**CS – 6200: Information Retrieval**

**Fall 2017**

**Professor Nada Naji**

**Team Members:**

**Akshay Singh**

**Sanjay Murali**

**Chaitanya Kaul**

Introduction

The goal of this project is to design and build our own information retrieval systems, evaluate and compare their performance levels in terms of retrieval effectiveness.

The project uses four retrieval models –

1. BM25 model
2. tf-idf model
3. Smoothed Query Likelihood Model (⅄ = 0.35) model
4. Lucene

Out of the four baseline runs mentioned above we choose BM25 model for performing Pseudo Relevance Feedback (PRF).

Additionally, project also performs the following runs on modified corpus:

1. Stopping technique (with no stemming) using the given “common\_words.txt” file was performed on the corpus. Out of the four baseline runs mentioned above , three baseline line runs namely , BM25 , tf-idf and Smoothed Query Likelihood Model were run on the corpus.
2. The given stemmed version of the corpus in the file “cacm\_stem.txt” is indexed and results are retrieved for the stemmed queries given in “cacm\_stem.query.txt” after running, baseline runs: BM25 , tf-idf and Smoothed Query Likelihood Model.

The Luhn’s algorithm is used as the snippet generation technique and performed on BM25 base line run.

The performance assessment of the retrieval models in terms of their effectiveness for all the eight distinct runs is performed using below measures.

1. MAP
2. MRR
3. P@K, K = 5 and 20
4. Precision and Recall

Individual Contribution

1. Akshay Singh :

He was responsible for generating new corpus after removing the stopped words given in the “common\_words.txt” file and Also generated Stemmed corpus from the given document “cacm\_stem.txt”. He also performed 6 runs required for the completion of task 3 , where he ran BM25 model , Smoothed Query Likelihood Model and tf – idf model on the “Stopped” version of the corpus and “Stemmed” Version of the corpus (with stemmed queries given in “cacm\_stem.query.txt” file).

He converted the results for all the 8 runs in the project from the text files to Excel sheets. He also made notable contribution in the Introduction , Literature , Resources , Implementation and Results section of the Project report.

Literature

A brief overview of the techniques and algorithm used for the implementation of different tasks of this project are given below:

1. **BM25 Model:**

The BM25 model is implemented as given in the recommended text book “Search Engines”. The file named “cacm.rel.txt” is used for determining the relevance information. The values of parameters k1 = 1.2, b= 0.75 and k2 = 100 are set for maximum effectiveness as found in TREC experiments.

1. **tf-idf Model:**

The term frequency for the document is multiplied with the inverse document frequency to calculate tf-idf for each document.

1. **Smoothed Query Likelihood Model (⅄ = 0.35):**

Jelinker – Mercer method is used for implementing Smoothed Query Likelihood model , as per TREC evaluation ⅄ = 0.1 is optimal for short queries and ⅄ = 0.7 is optimal for longer queries. As given in the document we have implemented the model with ⅄ = 0.35.

1. **Pseudo Relevance Feedback (PRF):**

The Pseudo relevance feedback has been performed using Rocchio Algorithm, on the baseline run for BM25 model.

1. **Stopping:**

The shared stop list in the text file “common\_words.txt” was used to create a new corpus after removing all the stop words in the file and the new corpus was then indexed. The three baseline runs performed on Stopping technique were BM25 , tf-idf and smoothed query likelihood.

1. **Stemming:**

A new corpus was generated from the shared stemmed file “cacm\_stem.txt” and stemmed query queries were used from the shared file “cacm\_stem.query.txt”.

Three baseline runs that used stemming were BM25 , tf – idf and smoothed query likelihood.

1. **Snippet Generation:**

The algorithm used in the project for generating snippets is called Luhn’s algorithm. This algorithm ranks each sentence in a document using a significance factor and use top sentences for the summary.

1. **MAP:**

The MAP or Mean Average Precision can be calculated using the below formula.

Mean Average Precision = Sum (All the Average Precision) / Total Queries

1. **MRR:**

MRR or Mean Reciprocal Rank is the average of reciprocal ranks for all the given queries.

1. **P@K, K = 5 and 20:**

It is the number of total relevant documents received per K top retrieved documents. In the project it has been calculated for K = 5 and K = 20.

1. **Precision and Recall:**

Precision is the number of relevant documents received per total retrieved documents.

Recall is the number of relevant documents received till a particular iteration per total number of relevant documents.

