IR Project 3

- *Sudhamayi Nanduri (nanduri.s@husky.neu.edu)*

Steps Involved in Building the Project

1. The CACM document collection was taken.
2. Each document is processed to extract the required text, which are mostly Titles or Abstracts and their Author Names.
3. The punctuation marks are removed and the letters are converted to lowercase.
4. Stopping technique is applied.
5. Stemming technique is applied.
6. Each word from the document is processed to build the inverted index as follows.
7. Totally, 3 files are maintained, “terms.txt”, “mappings.txt”, and “documents.txt”
8. In the “terms.txt”, each term encountered in the CACM documents are stored along with their term id, ctf (corpus frequency statistics), df (number of documents they occur in) and the offset value of their mapping to “mappings.txt” file and the length to be read from “mappings.txt”
9. In the “documents.txt”, we have all the documents details, their document Id, document Name and the document Length information stored.
10. In the “mappings.txt”, we store the mapping between the terms in “terms.txt” and documents in “documents.txt”. It stores the term id, document id’s of those documents in which the term is occurring and the frequency of its occurrence in those respective documents.
11. 64 raw queries are taken to give input to this inverted index.
12. These queries are again processed, by removing the punctuation marks, by apply stopping and stemming technique and are store into an input text file.
13. The following words having high frequency in the queries do not help in retrieving the documents. These words have been omitted for better query processing.
    a. Interested
    b. Dealing
    c. Especially
    d. Rather
    e. Dealing
14. Now, this input text file containing the filtered queries is run against this inverted index to retrieve the related documents.
15. Now, similar to the previous project, all the 5 models have been applied to find the mean precision values.
16. The following are the observations noted for the mean average precision values:

Models	**Mean Average Precision** **Of CACM qrel file**
Vector Space Model 1	0.2732
Vector Space Model 2	0.2885
Language Modeling with Laplace Smoothing	0.2292
Language Modeling with Jelinek-Mercer Smoothing	0.2355
BM-25	0.2940

The mean precision values for at 10 and 30 documents are as follows.

	CACM qrel File	
	Precision at 10 Documents	**Precision at 30 Documents**
Vector Space Model 1	0.2915	0.2531
Vector Space Model 2	0.3531	0.2653
Language Modeling with Laplace Smoothing	0.2394	0.2016
Language Modeling with Jelinek- Mercer Smoothing	0.2415	0.2238
BM-25	0.3492	0.2691

The following observations are made from the above tables:

1. We can observe that **BM-25** is the best model in comparison to other models. It has the highest mean precision value among all the models.
2. We can find that the next highest value is **Vector Space Model 2**.
3. **Vector Space Model 1** using Okapi-tf has lower precision than **Vector Space Model 2** with Okapi-tf using **idf** (inverse document frequency). **Idf** reflects the term's occurrences in documents and whether the term is discriminating to find relevant documents.
4. We can also observe that, as the number of retrieved documents increase, the precision value decreases.
5. By using the stopping technique (that is, the stop list provided), we are able to get a higher precision value by blocking those terms which are not discriminating to find relevant documents.

In comparison to the previous project 2, the values obtained in this case are higher than the previous values. The following are some of the reasons for this.

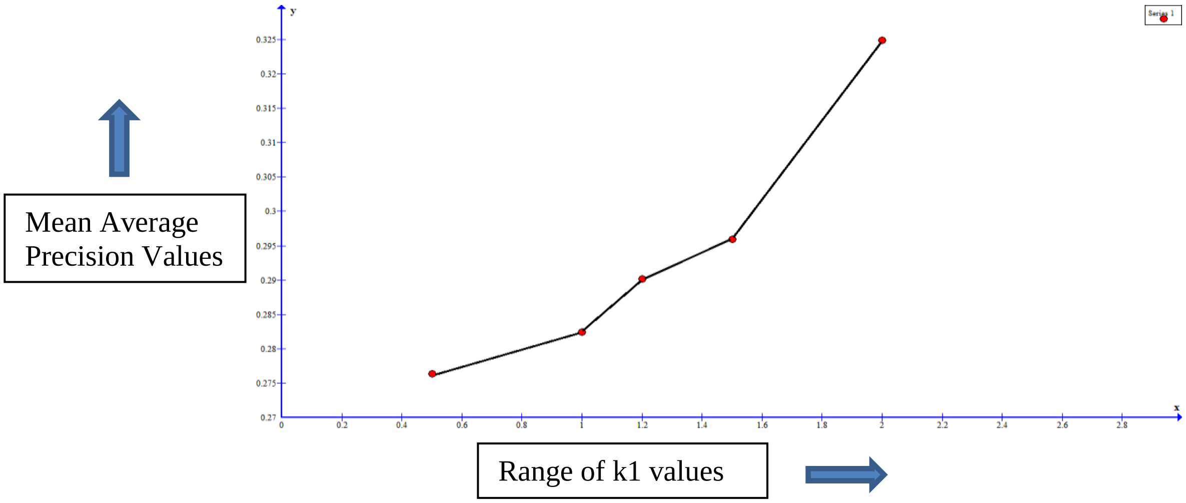
1. Project 2 uses a case-sensitive database, where the retrieved documents are less matches. In Project 3, we are converting every word to lowercase while building the index and during query processing.
2. Only few databases used in Project 2, performed both stopping and stemming. We are implementing those techniques while building the inverted index.
3. We processed the document content while building the inverted index and the raw queries during query processing to remove unnecessary punctuation marks. Hence, it gets more related documents.

