Search Engine Design and Evaluation

CS 6200: Information Retrieval
Northeastern University
Fall-2017 Prof. Nada Naji

Team Members:
Burhan Sadliwala
Shreysa Sharma
Akash Parikh

Table of Contents

1. Introduction

1.1 Overview

The purpose of this project is to build our own information retrieval systems, evaluate and compare them in terms of retrieval effectiveness, four distinct base retrieval models implemented are as follows:

1. Tf-Idf Retrieval Model
2. BM25 Retrieval Model
3. Smoothed Query Likelihood Model
4. Lucene Systems

We have performed additional runs on the above retrieval systems as follow:

1. BM25 Retrieval Model (Pseudo-relevance feedback)
2. Tf-Idf Retrieval Model (With Stemming)
3. Tf-Idf Retrieval Model (With Stopping)
4. BM25 Retrieval Model (With Stemming)
5. BM25 Retrieval Model (With Stopping)
6. Smoothed Query Likelihood Model (With Stemming)
7. Smoothed Query Likelihood Model (With Stopping)
8. Proximity-enabled Search with Tf-Idf
9. Proximity-enabled Search with Tf-Idf (With Stopping)

The performance of the retrieval systems has been evaluated in terms of retrieval effectiveness using the following measures:

1. Precision and Recall
2. Precision @ K (P@K) where K equals to 5 and 20
3. Mean Average Precision (MAP)
4. Mean Reciprocal Rank (MRR)

1.2 Team Members Contribution:

1. Burhan Sadliwala: Burhan was responsible for performing the four baseline runs BM25, Tf-Idf, Smoothed Query Likelihood model and Lucene System. He has also performed Task3 which included performing stopping and stemming on the baseline runs (except Lucene). He implemented the Proximity-enabled Search and was responsible for the documentation of the same. He, along with Akash, implemented the various evaluation measures like MAP, MRR, P@K, Precision and Recall for the eight distinct runs.
2. Shreysa Sharma: Shreysa was responsible for the implementation of Pseudo-Relevance Feedback and Query Expansion Technique. She was also responsible for documenting her design choices for the same and plotting time-graphs for each run. She also made some contribution in documenting the outlook for this project.
3. Akash Parikh: Akash worked on the report performing query by query analysis and writing the conclusion and bibliography.

2. Literature and Resources

2.1 Techniques Used:

The different techniques and concepts used in this project to perform different retrieval processes can be categorized in two ways:

2.1.1 Retrieval Models Implemented:

1. Tf-Idf Model: We use the normalized value of the term frequency multiplied by the inverse document frequency for calculating the tf*idf score.
2. BM25 Model: We are using the scoring function of BM25 to calculate the document scores. The values of k1, k2 and b are set according to TREC Standards.
3. Smoothed Query Likelihood Model: We calculate probability P(Q|D) for each query term. Smoothing is done using Jelinek-Mercer Smoothing Technique.
4. Lucene Model: We are using the standard Lucene Open Source Library to perform indexing and searching.
5. Pseudo Relevance Feedback Model: We pick up the top k documents from the search results of BM25, assign term weight to each term and pick up top n terms to expand the query.
6. Proximity-Enabled Search: We performed this by scoring the top 100 documents retrieved after running Tf-Idf Model on the queries and displaying only the documents whose score is above a given threshold.

2.1.2 Evaluation Concepts Implemented:

1. Precision & Recall: Following formulae are used to calculate precision and recall:
    - Precision = |Relevant ∩ Retrieved| / |Retrieved|
    - Precision = |Relevant ∩ Retrieved| / |Relevant|
2. Mean Average Precision (MAP): Following formulae are used to calculate precision and recall:
    - MAP = ∑ AveragePrecision / NumberOfQueries
3. Mean Reciprocal Rank (MRR): Get the Reciprocal Rank i.e. the reciprocal of the rank at which the first relevant document is retrieved. MRR is the average of the Reciprocal Ranks over all the queries.
4. P@K: Calculating P@K by finding the precision value at the kth rank. We have used 5 and 20 for the current project.

2.2 Tools Used:

1. Beautiful Soup: Beautiful soup is used to generate the corpus and index and also to store the query terms.
2. Lucene Open Source Library: To perform Lucene search following libraries are needed.

