Vincent Chee
Natural Language Processing
Professor Kristina Striegnitz
04/05/15

Problem Set 5
Questions 2 and 3

In order to improve my tf-idf based document retrieval system, I will modify the program to exclude stopwords in one version of the program and use a stemmer to reduce the words to their stems in the other version.

Evaluation

Descriptions of each improvement

Stopword Exclusion: The first improvement made would exclude stopwords by removing them from the inverted index as well as the query. This means that the program takes out high-frequency words such as *the*, *to* and *although*.

Use of *Stemmer*: The second improvement made was using a stemmer to reduce the words to their stems. This means that words such as 'lying' will be lemmatized (make sure resulting form is a known word in a dictionary) to something like 'lie'.

Prediction of Outcome

Stopword Exclusion: By excluding stopwords, it should increase the numeric value of the term frequency since the term frequency is given by the # of times a keyword occurs in a doc / # of words in a doc. By removing stopwords, we decrease the # of words in a doc therefore reducing the number we are dividing by and increasing the term frequency. This should give us a better indication of the relevant documents which we are searching through. This should also increase the efficiency of the search because if we are removing the stopwords from the inverted index thus we are counting fewer words for the normalized term frequency.

Use of Stemmer: By using a stemmer, it should once again increase the numeric value of the term frequency. However, this time it will increase the # of times a keyword occurs in a doc. This is because if we are searching for the word 'pie', and we have the word 'pies', the stemmer will change all occurrences of the word to 'pie', thus increasing the count the # of times a keyword occurs in a doc. This improvement may help us indicate whether some documents are more relevant or relevant at all if it contains an

instance of the word but has an affix. I think this may result in less efficiency because it may find the query word in more documents once the affix is removed.

Evaluation Results

Running Times and Efficiency of Each Program			
Queries	tfidf – original	tfidf – first	tfidf – second
“feminist movement”	15.543	13.148	N/a
“chinese economy”	16.258	15.731	N/a
“bill clinton”	5.881	5.769	N/a
“god”	3.709	4.440	N/a
“theory of relativity”	44.552	10.608	N/a
“binary code”	5.532	5.239	N/a
“photosynthetic protists”	0.375	0.311	N/a
“gettysburg address”	4.062	4.426	N/a
“hippocratic oath”	1.276	1.299	N/a
“pythagorean theorem”	0.940	0.777	N/a
Average Runtime	9.8128	6.1748	N/a

Average Precision of Each Program			
Queries	tfidf – original	tfidf – first	tfidf – second
“feminist movement”	2/10	2/10	N/a
“chinese economy”	4/10	4/10	N/a
“bill clinton”	4/10	4/10	N/a
“god”	10/10	10/10	N/a
“theory of relativity”	2/10	2/10	N/a
“binary code”	5/10	5/10	N/a
“photosynthetic protists”	3/10	3/10	N/a
“gettysburg address”	2/10	2/10	N/a
“hippocratic oath”	3/10	3/10	N/a
“pythagorean theorem”	4/10	4/10	N/a
Average Precision	3.9	3.9	N/a

Analysis

In order to determine whether the improvement attempts were successful we need to look at several statistics which we obtained from running the code. We can analyze the

efficiency of the different programs by looking at the runtime, we can also analyze the average precision for the different programs. It was clear that the program with removed stopwords ran slightly faster, however it is hardly significant, while the average precision for both programs was exactly the same.

Conclusion

These results did not confirm my predictions, in fact, since I was only able to compile data for the original program and the program with the excluded stopwords, I was unable to get a full picture of what truly occurred. However, based on the results I was able to obtain for the first two programs, the results seemed to be exactly the same. My explanation for this is that although we removed the stopwords from the query as well as the inverted index, there were not enough stopwords in the index to provide a significant enough of a change to be shown in the results. But, this does not explain why I obtained identical results for both the original program and the program with the removed stopwords. The only difference between the two programs was the average runtime, the removed stopwords program ran 3 seconds faster on average. Ultimately, it is very possible that there is something unaccounted for in the removed stopwords program and thus it did not produce desirable results.

List of Test Queries Used

- “feminist movement”
- “chinese economy”
- “bill clinton”
- “god”
- “theory of relativity”
- “binary code”
- “photosynthetic protists”
- “gettysburg address”
- “hippocratic oath”
- “pythagorean theorem”