Other Observations:

1. Consider the best model **BM-25**:
   a. By, changing the value of **k1** from 0.5 to 2, we get the following observations:

Range of k1 values	**Mean Average Precision** **For BM-25 Model**
0.5	0.2764
1.0	0.2824
1.2	0.2902
1.5	0.2959
2.0	0.3249

The graph has been plotted with k1 on x-axis and the mean average precision values on y-axis.

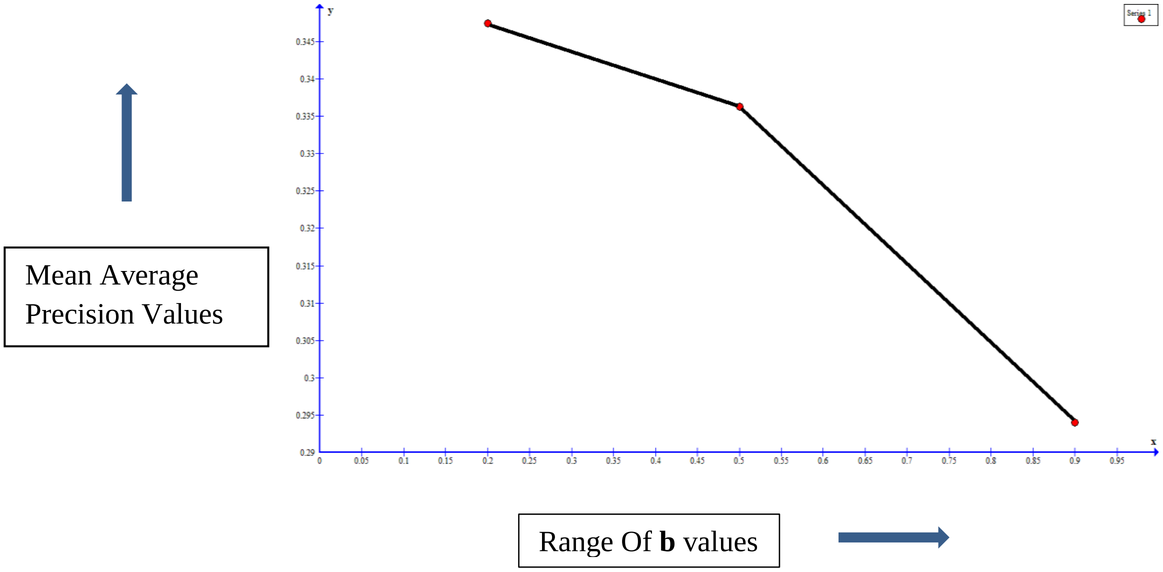


We can observe that as the value of **k1** is increasing as the Mean Avg Precision Value is also increasing.

b. The following are the observations when the **b** parameter is varied between 0.2 to 1.0

Range of b values	**Mean Average Precision** **For BM-25 Model**
0.2	0.3475
0.5	0.3363
0.9	0.2940

The following is the graph for the above observations:



We can also observe that by increasing the value of parameter **b**, the Mean Avg Precision Value decreases.

2. Consider the following 2 queries from the list of input.
   a. **The role of information retrieval in knowledge based systems (i.e., expert systems). [Query Number 49]**

      The Average Precision values obtained for giving this query as input is 0.1880 using BM-25 model.

