|  |
| --- |
| 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…。 4. 使用numpy做陣列操作。而math、operator做資料運算ex:math.log。 5. 使用sklearn.metrics.pairwise的cosine\_similarity計算餘弦定理。 |
| TFIDF參數設定-Documents |
|  |
|  |
| 1. TF計算：   計算該文章每個字出現的次數，並運用Log Normalization ( 1+log(tf) )概念調整TF參數，以完成該doc個文字的TF計算。   1. IDF計算：   self.docidf為文字出現的次數，計算方式為:設apple在a文章出現2次，b文章出現1次，則IDF[apple]為2而非3，即該字出現在幾篇文章中。   1. TFIDF計算：   為避免分母可能為0的現象，因此在IDF計算使用log(N+1/IDF+1)方法。 |
| TFIDF參數設定-Queries |
|  |
| 1. TF計算：   計算該query各文字出現的次數，並將「(上述計算結果\*0.2 /該query中出現最多次的文字次數) + 0.8」。0.2及0.8參數，是以Double normalization σ的概念去進行參數設定，經由實驗測出σ=0.8的準確較高。   1. TFIDF計算：Query TFIDF的計算方式與Document的計算概念相同。 |
| 心得 |
| 調整參數其實蠻耗費心力的，因為每次跑都需要花費一些時間，但在做第二份作業時，發現因為我在計算相似度時，一直重複計算tfidf，使時間複雜度太高，這時才了解在設計模型的時候，除提升準確率外，更要思考如何快速跑出結果。 |