Retrieval and Mining
hw1 report

B04902103 Yun Da Tsai

April 2019

1 Vector Space 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*. The TF-IDF vector would have 1193467 dimensions.

1.2 Query

To calculate the TF*IDF vector for queries, I considered all the text in the query file including the "title", "question", "narrative" and "concepts". I filter out some irrelevant words from the queries. I give different weights to different query tags and different terms. The weights are by the following formulas:

$$Score = bigram * 1.5 + unigram$$

$$Score = title * 5 + question + narrative + concepts * 3$$

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 $k = 2.5, b = 0.5$.

1.4 Similarity

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

1.5 Implement

I use sparse matrix from the *scipy* package to store vectors and do the matrix operations.

2 Rocchio Relevance Feedback

I choose the top 30 documents in the *ans-train.csv* as relevant document and set $\beta = 0.8$.

3 Experiment Results and Analysis

3.1 Normalize vs. non-Normalize

method	Raw	Maximum	Okapi
MAP	0.745	0.764	0.814

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. The experiments are done in local with *ans-train.csv* and *query-train.csv*

parameters	k=1.4 b=0.7	k=1.5 b=0.7	k=1.5 b=0.5	k=1.9 b=0.5	k=2.3 b=0.5	k=2.5 b=0.5
MAP	0.796	0.797	0.736	0.802	0.793	0.814

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	β=0.5	β=0.7	β=0.8	β=0.9
MAP	0.8144	0.8144	0.8145	0.8144	0.8143

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.

4 Discussion

In this homework, I have learned the vector space model with TF-IDF representation, different normalizing methods and feedback methods. These concepts reminds me of some new deep learning methods which I have learned in other courses such as RNN encoder, doc2vec and Paragraph vector.