# Web Retrieval and Mining 2021 Spring Programming HW1

r09922104 余友竹

## Vector Space Model

My vector space model is a pipeline of several procedures, including query preprocessing, inverted-file preprocessing, documents' length extraction, and scoring mechanism. In the following subsections, I'll introduce these methods in detail.

### Query Preprocessing

To query and score each document much faster, I first extract term frequency and document frequency in inverted-file for each term.

I extract all query terms with unigram and bigram in "concepts", and transform all terms to their corresponding terms id. I further sorted these ids, and **only** processing these terms (skipping those who never appear in queries.)

### Documents' Length Extraction

In the above procedure, we skip the word who never appears in queries, the method reaches a better efficiency, however, we have no choice but to open every document and directly calculate its length.

### Scoring Mechanism

I use Okapi BM25 as my scoring/ranking function. The score for a document $D$ with a query $Q$ is defined as follow:

$$\text{score}(D,Q)=\sum_{t\in Q}\text{IDF}_t\cdot\frac{\text{tf}_{t,D}\cdot(k_1+1)}{\text{tf}_{t,D}+k_1\cdot(1-b+b\cdot\frac{|D|}{\text{avgdl}})}$$

and $\text{IDF}_t$ is defined as:

$$\text{IDF}_t=\ln\left(\frac{N-n_t+0.5}{n_t+0.5}+1\right)$$

where $\text{tf}_{t,D}$ is $t$'s term frequency, $|D|$ is the length of document $D$, avgdl is the average document length in the collection, $N$ is the number of documents in the collection, $n_t$ is the document frequency of $t$, and $k_1$, $b$ are free parameters.

## Rocchio Relevence Feedback

In my feedback model, there are 3 free parameters:

- `num_iter`: the numbers of iteration to feedback our query.
- `num_related`: the numbers of documents with the highest scores to be labeld as **relevent documents**.
- `alpha`: the parameters to determine the degree of mixing between the original query vector and the centroid of the relevent documents' term vector.

In the begining, the default query vector is $\mathbf{1}$. I tune the query weights by the following equation iteratively.

$$\mathbf{q}_{r+1} = \alpha \cdot \mathbf{q}_r + (1-\alpha) \cdot \frac{1}{|D_r|} \sum_{\mathbf{t} \in D_r} \mathbf{t}$$

where $D_r$ collects all relevent documents' term vectors.

## Experiments

I use MAP@100 to evaluate the performances. The performances in training set, public testing set, and in private testing set would be reported seperately in the following subsections.

### Different Parameters of VSM

The only free parameters in my VSM are $k_1$, $b$, which are the parameters in the scoring function metioned above.

| $k_1$ | $b$ | TRAINING | PUBLIC TESTING | PRIVATE TESTING |
|---|---|---|---|---|
| 1.2 | 0.50 | 0.77347 | 0.78540 | 0.75565 |
| 1.2 | 0.75 | 0.78055 | 0.78405 | 0.75697 |
| 1.2 | 1.00 | 0.78523 | 0.77068 | 0.76165 |
| 1.6 | 0.50 | 0.77186 | **0.78581** | 0.75986 |
| 1.6 | 0.75 | 0.78548 | 0.78173 | 0.76160 |
| 1.6 | 1.00 | 0.78304 | 0.76480 | **0.76632** |
| 2.0 | 0.50 | 0.77181 | 0.78507 | 0.75995 |
| 2.0 | 0.75 | **0.78572** | 0.77287 | 0.76349 |
| 2.0 | 1.0 | 0.78304 | 0.76093 | 0.76449 |

### With/Without Relevence Feedback

Comparing with/without relevence feedback with the following parameters.

- $k_1 = 1.2, b = 0.75$
- num_iter = 10, num_related = 5, alpha = 0.98

| | TRAINING | PUBLIC TESTING | PRIVATE TESTING |
|---|---|---|---|
| w. fb | **0.81022** | **0.78647** | **0.77584** |
| w/o. fb | 0.78055 | 0.78405 | 0.75697 |

### Another Strategy

I also try to use different terms for searching, and further mixing their scores with a parameter $\beta$, the scoring function is:

$$\text{score} = \beta \cdot \text{score}_c + (1 - \beta) \cdot \text{score}_t$$

Where $\text{score}_c$ is the result by using concepts for searching, and $\text{score}_t$ is the result by using titles for searching.

The experiments are recorded as follow.

| | TRAINING | PUBLIC TESTING | PRIVATE TESTING |
|---|---|---|---|
| w. Fb | **0.81022** | **0.78647** | **0.77584** |
| $\beta = 0.5$ w. fb | 0.80764 | 0.78251 | 0.76543 |

### Other Parameters by Using Relevence Feedback

Fixing $k_1 = 1.2, b = 0.75$.

| num_iter | num_related | alpha | TRAINING | PUBLIC TESTING | PRIVATE TESTING |
|---|---|---|---|---|---|
| 10 | 5 | 0.98 | 0.81022 | **0.78647** | **0.77584** |
| 10 | 5 | 0.95 | 0.80835 | 0.77748 | 0.74349 |
| 20 | 10 | 0.8 | **0.82511** | 0.74657 | 0.68307 |

## Discussions

Different parameters in BM25 don't seriously affect the overall performance. Generally the performances are about $0.77 - 0.78$ in training, and public testing set, and $0.75 - 0.76$ in private testing set.

The performance soars up in training set by using relevence feedback. Besides, it seems to be much more robust with appropriate parameters. However, we have to carefully tune the query vector. As when alpha = 0.8, the performance dropped signifincantly in private testing set.

Trying to use another space for querying seems to be a wrong decision. After several experiments, I think the best query space should be the words in **concepts**.

The efficiency is also a big issue. I speed up my retrieval model by several ways, including:

- Using numpy, pandas functions to read files.
- Trying to avoid **for loop** as much as possible.

- Using sparse matrix to store term-document relationship
- calculate BM25 scores by using the matrix multiplication.

Overall, my retrievel system reaches a significant efficiency, which the whole procedure could be done in less than **20 sec.** when all CPUs in the linux3 server are idle. Besides, it keeps an acceptable performance. In final submissions, I get **11th** in public testing set, and **10th** in private testing set.