CS6200 Information Retrieval

Project Report
Semester: Fall 2017

Submitted by

Mustafa Kapadia
Sarthak Kothari
Shivam Odeka

Under the guidance of

Prof. Nada Naji

Introduction

Description

The goal of this project was to design and build our information retrieval systems as well as evaluate and compare their performance levels in terms of retrieval effectiveness. This project implements text transformation, indexing, retrieval, snippet generation and result evaluation on test data consisting of a raw corpus and 64 queries. All modules except the Lucene module are coded in Python. Lucene is coded in Java.

Contributions

- Mustafa Kapadia:
  - Snippet generation
  - Text processing and transformation, stopping, stemming
  - Documentation

- Sarthak Kothari:
  - Pseudo relevance feedback, Lucene modules
  - Evaluation
  - Documentation

- Shivam Odeka:
  - Tf-idf, BM25, Smoothed query likelihood modules
  - Corpus and query transformations
  - Documentation

Literature and Resources

Retrieval Models

We used the following retrieval models:

1. BM-25
2. Tf-Idf
3. Smoothed Query Likelihood
4. BM25+Qeury expansion using Pseudo Relevance Feedback
5. Lucene (default setting)

Stopping

Stopping is a technique that excludes words that are common in every document and convey no information about the content of the document. Examples: A, the, of etc. The provided common words list is used. When applying stopping, these words are removed from the queries.

Stemming

Stemming involves reducing a word to its root. For example, the words, processes and processing will both be reduced to the word process. Indexing and retrieval have been performed on the stemmed versions of the corpus and queries.

Query Refinement

The query file was case folded and handled for punctuations. Stopping was also used for queries in certain systems as well as during snippet generation. Special care was taken to preserve hyphens and punctuations that appear in numbers. Ex. 22.5. The query was also expanded using pseudo relevant feedback.

Snippet Generation

The window approach was used, where we look for the window which has the largest number of terms appearing in the query, barring stop words. This particular approach was found in http://www.cs.pomona.edu/~dkauchak/ir_project/whitepapers/Snippet-IL.pdf

Implementation and Discussion

Phase 1: Indexing and Retrieval

Task 1:

1. The corpus and queries provided were raw html files and the first step was to clean them. Techniques used- Case folding, remove punctuations
2. Index the documents- Created an inverted index consisting of document ID and term frequencies within the document
3. Implement BM25, tf-idf and query likelihood models. The formulas for all of these were taken from Croft et al. The top 100 results of these are stored in the format query_id Q0 doc_id rank BM25_score system_name in their respective text files
4. For BM25 it is assumed that no relevance information is available.
5. Lucene is left unchanged from HW4

Task 2:

1. Pseudo relevance feedback is performed using the BM25 retrieval module
2. The top 10(n) documents were chosen and from them the top 5(k) words after stopping were chosen for query expansion.
3. This was done because these were found to be producing the best overall results for the 64 queries.
4. Larger values would return irrelevant documents due to over expansion and smaller values did not cover the required terms.

Task 3:

1. BM25, TF-IDF, and Smoothed Query Likelihood retrieval models were used to perform stopping and stemming.
2. Stopping was only performed on the query and not on the corpus.
3. For stemming, the provided stemmed corpus and queries were used.

Phase 2: Displaying Results

1. Snippet generation was performed on the results for each query for the BM25 retrieval module.
2. http://www.cs.pomona.edu/~dkauchak/ir_project/whitepapers/Snippet-IL.pdf - Paper referenced for snippet generation technique.
3. A window size of 20 words was chosen for the snippet.
4. The window with the largest number of query terms, barring stop words, was found.
5. This technique was chosen because the snippet would contain the part of the document which contains information regarding the query terms.
6. The query terms in the window were highlighted and made bold in order to stand out.
7. The DocID along with the 20 word snippet is displayed as the result.