Resources

1. **Beautiful Soup:**

BS4 has been used to parse and process the documents given in the zip folder “cacm.tar.gz” and queries given in the file “cacm.query.txt”

1. **Lucene:** The following Lucene libraries were used:

a. lucene-core-VERSION.jar

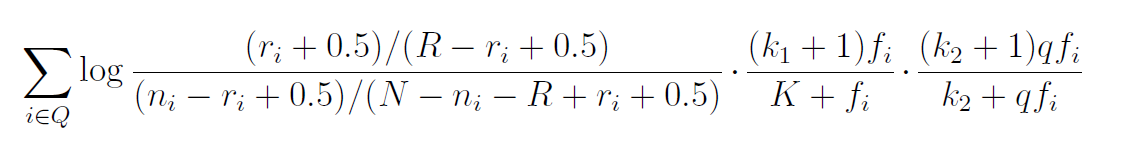
b. lucene-queryparser-VERSION.jar

c. lucene-analyzers-common-VERSION.jar.

Implementation

1. **BM25 Model:**

In the project the following formula was implemented to calculate the scores of the documents, which were then sorted in decreasing order to calculate the ranking of the documents.



Q = Query Term Set (summation is done for each term inside the query)

ri = The number of relevant documents containing term i.

ni  =The number of documents containing term i

N = The total number of documents in the collection

R = The number of relevant documents for this query.

fi = The frequency of the term i in documents

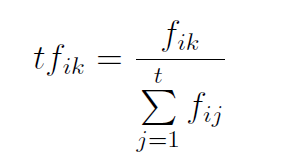
qfi = The frequency of term i in the query

k1 , k2 and K are the parameters whose values are determined empirically.

1. **Tf – idf Model:**

In the tf – idf model ,

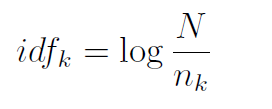
the tf or the “Term Frequency” can be calculated as :



tfik = Term frequency weight of the term k in the document Di.

fik = The number of occurrences of term k in the document.

the idf or the “Inverse document frequency” can be calculated as :



Here

idfk = The inverse document frequency weight for term k

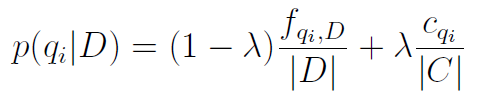
N = The number of documents in the collection

nk = The number of document in which the k occurs.

In tf-idf these weights are multiplied.

1. **Smoothed Query Likelihood Model (⅄ = 0.35):**

The Smoothed Query Likelihood Model is implemented using the Jelinek – Mercer Method given below :



Here ,

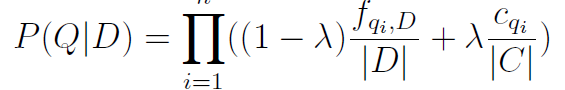
fqi,D = The number of times word qi occurs in the document D.

|D| = The number of words in D.

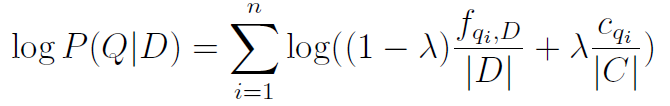
cqi= The number of times a query word occurs in the collection of documents

|C| = The total number of words occurrences in the collection.

Substituting this in the query likelihood model gives –

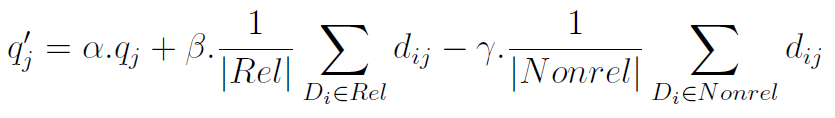


Taking log to preserve accuracy , we get –



1. **Pseudo Relevance Feedback (PRF):**

Pseudo Relevance feedback was implemented using Rocchio Algorithm given below.



Here ,

qj = The initial weight of query term j

Rel = The set of identified relevant documents,

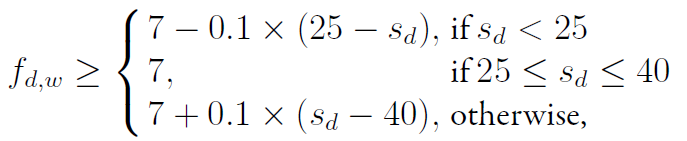
Nonrel = The set of non relevant documents

dij = The weight of the jth term in document i

α , β , ⅄ are the paremeters.

1. **Snippet Generation Technique:**

The Snippet generation technique is implemented using the Luhn’s Algorithm given below.



Here ,

fd,w = The frequency of word w in document d.

sd = Number of sentences in document d.

w = Significant word.