2.3 Research and Articles:

- The following research paper was referred for designing the proximity-enabled search: https://etd.ohiolink.edu/rws_etd/document/get/case1197213718/inline
- The logging code was referred from: https://stackoverflow.com/questions/40858658/python-logging-to-stdout-and-log-file

3. Implementation and Discussion:

3.1 Tf-Idf Retrieval Model:

Term Frequency and Inverse Document Frequency for term t in document d is calculated using
Term Frequency = Frequency of term t in d/Total number of terms in d.
Inverse Document Frequency = log(Total Number of Docs/Number of Docs with term t)
Steps to perform Tf-Idf Retrieval:

1. Calculate Tf-Idf score for each document having with respect to each query term t.
2. Sorting the documents as per their score.
3. Display the top 100 documents.

3.2 BM25 Retrieval Model:

BM25 is a popular and effective ranking algorithm based on Binary Independence model. In BM25 we are calculating the score for individual document using the following formula:

$$\sum \log [(r_i+0.5)/(R- r_i+0.5)]/[(n_i -r_i+0.5)/(N- n_i-R+ r_i+0.5) \ x \ (k_1 +1)f_i / K+f_i \ x \ (k_2+1)qf_i/ k_2+ qf_i$$

Steps to perform BM25 retrieval model:
1. Calculate BM25 score for each document having with respect to each query term t.
2. Sorting the documents as per their score.
3. Display the top 100 documents.

3.3 Smoothed Query Likelihood Retrieval Model:

Score for each document for each query term is calculated using
$\log P(Q|D) = \sum^{n}_{i=1} \log ((1- \lambda).f_{qi,}D/|D| + \lambda.c_{qi}/|C|)$

Steps to perform Smoothed Query-Likelihood Model:
1. Calculate query-likelihood score for each document having with respect to each query term t.
2. Sorting the documents as per their score.
3. Display the top 100 documents.

3.4 Pseudo-Relevance Feedback:

We have performed pseudo-relevance feedback using the following steps:

1. We pick the top 10 documents from the results generated by the BM25 model.
2. From the top K documents take top 3 words using Tf-Idf and ignoring the stopwords and expand the query.
3. Now using the new query run through the initial search algorithm.

3.5 Proximity-Enabled Search:

1. Get the results for a query by running any retrieval model. (We have used Tf-Idf)
2. Compute Proximity-Score for each document and display only the documents which have scores greater than a threshold score.
3. Each doc is scored based on the nearness of the query terms in the documents, Docs with query terms alongside are given more score and the scores are decreased gradually as the number of terms between two query terms increase. Query Term order is maintained.

3.6 Query-by-Query Analysis:

3.6.1 Task 1:

Top 5 Documents Retrieved for Query 1 from the four baseline runs:

Tf-Idf Model:	CACM-1393,2319,1304,2796,2379
BM25 Model:	CACM-2319,1410,2379,1519,1605
Smoothed Query Likelihood Model:	CACM-2319,1410,1605,2379,1393
Lucene Model:	CACM-2319,1657,1410,1827,1938

Analysis:

- Document CACM-2319 is first for all the three systems except Tf-Idf. Due to lower inverse document frequency of terms, CACM-2319 ranked second in Tf-Idf model.
- Document CACM-1410 is second in case of BM25 model and Smoothed Query Likelihood Model and drops at rank 3 for Lucene Model.

3.6.2 Task 2:

The following three stemmed queries were considered to perform analysis:

Query 3: “parallel algorithm”

Top 5 Documents Retrieved:

BM25 Model:	CACM-2714,2973,0950,2433,2785
Smoothed Query Likelihood Model:	CACM-2266,2714,2973,0950,3156
Tf-Idf Model:	CACM-2664,2714,2973,0141,1262

Analysis:

- Document CACM-2714 is first in BM25 and drops to second in Smoothed Query Likelihood and Tf-Idf Model. CACM-2714 contains query term ‘algorithm’ 5 times and ‘parallel’ 7 times.
- Document CACM-2793 is second in BM25 and drops to third in case of other two models.
- In Tf-Idf model score drop in top 5 documents is gradual from 0.163 to 0.141 at rank 5. In case of BM25 model also, the score shows a gradual drop.

Query 7: “parallel processor in inform retriev”

Top 5 Documents Retrieved:

BM25 Model:	CACM-1811,2967,2714,2973,3156
Smoothed Query Likelihood Model:	CACM-2967,1811,2973,2664,2714
Tf-Idf Model:	CACM-1811,2288,2278,0891,1457

Analysis:

- Document CACM-1811 is at rank 1 in BM25 and Tf-Idf model and drops to second in Smooth Query Likelihood Model. CACM-1811 came up first as it has high frequency of the term ‘parallel’.
- A sharp fall in the score is observed in case of BM25 results.

