# Retrieval and Mining
## hProgramming assignment 2 report

B04902103 Yun Da Tsai

5/24/2019

## 1 Model

### 1.1 TF-IDF

I used all the terms in *inverted-file* as my vocab list instead of using *vocab.all*. I calculated the TF*IDF vector for every documents directly using the term counts provided in *inverted-file*. I take the log value of IDF provided in *inverted-file*.

### 1.2 Query

To calculate the TF*IDF vector for queries, I use jieba to cut the query into several keywords same as the example source code.

### 1.3 Normalization

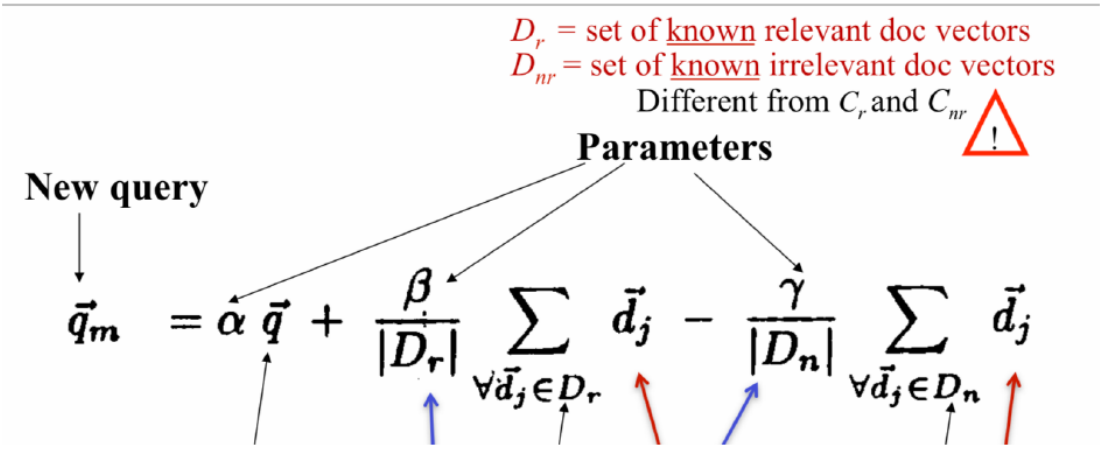
I implemented the normalization methods according to the slides including "raw TF", "Maximum frequency normalization" and "Okapi/BM25". I tried different normalization methods and achieved best results with Okapi/BM25. Then I tried different parameters $k$ and $b$ for Okapi/BM25 and the best parameters I found is $k = 2.5, b = 0.5$.

$$TF = \frac{(1 + k * freq)}{(k + freq)(1 - b + b * docLength/avdf)}$$

### 1.4 Similarity

I use *cosine* to calculate the similarity of two vectors.

## 2 Feedback




Since we do not know which query and document are truly related, I used pseudo feedback and take top 50 as related documents. The results are slightly lower than not using feedback.

## 3 Experiment Results and Analysis

### 3.1 Normalize vs. non-Normalize

| method | Raw | Maximum | Okapi |
|---|---|---|---|
| MAP | 0.136 | 0.157 | 0.204 |

Table 1: Normalize vs. non-Normalize

Table1. shows the performance of different normalization methods. From the result, we can see that Okapi/BM25 normalization has improved the most.

| parameters | k=2.5 b=0.3 | k=2 b=0.3 | k=5 b=0.3 | k=2.5 b=0.5 | k=2 b=0.5 | k=5 b=0.5 |
|---|---|---|---|---|---|---|
| MAP | 0.204 | 0.207 | 0.205 | 0.210 | 0.206 | 0.211 |

Table 2: Okapi/BM25 parameters

Table2. are experiments of different parameters. From the results we can find that larger k and b can have better result, which means the length of document have greater impact so we can penalize more on long documents when normalizing.

### 3.2 Feedback vs. non-Feedback

| feedback | non-feedback | $N$=50 $\beta$=0.75 $\alpha$=1 | $N$=100 $\beta$=0.7 $\alpha$=1 |
|---|---|---|---|
| MAP | 0.2109 | 0.1822 | 0.1725 |

Table 3: Feedback vs. non-Feedback

Table3. the feedback method makes slighty difference on the result in this case. This might because the weight of "key term" of query is large enough and the adjusted term weight has little impact on the whole, or because the extra terms and weights from the relevant document bring noise. $N$ is the number of documents used as relevant documents.