Discussion – Query by Query Analysis

For the task 3 , where stemmed queries are used for running baseline runs for models BM25 , Smoothed Query Likelihood and tf-idf on the given stemmed corpus.

We consider below three queries for analysis:

1. STEMMED : portabl oper system , ORIGINAL : portable operating systems

2. STEMMED : parallel algorithm , ORIGINAL : parallel algorithms

3. STEMMED : perform evalu and model of comput system , ORIGINAL : performance evaluation and modelling of computer systems

For the first query , it is observed that all the terms in original query have been transformed in the stemmed version of the query i.e.,

‘portable’ 🡺 ‘portabl’ , ‘operating’ 🡺 ‘oper’ and ‘systems’ 🡺 ‘system’.

In case of the second query , there is only one difference between original and stemmed version i.e. , ‘algorithms’ 🡺 ‘algorithm’

For the third query , there are 5 differences between the original and stemmed version i.e.,

‘performance’ 🡺 ‘perform’ , ‘evaluation’ 🡺 ‘evalu’ , ‘modelling’ 🡺 ‘model’ , ‘computer’ 🡺 ‘comput’ and ‘systems’ 🡺 ‘system’

For the stemmed run , the stemmed corpus is generated , which includes same stem for all the words which belong to same stem class. For instance , “portable , portably , portables” will be replaced by word “portabl” in the stemmed corpus. Similarly , “opera , operabilities , operability, operable, operation , operating etc.” will be replaced by word “oper” in the stemmed corpus.

Thus the documents having more words of the same stem class , will be more relevant to stem word present in the stemmed query.

For the original run , the relevance of the document will be measured by the original word in the query and the same word in the document , For instance , the words “Portable” and “Portably” will be treated differently.

**Comparison of Results for Original Queries :**

|  |  |
| --- | --- |
| **Original Queries (On Stopped Corpus)** | **Results** |
| portable operating systems  (Query number = 12) | **BM25**   |  | | --- | | CACM-3127.txt | | CACM-2246.txt | | CACM-3068.txt | | CACM-1930.txt | | CACM-1462.txt  **Smoothed Query Likelihood**   |  | | --- | | CACM-3127.txt | | CACM-1461.txt | | CACM-3068.txt | | CACM-1462.txt | | CACM-2111.txt  **tf-idf**   |  | | --- | | CACM-3127.txt | | CACM-1461.txt | | CACM-3068.txt | | CACM-2246.txt | | CACM-2319.txt | | | |
| parallel algorithms  (Query number = 19) | **BM25**   |  | | --- | | CACM-2973.txt | | CACM-3075.txt | | CACM-0950.txt | | CACM-1601.txt | | CACM-2266.txt |   **Smoothed Query Likelihood**   |  | | --- | | CACM-2973.txt | | CACM-3075.txt | | CACM-0950.txt | | CACM-2266.txt | | CACM-1601.txt |   **tf-idf**   |  | | --- | | CACM-2973.txt | | CACM-1262.txt | | CACM-2714.txt | | CACM-0371.txt | | CACM-0141.txt | |
| performance evaluation and modelling of computer systems  (Query number = 25) | **BM25**   |  | | --- | | CACM-2318.txt | | CACM-1938.txt | | CACM-2319.txt | | CACM-3089.txt | | CACM-3119.txt |   **Smoothed Query Likelihood**   |  | | --- | | CACM-2318.txt | | CACM-2319.txt | | CACM-2268.txt | | CACM-1938.txt | | CACM-2812.txt |   **tf-idf**   |  | | --- | | CACM-2318.txt | | CACM-1653.txt | | CACM-2984.txt | | CACM-1938.txt | | CACM-1374.txt | |

In the table above , the results for the BM25 model are as expected because of the relevance information given in the file “cacm.rel.txt”.

For query 12 , 19 and 25 the top 5 results includes the documents mentioned in the “cacm.rel.txt ”file.

For query 12 i.e. , “portable operating systems” the top 5 documents includes CACM-3127 and CACM-2246 which is expected given their relevance information in the “cacm.rel.txt” file for query 12.

For other two models i.e. , Smoothed Query Likelihood and tf-idf the top 5 results have common documents indicating the correctness of the algorithm , the results are different from BM25 indicating lack of relevance information , which was provided to BM25.

