11-791: Design and Engineering of Intelligent Information System

Homework Assignment #3

Yu-Hsin Kuo
Andrew ID: yuhsink

Report

Error Analysis

The baseline MRR score is 0.4375. I list the first five queries with the score and the relevance. Here, I only list the rank that is higher than the relevant sentence.

cosine	rank	rel
qid = 1		
0.6396	1	0
0.2791	2	1

cosine	rank	rel
qid = 2		
0.3010	1	0
0.2858	2	1

cosine	rank	rel
qid = 3		
0.4216	1	0
0.2673	2	0
0.2357	3	1

cosine	rank	rel
qid = 4		
0.2739	1	0
0.2315	2	1

cosine	rank	rel
qid = 5		
0.4082	1	0
0.1633	2	0
0.0000	3	1

The original questions are:
(1) Give us the name of the volcano that destroyed the ancient city of Pompeii
(2) What has been the largest crowd to ever come see Michael Jordan
(3) In which year did a purchase of Alaska happen?
(4) What year did Wilt Chamberlain score 100 points?
(5) What river is called China's Sorrow?

If you compare the result with the questions you can classify some of the errors into the following types:
(1) **Capital and non-capital**. For example, China's Sorrow should be equal to sorrow of

China. Without doing normalization, this "Sorrow" is not the same as this "sorrow"
(2) **Punctuation**. For example, we have "Pompeii;", "Michael Jordan–", ""sorrow", "China"", etc., and these are not the same as original words when we calculate cosine similarity.
(3) **Stop words removal**. Words such as "of", "a", "the", etc. should also be removed
(4) **Active/passive and plural/singular**. For example, China's sorrow should be equal sorrow of China or did a purchase of Alaska should be equal to Alaska was purchased. In addition, like and likes or die and dies should also be treated as the same word. To handle this, using proper stemming would be helpful.
(5) **Semantic and synonym**. Some of the questions are asking where, when, etc., and we didn't consider if the sentence contains the corresponding answers like date and places. Or different words convey the same meanings which the system cannot capture like spaceship and spacecraft.

By manually classifying errors of all the 20 queries given the types above, we can have the following table: (Notice that I don't include "stop word error" as an error type when making the table, because all of them contain stop words and have to do removal anyway.)

Error Type	Percentage
Capital and non-capital	5 %
Punctuation	40 %
Active/passive and plural/singular	25%
Semantic (Synonym)	44%

Describe your systems design

It will first create a term vector for each document using the default tokenizer. I also implemented a tokenizer that does stop words removal, case normalization and stemming. To store the whole term frequency vector, the data structure I used is hashmap which uses a string as key and term frequency as value. Then for each query we will calculate similarity of each document. Since we have to sort the score while still know the corresponding entry, I created a Pair class which stores the score and the index. To use TF-IDF you have to use the function "transformTfIdf" to update your term frequency vector into Tf-Idf vector before calculating the cosine similarity. For BM25, you can just called the "bm25" function.

BONUS Part

From the error analysis above, I propose some initial processing: (a) stop words removal, (b) normalization (transform into lowercase) and (c) remove punctuation to handle the problem (1), (2) and (3).

After doing the proposed processing, I got MRR = 0.6208. While the rank of most relevant queries improves, some of them didn't but decreases.
To handle the problem (4), I used StanfordLemmatizer to do stemming and I got MRR = 0.6792.

Later on, I used TF-IDF approach and got MRR = 0.6958 and BM25 with b = 0.75 and 0.5 and got MMR = 0.8125 and 0.8208, respectively.
The result is as follows:

Approach	MMR
(1) stop word removal + (2) normalization + (3) punctuation removal	0.6208
(1) + (2) + (3) + TF-IDF	0.6958
(1) + (2) + (3) + BM25 (b= 0.75)	0.8125
(1) + (2) + (3) + BM25 (b= 0.5)	0.8208