Query 2: “code optim for space effici”

Top 5 Documents Retrieved:

BM25 Model:	CACM-2748,2491,2897,2033,2495
Smoothed Query Likelihood Model:	CACM-2748,2491,1795,2530,2701
Tf-Idf Model:	CACM- 1795,2897,2748,2495,2491

Analysis:

- Document CACM-2748 is at rank 1 in BM25 and Smoothed Query Likelihood model and drops at rank 3 in Tf-Idf Model. CACM-2748 is a highly relevant from a user’s perspective as it contains data about code efficiency.
- Document CACM-2491 is at second in BM25 and Smoothed Query Likelihood model and drops to fifth position in case of Tf-Idf model due to low inverse document frequency.
- In BM25 model score drop in top 5 documents is sharp as it falls from 12.261 at rank 1 to 11.661 at rank 2 and then to 10.75 at rank 3. A gradual drop is observed afterwards.

4. Results

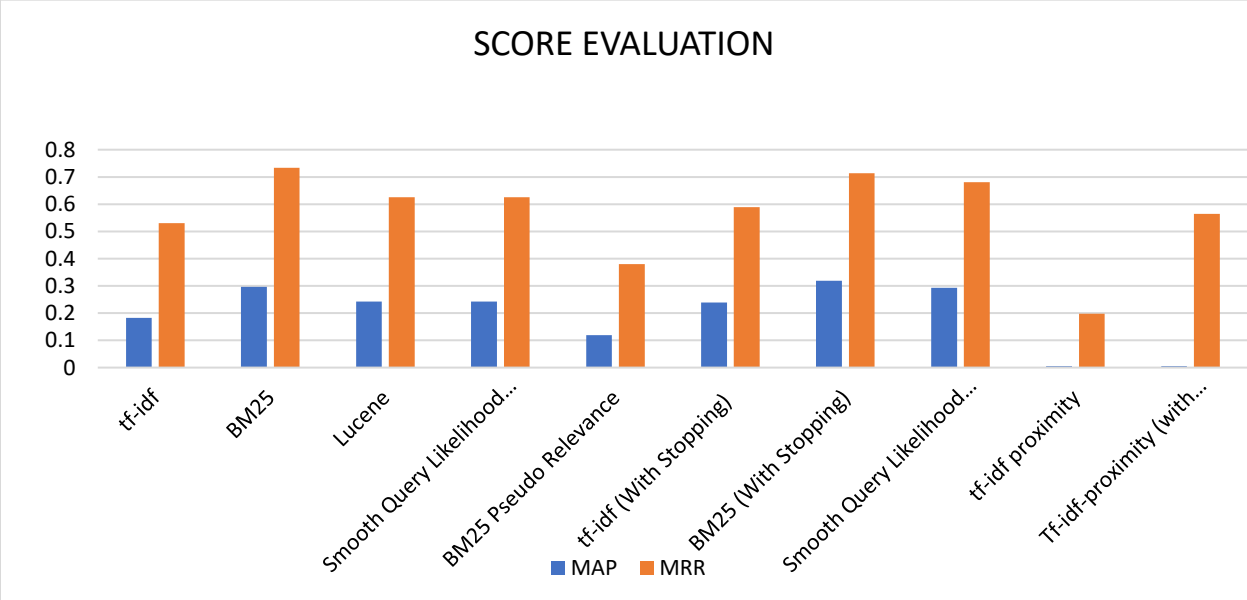
4.1 Query Runs and Precision-Recall Tables:

- The results for all the base retrieval models are stored in text files which can be located inside the Results folder of their respective phases and tasks.
- All the precision and recall tables for the 8 distinct runs can be found in the Results folder in the Phase3 folder.

