IR2017 HW2 Program Assignment

r05922094 葉俊言

Description of VSM model

- query tf_vector
  用 <concept> 的關鍵字與 <question> 的 bigram (先經由jeiba斷詞再過濾 stopwords)當作term of vector，會有以下三種狀況：
  1. 若 element 長度 < 2，則捨棄
  2. 若 element 長度 == 2，則加入vector
  3. 若 element 長度 > 2，則進而將其拆做多個 bigram (ex: “流浪狗” -> “流浪”, “浪狗”)

  用 <title> + <question> + <narrative> + <concept>所有的內容來計算作 term 出現的次數即為 query tf_vector
  ex: element 包含 “流浪”, “浪狗”, “問題”，若內文中出現上述三個 bigram 次數分別為 (5,4,3)，則 query tf_vector = [5, 4, 3]
- doc tf_vector
  使用與 query 共同的 tf_vector terms，並計算各 term 出現在 doc 裡面的數量
- measure similarity
  使用 BM25 公式

Description of Rocchio relevance feedback

將相關文件經 BM25 score 做排序後，將前20個相關文件,與後20相關文件代入下方公式

$$\vec{q} = \alpha\vec{q} + \frac{\beta}{|D_r|}\sum \overrightarrow{d_j} - \frac{\gamma}{D_n}\sum \overrightarrow{d_j}$$

參數:

$$\alpha = 1.0,\ \beta = 1.0,\ \gamma = 0.5$$

Results of Experiments

	query term	tf	idf	similarity	normalize	score
1	<concept>	<concept>	log(N/k)	cosine	None	0.58134
2	<concept> + <title>	<concept> + <title>	log(N/k)	cosine	None	0.52484(變爛)
3	<concept> + <title>	<concept> + <title>	log(N/k)	cosine	TF(t,d) = 0.5 +0.5*f(t,d)/ MaxFreq(d)	0.52484(變爛)

experiment result:

- ➢ query term, tf 似乎只取 <concept> 的內容效果較好
- ➢ 若將 tf 依照上方 normalize 公式似乎沒有顯著的效果(用 Okapi~)

---------- Add BM25 ----------

parameters_1 : k1=1.2, k3=100, b = 0.75
parameters_2 : k1=2, k3=500, b = 0.75
全文 : <title> + <question> + <narrative> + <concept>

	query term	tf	BM25	other	score
4	<concept>	<concept>	parameters_1	None	0.76922(top7)
5	<concept>	<concept>	parameters_2	None	0.76709(變爛)
6	<concept>	全文	parameters_1	None	0.77112(top5)
7	<concept>	全文	parameters_1	算 tf vector 時 <title> <question> <narrative> 裡面若有出現 query term，將其出現次數 * 1.5	0.75274(變爛)
8	<concept>	全文	parameters_2	None	0.76085(變爛)

experiment result:

- ➢ 實驗4, 5、實驗6, 8，發現參數 k1=1.2(1~2), k3=100(0~1000) 左右有不錯的效果
- ➢ 實驗4, 6發現在算query term出現次數時(tf vector)用全文較只用 <concept>效果佳

---------- Add Rocchio feedback ----------

feedback_num : 相關與不相關的文件各取 [feedback_num] 個來優化 model

	epoch	feedback_num	BM25	other	score
9	2	5	parameters_1	None	0.73500
10	2	20	parameters_1	None	0.76351(變好)
11	10	20	parameters_1	with irrelevant docs	0.70830(變爛)

experiment result:

- ➢ Rocchio feedback 跑愈多次效果愈差，不用了！

---------- Modify query term, remove stopwords, use jieba 斷詞 ----------

	query term	tf	BM25	other	score
12	<concept> + <question>	全文	parameters_1	None	0.77418(top4)
13	全文	全文	parameters_1	取query term 時，先用 jieba 斷詞	0.74492(變爛)
14	<concept> + <question>	全文	parameters_1	取query term 時，<question> 先用 jieba 斷詞 + remove stopwords	0.78652(top2)
15	<concept> + <question> + <narrative>	全文	parameters_1	取query term 時，<question>, <narrative> 先用 jieba 斷詞 + remove stopwords	0.77520(變爛)
16	<concept> + <question>	全文	(part) parameters_1	將 parameters_1 的 b = 0.75 -> 0.65, other of 實驗14	0.78742(top1)
17	<concept> + <question>	全文	(part) parameters_1	將 parameters_1 的 b = 0.65 -> 0.55	0.78620(變爛)

experiment result:

- ➢ 實驗6, 12, 13, 15，發現 query term 用 <concept> + <question> 效果較佳
- ➢ 實驗14，取query term 時，若將 <question> remove stopwords，是從0.7 -> 0.8 的關鍵
- ➢ 實驗14, 16, 17，BM25的參數 b = 0.65 有較佳的效果，也讓我從top2 -> top1

Best performance (depends on the best ranking list on the public leaderboard)

- Query term extraction
  用 <concept> 的關鍵字與 <question> 的 bigram(先經由jeiba斷詞再過濾 stopwords)
- Calculate tf vector
  用 <title> + <question> + <narrative> + <concept>所有的內容來計算 term 出現的次數即為 query tf_vector
- Measure similarity
  Okapi/BM25 score

- Tips
  將 <title> 前兩字後兩字直接刪掉，不知為何效果變好^^

Discussion: what I learn in the homework

- 做 query 時 query term 要慎選，比如多考慮 <narrative> 的內容，雖然大多數的 term都會考慮到，但同時也會有太多雜訊(不重要的 term)，導致效果變差
- Rocchio feedback實質效果不好，或許是因為相關的文件 vector 都很分散，而導致此種情況發生
- 對於 tf vector 的 normalize 非常重要，將 measure similarity 換成 Okapi (裡面有 normalize 公式)，效能瞬間大幅提升

External tools

1. xmlparser
2. jieba
3. stopwords (手動取)