看在 inverted-file 裡面所記錄的:

1. 那個 query term 出現在幾個 document 中
2. 那個 query term 在此 document 中出現幾次

Step3 -> 計算 IDF(w) = log(m+1/k)

m = total number of documents

k = numbers of docs with term t

Step4 -> 計算 TF(W) ，有使用兩種做法

1. Pivot BM25/Okapi (Parameters: b,k)

TF(t,d) = (k+1)*count(t,d) / (count(t,d) + k(1−b+ b * |d| / Avg Doc Len))

2. Pivoted Length Normalization VSM (Parameter: b)

TF(t,d) = ln[1+ln[1+count(t,d)]] / (1−b+b * |d| / Avg Doc Len)

Step5 -> return term 分數 IDF * TF

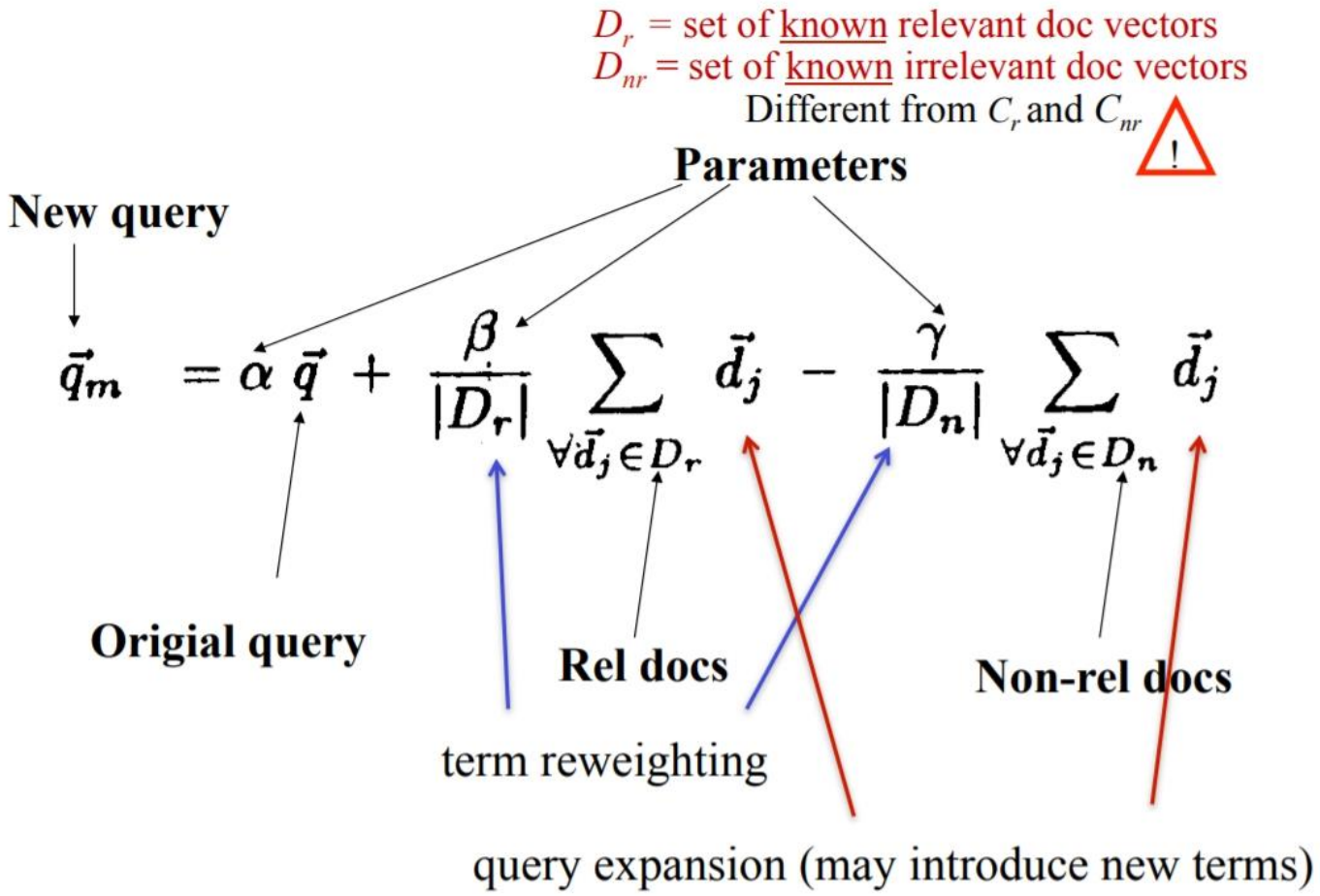
Rocchio Relevance Feedback

Parameters: (ratio,alpha,beta,gamma)

ratio -> 將已經 sum 好排序的 ranking 中取前幾%當作 relevant documents

，剩下的當作 Non-relevant documents

alpha,beta,gamma -> 為對應下面的參數



Results of experiment

b = 0.2，no feedback，use Pivoted Length Normalization VSM

Score = 0.80411

---

b = 0.2，use Pivoted Length Normalization VSM

Rocchio Feedback parameters

ratio = 0.1，alpha = 0.9，beta = 0.3，gamma = 0.4

Score = 0.79800

---

k = 0.7，b = 0.2，Pivot BM25/Okapi

#Rocchio feedback parameters

ratio = 0.05，alpha = 0.7，beta = 0.3，gamma = 0.2

Score = 0.79363

---

k = 0.8，b = 0.3，Pivot BM25/Okapi

#Rocchio feedback parameters

ratio = 0.02，alpha = 0.8，beta = 0.3，gamma = 0.5

Score = 0.79386

---

b = 0.3，with Pivoted Length Normalization VSM

#Rocchio feedback parameters

ratio = 0.02，alpha = 0.8，beta = 0.3，gamma = 0.5

Score = 0.79129

---

b = 0.4，no feedback，use Pivoted Length Normalization VSM

Score = 0.78286

With Feedback vs. without Feedback

With Feedback 較不好，without Feedback 較好

Other experiments you tried

TF 有用三種算法:一種基本的 Okapi/BM25 without normalize，Pivot BM25/Okapi，Pivoted Length Normalization VSM

Discussion

更進一步了解 query 跟 document 的 term 的關係，VSM model 和 Rocchio feedback 的演算法，分數要高的話最好需要 normalize，有 feedback 的分數會降低。