4.2 Comparing MAP and MRR:

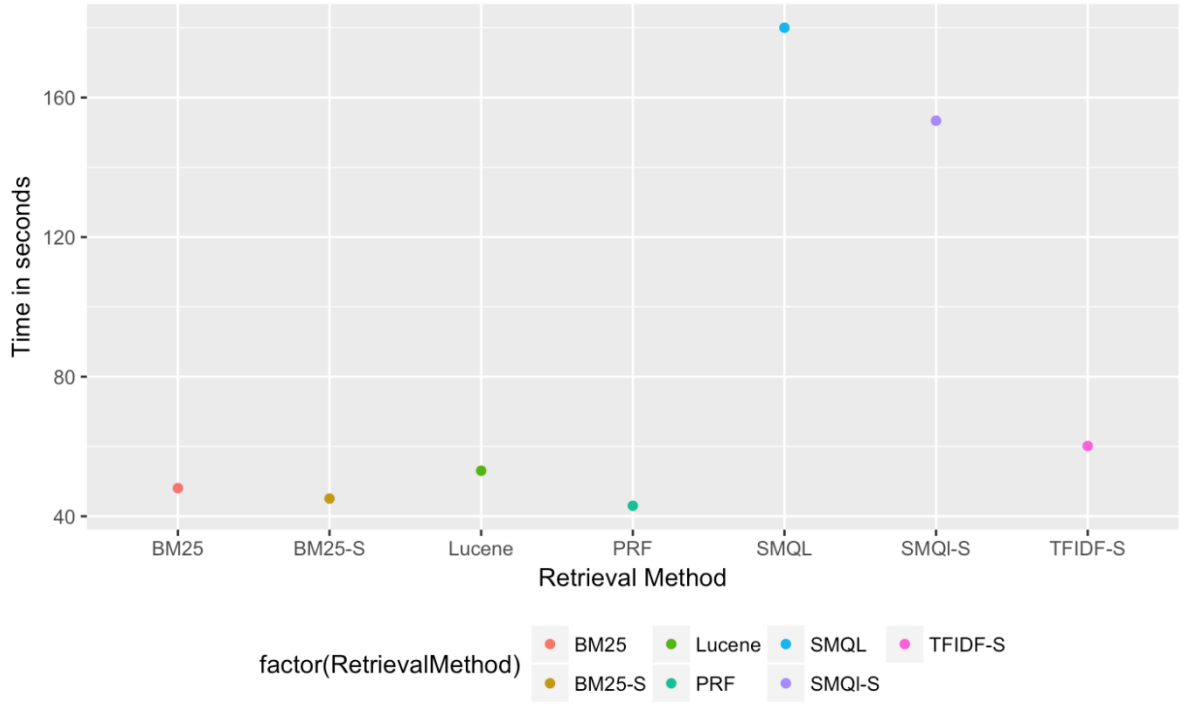
In this section, the comparison of Mean Average Precision (MAP) and Mean Reciprocal Rank (MRR) of different retrieval models with/ without stopping and with/ without proximity-enabled are given. A graph is plotted using the MAP and MRR values for each run which provides further information.

SR No.	Retrieval Model	Mean Average Precision (MAP)	Mean Reciprocal Rank (MRR)
1	Tf-Idf	0.181983	0.530022
2	BM25	0.295994	0.733430
3	Lucene	0.342531	0.525901
4	Smooth Query Likelihood Model	0.242139	0.625961
5	BM25 Pseudo Relevance	0.119439	0.379316
6	Tf-Idf (With Stopping)	0.238806	0.588623
7	BM25 (With Stopping)	0.319192	0.713050
8	Smooth Query Likelihood (Stopping)	0.293437	0.680131
9	Tf-Idf Proximity-Enabled	0.005290	0.198208
10	Tf-Idf Proximity-Enabled (with Stopping)	0.005183	0.564285



4.3 Execution Time:

Following graph is plotted using the execution time of each model run over all the 64 queries.



PRF: Pseudo Relevance Feedback (BM25)
SMQL: Smoothed Query Likelihood Model
SMQL-S: Smoothed Query Likelihood Model-Stopping

5 Conclusions and Outlook

5.1 Conclusions:

Our overall findings after performing all the runs and evaluating on various measures are given below:

- The best results were obtained using BM25 with stopping. Even the base results of BM25 gave the best scores for MAP and MRR. Also, the execution time taken by BM25 is the lowest which further adds to its fineness.
- The lowest result is seen in case of Tf-Idf with the Proximity-enabled search. The MAP score obtained was very low.
- Smoothed Query Likelihood gives comparatively lower scores with respect to MAP and MRR. The downside of using Smoothed Query Likelihood Model is that it takes a lot of time to execute.
- Lucene provides a significantly better MAP and MRR than Smoothed Query Likelihood and Tf-Idf since it incorporates both Boolean Model along with Vector-Spaced Model.
- Thus, BM25 Model performs much better compared to other three systems and its performance can be enhanced when the index is generated using stopping.

5.2 Outlook:

Our project incorporates many of the features of an IR system. However, there's always room for improvement and following features can be included:

- PageRank could be included before the document scoring function in each model which would enable to perform early termination of document searching. Documents with lower PageRank, could be left out while a search is being performed.
- Different compression techniques could be used for the indexes which store term positions such as delta encoding.

6.0 Bibliography

6.1 Books

- "Search Engines: Information Retrieval in Practice," written by W. Bruce Croft, Donald Metzler, and Trevor Strohman.

6.2 Articles, Papers and Websites:

- https://etd.ohiolink.edu/rws_etd/document/get/case1197213718/inline
- https://nlp.stanford.edu/IR-book/html/htmledition/results-snippets-1.html
- https://www.youtube.com/watch?v=yZc24J-Ox2c