Report of Web Retrieval and Mining Programming Assignment 1

R08922A04 林承德

1.My VSM(Vector Space Model)

a. >>Query

我切query的方式，是先使用拿取query_test和query_train中每個query的<Question+Narrative+Concepts>，並使用Jeiba套件，將query切分成單詞，倘若其中出現某個term 長度>2，便將其切分成bigram。舉例：流浪狗，將切分成：流浪/浪狗。此外，倘若連續出現長度為1的term，我便將他組合成一個bigram，以免失去某些可能很重要的資訊。舉例：如果切出來的是'合','併'，我將而外新增一個term'合併'進入我的query。

b.>>TFIDF

在這邊我使用okapi/bm25的演算法來實作tfidf，其中，我最好的結果所使用的k值為1.25 b值為0.75。

我分別實作了三個function

query_tf：會計算出query自己的term frequency

doc_tf: 會計算出，每個document 對每個query 的term frequency

query_idf: 計算query 的每個term 在所有文章中的idf

c. >> Predict

最後以cosine similarity來計算他們之間的相似度，並排序，取出相似度最高的前100篇文章，作為我的predict。

2.My Rocchio Relevance Feedback

a >> Relevant documents

因為我們並沒有真實的答案，可以定義誰真的是相關的，因此我使用pseudo relevance feedback，先進行一次VSM，從我predict的結果中，取出前10篇，假設他們都相關，用此來進行rocchio 演算法。

b >> Query Expansion

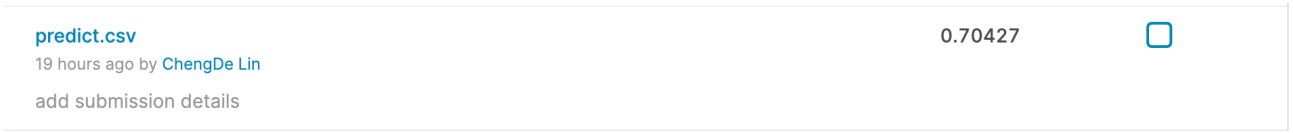
我將取出前10篇相關的document，並將他們所含有的文字進行斷詞，並找出在這10篇文章中，term frequency最高的20個term，並將他們加進我原本的query，然後再進行下一輪的VSM

3.Result of Experiments

a>> MAP value of different Parameters without Relevance Feedback

when k = 1.75, b = 0.75 acc = 0.73521

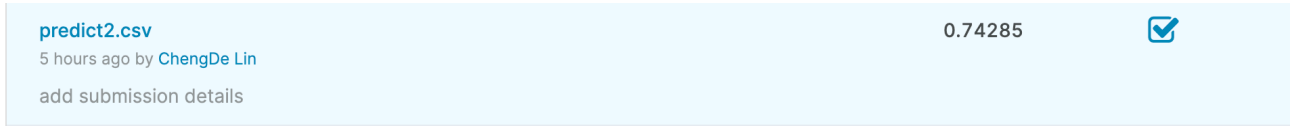
when k = 1.5, b = 0.75 acc = 0.70427



when k = 1.25, b = 0.75 acc = 0.74274

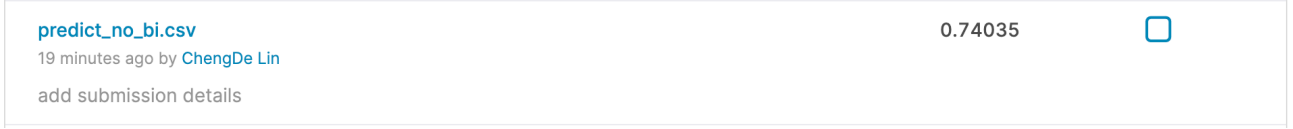
b>> MAP value with Relevance Feedback

when k = 1.25, b = 0.75, pseudo_relevant = top 10
acc = 0.74285



c>> MAP value of didn't merge two continue unigram to a bigram

when k = 1.25 b = 0.75 (without relevance feedback)
acc = 0.74035



4.Discussion

這是一份相當有趣的作業，能讓我們簡單時做一個retrieval model。

透過幾個簡單的實驗我們可以發現，不同的參數，可以對搜尋的結果造成不小的影響，此外，若以query的切法來看，倘若不將兩個連續的unigram合併成一個bigram，就可能失去部分的資訊，導致結果變差，與我們的假設相符。最後則是relevance feedback，他能夠幫助我們少量的提升搜尋的結果。