      Now, by replacing the word “retrieval” by “extraction” in this query, and the word “complexity”, the mean avg precision value is 0.1800

      There is a decrease in the value of mean avg precision implying that the related documents retrieved are less in the second case.

   b. **Results relating parallel complexity theory (both for PRAM's and uniform circuits) [Query Number 62]**

      The Average Mean Precision Value obtained for this query is 0.1800 using BM-25 model.

      The word “complexity” in the query has been replaced by “difficulty” and is given as input. The Mean Average Precision obtained is still 0.1800

      Hence, we can find that there is no change in the mean average precision values.

      When this experiment was conducted in Project 2 on other different queries, we found that the average mean precision values were increased. But, in this case we are recording values with very less difference. Hence, we can say that the stemming technique (Porter Stemmer) implemented in our project is more efficient than the stemming technique used in the databases of Project 2.

3. Now, the queries have been parsed a bit more by excluding the numbers from them.

   The following mean precision values have been recorded.

Models	**Mean Average Precision** **Of CACM qrel file**
Vector Space Model 1	0.2868
Vector Space Model 2	0.2892
Language Modeling with Laplace Smoothing	0.2577
Language Modeling with Jelinek-Mercer Smoothing	0.2598
BM-25	0.2963

   We can observe that the mean average precision values have increased compared to the previous mean average precision values. Hence, the documents retrieved in this case are more relevant.

4. Consider the following 2 queries.

   a. Parallel processors in information retrieval [Query Number 50]
   b. Parallel processors and paging algorithms [Query Number 51]

   The mean average precision for these both queries for **BM-25 Model** is 0.1831

   Now let us remove the words, **"parallel"** and **"processors"** from the above queries. The mean average precision for these queries is 0.1800

   Hence, we can understand that these words are important in the retrieval of the related documents.

   This can be proved from the number of documents associated with these both words from "mappings.txt"

   CACM-0727 CACM-2346 CACM-2106 CACM-2707 are the documents retrieved for the words both "**parallel"** and "**processor"**.

The documents retrieved for the words **“information”, “retrieval”, “paging”, “algorithms”** are CACM-0272 CACM-0598 CACM-0630 CACM-0950 CACM-1010 CACM-1047 CACM-1171 CACM-1179 CACM-1426 CACM-1491 CACM-1591 CACM-1680 CACM-1693 CACM-1718 CACM-1743 CACM-1885 CACM-1907 CACM-1938 CACM-1965 CACM-1974 CACM-2018 CACM-2035 CACM-2144 CACM-2194 CACM-2198 CACM-2208 CACM-2251 CACM-2277 CACM-2543 CACM-2580 CACM-2584 CACM-2727 CACM-2770 CACM-2948 CACM-3076 CACM-3185

Hence, the documents for these words are more (36 documents). Hence, these words are not that useful to get the relevant documents to the query as compared to “parallel” and “processors” which retrieve the specific required documents (4 documents).

5. The variations of smoothing parameter “λ” between 0.2 to 1.2:

Range of λ values	**Mean Average Precision** **For Jelinek-Mercer Model**
0.2	0.2598
0.5	0.2588
0.8	0.2597
1.9	0.2598

Hence, we can observe that the mean average precision is almost the same for the range of “λ” values.

6. While building the inverted index, we used the stopping technique for words which occur very frequently and which are not helpful in determining the relevant documents. Let us now build the inverted index without using the stopping technique.

   The total number of terms in the inverted index (without using stopping technique) are 16378. But, the total number of terms in the inverted index (using stopping technique) are 7979. Hence, it occupies more memory if we do not use stopping technique.

   The time taken to build the index (without stopping technique) is 5.5 seconds as compared to the index with stop words was 6.3 seconds.

   The mean average precision values recorded the queries are:

Models	**Mean Average Precision** **Of CACM qrel file**
Vector Space Model 1	0.2867
Vector Space Model 2	0.2895
Language Modeling with Laplace Smoothing	0.2578
Language Modeling with Jelinek-Mercer Smoothing	0.2599
BM-25	0.2962

   We can observe that there is difference from the original values only by 0.0001 (which is negligible) because we are still using the stopping technique to the queries.

   Let us see the mean average precision values when stopping is not applied to the queries.

7. Without applying the stopping technique to the queries, the following mean average precision values are recorded:

Models	Mean Average Precision Of CACM qrel file
Vector Space Model 1	0.2350
Vector Space Model 2	0.2404
Language Modeling with Laplace Smoothing	0.2301
Language Modeling with Jelinek-Mercer Smoothing	0.2325
BM-25	0.2421

Hence, we can observe that the mean average precision values have decreased when we do not use the stopping technique. Hence, we can conclude that we are retrieving less relevant documents.

