Web Retrieval and Mining Spring 2019
Programming Assignment 1 - VSM Model
Build a document retrieval system!

R07946007, Ting-An Chen, 陳庭安

0. Rules

Please write a report.pdf file and submit it.
● The report should contain the following:
○ (2%) Describe your VSM (e.g., parameters….)
○ (2%) Describe your Rocchio Relevance Feedback (e.g., how do you define relevant documents, parameters…)
○ (3%) Results of Experiments, including:
  ○ MAP value under different parameters of VSM
  ○ With Feedback vs. without Feedback
  ○ Other experiments you tried
○ (1%) Discussion: what you learned from the work.

1. VSM – Okapi BM25

在此採用下方公式作為 Queries 及 Documents 表示向量之權重。

$$ln\frac{N-df+0.5}{df+0.5}\cdot\frac{(k_1+1)tf}{(k_1(1-b+b\frac{dl}{avdl}))+tf}\cdot\frac{(k_3+1)qtf}{k_3+qtf}$$

Queries 及 Documents 表示向量之維度取決於共同出現 Terms 個數 N。在此不選擇以詞彙總數為維度數，是為減少儲存空間以及數值零相乘的運算。表示向量權重為每一共同出現 term 對該 query 或 document 的重要程度。上式第一項為 IDF, Inverse document frequency，與 term 出現的 documents 數 df 大小呈反向關係，即文件中一 term 少出現在其他文件數，該 term 對所在文件具高度代表性; 後二項分別為 term 在查詢的 query 與搜索的 document 出現次數 tf, qtf，出現頻率高，同樣具高度代表性。其中 k1, k3 為調整係數，調整函式變化程度，B 為對文件長度 normalization 的程度，值愈大 normalization 程度愈高。

以上述權重計算而得 Query 與 Document 代表向量後，採內積作為兩者之間的相似程度，作為 Documents 間排序的依據，與 Query 相似度愈高，排序愈前面。

$$\vec{Q}=(w_{q1},...,w_{qN}) \qquad \vec{D}_i=(w_{i1},...,w_{iN})$$

$$sim(\vec{Q},\vec{D}_i)=\sum_{j=1}^{N} w_{qj}*w_{ij}$$

Dataset 每一 Query 內容含 Title, Question, Narrative 與 Concepts 四部份，每部份計算上述 BM25 tf-idf 後，分別給予不同加權權重, 作為 Query 表示向量。不同權重組合在訓練集與測試集資料的表現於 3. 討論。

2. Rocchio Relevance Feedback

根據前一次文件相似度排序，取前 100 高相似度的文件作為相關文件，剩餘為不相關文件，並對 Query 表示向量作修正，作下一輪相似度的計算與排序，作 t 輪。

Query 表示向量的修正方式如下式，a, b, c 分別為調整原 Query 向量、相關文件向量以及不相關文件向量長度之係數。新 Query 表示向量傾向朝相關文件向量的方向調整，而遠離不相關文件向量。

$$\overrightarrow{Q_m} = (a \cdot \overrightarrow{Q_o}) + \left(b \cdot \frac{1}{|D_r|} \cdot \sum_{\overrightarrow{D_j} \in D_r} \overrightarrow{D_j}\right) - \left(c \cdot \frac{1}{|D_{nr}|} \cdot \sum_{\overrightarrow{D_k} \in D_{nr}} \overrightarrow{D_k}\right)$$

3. Results of Experiments

(a) Parameters tuning – Rocchio relevance feedback

k1 = 1, k3 = 1, B = 0.75

Feedback 輪數 t	Query 段落權重 [title, question, narrative, concept]	原 Query 向量、相關 / 不相關文件向量長度倍數 (a, b, c)	訓練集 mAP	測試集 (Kaggle) Public scores
0 (No feedback)	[50, 0, 0, 2] [40, 0, 0, 2] [35, 0, 0, 2] [25, 0, 0, 2]	x	0.80864 **0.81087** 0.80999 0.80704	- - 0.79232 0.79161
2	[55, 0, 0, 2] **[50, 0, 0, 2]** [40, 0, 0, 2] [35, 0, 0, 2] [25, 0, 0, 2]	(0.1, 0.0, 25) **(0.1, 0.0, 25)** (0.1, 0.0, 25) (0.1, 0.0, 25) (0.1, 0.2, 25) (0.3, 0.0, 25) (0.1, 0.0, 20) (0.1, 0.0, 30) (0.1, 0.0, 25) (0.1, 0.2, 25) (0.3, 0.0, 25) (0.1, 0.0, 20) (0.1, 0.0, 30)	0.82439 **0.82918** 0.82681 0.82610 0.78961 0.81719 0.82484 0.82594 **0.83268** 0.78855 0.81617 0.82624 0.82547	- **0.80216** 0.80153 0.80067 - - - - 0.80030 - - - -
3	[50, 0, 0, 2]	(0.1, 0.0, 25)	0.82788	-

＊ 以上僅部份實驗結果，如: Query 段落權重僅呈現表現較好的組合。
＊ Kaggle Public scores 只呈現在訓練集表現較好組合之結果。

(b) Parameters tuning – Okapi BM25

t = 2, w = [50, 0, 0, 2], l = (0.1, 0.0, 25)

k1	k3	B	訓練集 mAP	測試集 (Kaggle) Public scores
1 1 1 0.8 1.2 **1.2**	1 1 1 1 1 **0.80**	0.70 0.75 0.80 0.75 0.75 **0.80**	0.82310 0.82918 0.82748 0.82328 0.82191 **0.83043**	0.79607 0.80216 **0.80250** 0.79877 **0.80510** **0.80539**

t = 2, w = [10, 0, 0, 2], l = (0.1, 0.0, 25)

k1	k3	B	訓練集 mAP	測試集 (Kaggle) Public scores
2.6	**0.80**	**0.85**	**0.84218**	**0.80656**

將 Testing public scores > 0.802 各個 Query-documents 的 Similarities 結果平均，再以此分數排序，取分數最高前 100 個 Documents。表現結果最佳，如下表所示：

訓練集 mAP	測試集 (Kaggle) Public scores
0.83433	**0.80751***

4. Discussion / Learned from this work

(a) 在訓練集 Query title 給較大的權重，會有較好的表現，但在測試集的表現卻不如預期，可能 Over-fitting，因此還是別太依賴訓練集的表現而給予過大或過小的參數，使得測試集結果不盡理想。又或者再將資料集切成多份訓練、驗證資料交互驗證，取適當大小的參數值。

(b) 由原 Query 向量、相關文件向量、不相關文件向量調整幅度知，修正後的 Query 表示向量其中含部份原向量的資訊; 相關文件向量資訊對修正並沒有太大幫助，故調整係數值小; 不相關文件向量資訊則尤其重要，表示原 Query 表示向量極可能與不相關文件向量仍有一定程度的相似性，造成檢索系統仍有很高的機會誤選不相關文件。當調整 Query 表示向量以遠離不相關文件向量後，系統選擇到相關文件的準確率提升。

(c) 對文件長度進行 Normalization ，能避免長文件因含有較多詞彙而容易被辨識為相關文件，因而達到較好的表現。

(d) 將表現好的結果平均，儘管在訓練集表現下滑，但在測試集表現卻最佳。平均的結果可有效降低 over-fitting 的情形，也讓預測結果較為穩健、可信。