Phase 3: Evaluation

1. Queries that have no entries in the relevance judgement were removed. There were 12 such queries.
2. MAP, MRR, P@K, Precision and Recall were found for the 8 different runs. These are- The 4 baseline runs, 3 stopping runs and the query refinement run.
3. The results were stored in excel sheets for analysis.

Query-By-Query Analysis for Stemming

Q1: portabl oper system

Top 5 documents retrieved : 3127, 2246, 1930, 3196, 2593

When we build a retrieval model for the stemmed version of the corpus against the stemmed version of the queries, the results are prone to errors. For example, the document 3127 is most relevant as it contains all three query terms. However, when we analyze the document 2246, we can observe that the document is retrieved as it contains two of the query terms - “portabl AND oper”. Having observed that and the query, the term ‘oper’ should refer to operating but the document 2246 contains the term ‘operation’ which in turn is stemmed down to ‘oper’ and hence results in higher rank.

Q4: distribut comput structur and algorithm

Top 5 documents retrieved : 2914, 2276, 2454, 2949, 2926

So if we look at the query, terms like distribut can expand to distribution and distributions. The document 2914 has higher cumulative frequency of distribution and distributions. This results in better result for the stemmed version of the query.

Q2: code optim for space effici

Top 5 documents retrieved : 2748, 2491, 2897, 2033, 2495

For this query, the stemmed terms do not make a higher contribution for the retrieval. The document 2748 which is the most relevant document according to BM25 contains ‘code’ with the highest frequency. Terms like ‘optim’ and ‘effici’ however make some difference but that also depends upon the expanded terms.

Results

SYSTEM	MAP	MRR
BM25_QNotStopped_CaseFolded_System	0.396226388	0.616622688
Lucene_System	0.420404991	0.697323057
PRF_BM25_Stopped_CaseFolded_System	0.380109525	0.607028901
Smooth_QNotStopped_CaseFolded_System	0.017914487	0.029156845
BM25_QStopped_CaseFolded_System	0.437244055	0.651321675
Smooth_QStopped_CaseFolded_System	0.080268704	0.128289781
TFIDF_QStopped_CaseFolded_System	0.322830125	0.568768278
TFIDF_QNotStopped_CaseFolded_System	0.323805589	0.567311388

Conclusions and Outlook

From the table above we can conclude that the Query Stopped Case Folded BM25 system, with a Mean Average Precision of 0.437 and a Mean Reciprocal Rank of 0.651 and the Lucene System with a MAP of 0.42 and a MRR of 0.697 give us the best results. BM25 beats Lucene marginally when it comes to precision and Lucene slightly beats BM25 when it comes to having the first relevant result at a higher rank. Choosing the better system between these 2 comes down to a personal choice.

As expected BM25 performs better than most of the other systems. This is because it not only uses a TF-IDF like component, but also normalizes the score for documents of different lengths. Had relevance information been used it would have probably performed even better.

In order to improve the evaluation scores for BM25, and as a result improve the project, I would use relevance information for the BM25 model.

It is also noticed that one needs to be very careful with stemming as many terms have the same root, leading to errors. As an improvement, I would use an association measure in creating the stem file.

Another finding is that the results found by query expansion are highly dependent on the values of k and n. Optimum values can only be achieved through testing. But the values which work best for a particular corpus, may not work well for another. Hence special attention needs to be given while picking these values.

Bibliography

1 Manning, Christopher D., Prabhakar Raghavan, and Hinrich Schütze. "Introduction to information retrieval/Christopher D." (2008).

2 Croft, W. Bruce, Donald Metzler, and Trevor Strohman. Search engines: Information retrieval in practice. Vol. 283. Reading: Addison-Wesley, 2010.

Course Notes and Slides of CS 6200 Fall 2017, Northeastern University.
http://www.cs.pomona.edu/~dkauchak/ir_project/whitepapers/Snippet-IL.pdf : For snippets http://www.regex100 : For regular expressions