The time taken to process the queries and calculate the mean average precision values for the above models (using stopping technique) takes 24.2 seconds. But, the time taken to process the queries and calculate the mean average precision values (without using the stopping technique) for the above models is 60 seconds. Hence, the search and retrieval time is more if we do not use stopping technique.

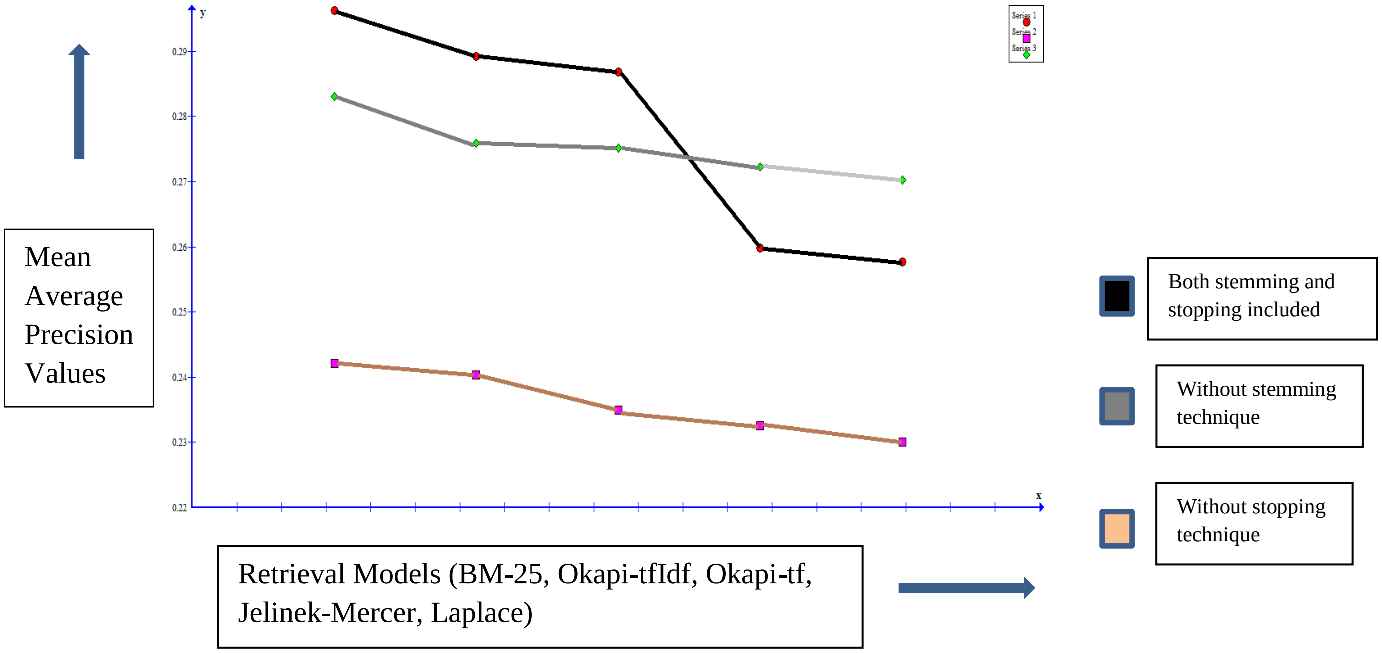
Hence, the advantages of using stopping techniques such as retrieval of related documents and faster search process for the queries is proven from the above observation. The memory occupied is also less when stopping technique is used. The only tradeoff is the time taken to build the index.

8. By not apply the stemming technique to the inverted index and the queries, the following mean average precision values are recorded.

Models	Mean Average Precision Of CACM qrel file
Vector Space Model 1	0.2751
Vector Space Model 2	0.2759
Language Modeling with Laplace Smoothing	0.2702
Language Modeling with Jelinek-Mercer Smoothing	0.2722
BM-25	0.2831

Hence, by not using the stemming technique the values obtained are less than the original values. Hence, the documents retrieved are less in this case as the particular words in query are searched without stemming.

9. Below is the graph which shows the comparison of Mean Average Precision Values when stemming is not applied, stopping is not applied and both stemming and stopping are applied to the inverted index and the queries.



The above graph is plotted in the order,
BM-25 > Okapi-tfIdf > Okapitf > Jelinek-Mercer Smoothing > Laplace Smoothing

Hence, from the above graph we can conclude that both stemming and stopping increase the mean precision values. Hence, the documents retrieved are more relevant.