**Comparison of Results for Stemmed Queries :**

|  |  |
| --- | --- |
| **Stemmed Queries (On Stemmed Corpus)** | **Results** |
| portabl oper system  (Query number = 1) | **BM25**   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | |  | | --- | | CACM-3127.txt | | CACM-3068.txt | | CACM-2319.txt | | CACM-2379.txt | | CACM-1591.txt | | |  | | **Smoothed Query Likelihood**   |  | | --- | | CACM-3127.txt | | CACM-2246.txt | | CACM-3196.txt | | CACM-1461.txt | | CACM-2593.txt |   **tf-idf**   |  | | --- | | CACM-3127.txt | | CACM-1461.txt | | CACM-2246.txt | | CACM-3068.txt | | CACM-2796.txt | | |  | |  | |
| parallel algorithm  (Query number = 3) | **BM25**   |  | | --- | | CACM-2664.txt | | CACM-2685.txt | | CACM-1262.txt | | CACM-2700.txt | | CACM-1828.txt |   **Smoothed Query Likelihood**   |  | | --- | | CACM-2714.txt | | CACM-2973.txt | | CACM-2266.txt | | CACM-3075.txt | | CACM-3156.txt |   **tf-idf**   |  | | --- | | CACM-2664.txt | | CACM-0141.txt | | CACM-1302.txt | | CACM-0392.txt | | CACM-2685.txt | |
| perform evalu and model of comput system (Query number = 6) | **BM25**   |  | | --- | | CACM-2318.txt | | CACM-3070.txt | | CACM-3048.txt | | CACM-2988.txt | | CACM-3147.txt |   **Smoothed Query Likelihood**   |  | | --- | | CACM-2318.txt | | CACM-3070.txt | | CACM-3048.txt | | CACM-2741.txt | | CACM-2319.txt |   **tf-idf**   |  | | --- | | CACM-2318.txt | | CACM-2984.txt | | CACM-3070.txt | | CACM-1653.txt | | CACM-3089.txt | |

For the stemmed corpus , Stemmed queries 1,3 and 6 were run on the models BM25 , Smoothed Query Likelihood and tf-idf .

The effect of the stemming were observed with varying degrees on the queries , for instance , in query 1 , which is the stemmed version of original query 12 , only one relevant document i.e.

CACM-3127 is present.

There is similarity observed between the top 5 results of BM25 model (with relevance information) and other models , with the stemming because the relevance information effect has been decreased , since all the words belonging to same stem class were reduced to same stem.

This behavior is opposite to the observation of the original corpus (with stopping) where relevance information evidently played a major role.

Results

|  |  |
| --- | --- |
| Task 1 : Baseline\_run\_BM25\_result | IR\_Project\Task 1\BM25\BM\_25\_RESULTS.xls |
| Task 1 : Baseline\_run\_Lucene\_result | IR\_Project\Task 1\Lucene\ Lucene\_Scores\_RESULTS.xls |
| Task 1 : Baseline\_run\_Smoothed\_Query\_ Likelihood \_result | IR\_Project\Task 1\ Smoothed Query Likelihood Model\ Jelinker\_Scores\_RESULTS.xls |
| Task 1 : Baseline\_run\_ tf-idf \_result | IR\_Project\Task 1\ tf-idf\ TF-IDF\_Scores\_RESULTS.xls |
| Task 2 : Psedo\_relevance\_Feedback\_result | IR\_Project\Task 2\BM25\ BM\_25\_PSEUDO\_RESULTS.xls |
| Task 3 : Stemming\_BM25 | IR\_Project\Task 3\Three\_baseline\_runs\_for\_Stemming\BM25\ BM\_25\_STEMMING\_RESULTS.xls |
| Task 3 : Stemming\_Smoothed\_Query\_Likelihood\_Model | IR\_Project\Task 3\Three\_baseline\_runs\_for\_Stemming\Smoothed Query Likelihood Model\ Jelinker\_Scores\_STEMMED\_RESULTS.xls |
| Task 3 : Stemming \_tf-idf | IR\_Project\Task 3\Three\_baseline\_runs\_for\_Stemming\tf-idf \TF-IDF\_Scores\_STEMMING\_RESULTS.xls |
| Task 3 : Stopping\_BM25 | IR\_Project\Task 3\Three\_baseline\_runs\_for\_Stopping\BM25\ BM\_25\_STOPPING\_RESULTS.xls |
| Task 3 : Stopping \_Smoothed\_Query\_Likelihood\_Model | IR\_Project\Task 3\Three\_baseline\_runs\_for\_Stopping\Smoothed Query Likelihood Model\ Jelinker\_Scores\_STOPPING\_RESULTS.xls |
| Task 3 : Stopping\_tf-idf | IR\_Project\Task 3\Three\_baseline\_runs\_for\_Stopping\tf-idf\ TF-IDF\_Scores\_STOPPING\_RESULTS.xls |
| MAP |  |
| MRR |  |
| P @ K |  |
| Precision and Recall |  |