CS6200 INFORMATION RETREVIAL PROJECT 03

Sarang Hattikar
Master in Computer Science
Fall 2013

1. Problem

This project requires to build a inverted index on a collection of 3204 documents. The index incudes creation of three files ‘document mapping’ file which maps internal and external ids of the documents, a ‘term mapping’ file which contains every term along with id and its corpus statistics and a ‘inverted index’ file that contains term and ids of all the documents along with the associated document length. And use this inverted index for on five variations of search engine.

Following tools are provided for building index from the :

i. a collection of 3204 documents for creating index
ii. A stemmer i.e. porter stemmer to use while creating index
iii. a queries file to run on the given corpus
iv. a qrel file for performance evaluation.
vii. The following five variations of a retrieval system are to be implemented.
1. Vector space model, Okapi-tf
2. Vector space model, Okapi-tf-idf
3. Language modeling, maximum likelihood, Laplace smoothing
4. Language modeling, maximum likelihood, Jelinek - Mercer
5. BM25

2. Implementation:

Approach:

1. Create a file per unique term in the collection that contains term_id of that term along with internal document id of every document in which that particular term is present along with corresponding term frequency.

2. Use data from all this file to create a ‘inverted index’ file for the entire collection and ‘term mapping‘ file which contains corpus statistics for the entire collection.
3. Along with this also create ‘document mapping’ file which actually maps internal id and external ids of all the documents in the collection.

Algorithm:

1. For each term in the collection extract text containts:

For this purpose Beautiful soup library is used which provides a method to remove tags from give html file and extracts text.

2. Remove whitespaces and punctuation marks from the given text
3. Make a list of all the terms in the text
2. For every term in the list:
i. Remove special character (if any)
ii. Check whether that word is in stopList.
a. if not in stipList
b. Pass the term to stemmer
c. Open a file in ‘**append**’ mode with file name same as stemmed_term
d. create a term id for the term with some specific format
e. count the number of occurrences of the word in the list.
f. write term_id, document_id, and associated term_frequency in the file.
g. close the file

4. File for each term will now contain list of all the documents in which that word has occurred along with associated term frequency(tf).
5. ‘inverted_index’ file is created by appending all this file into a single file.
6. ‘term_mapping’ file is created by calculating for each term the total number of lines associated its file contains as document frequency (df) and addition of all the term frequencies of all the documents as collection_term_fequency(ctf)

<u>CHANGES MADE IN SECOND PROJECT</u>:

1. Code for processing query was replaced by the new code as to use Porter_Stemmer for CACM queries.

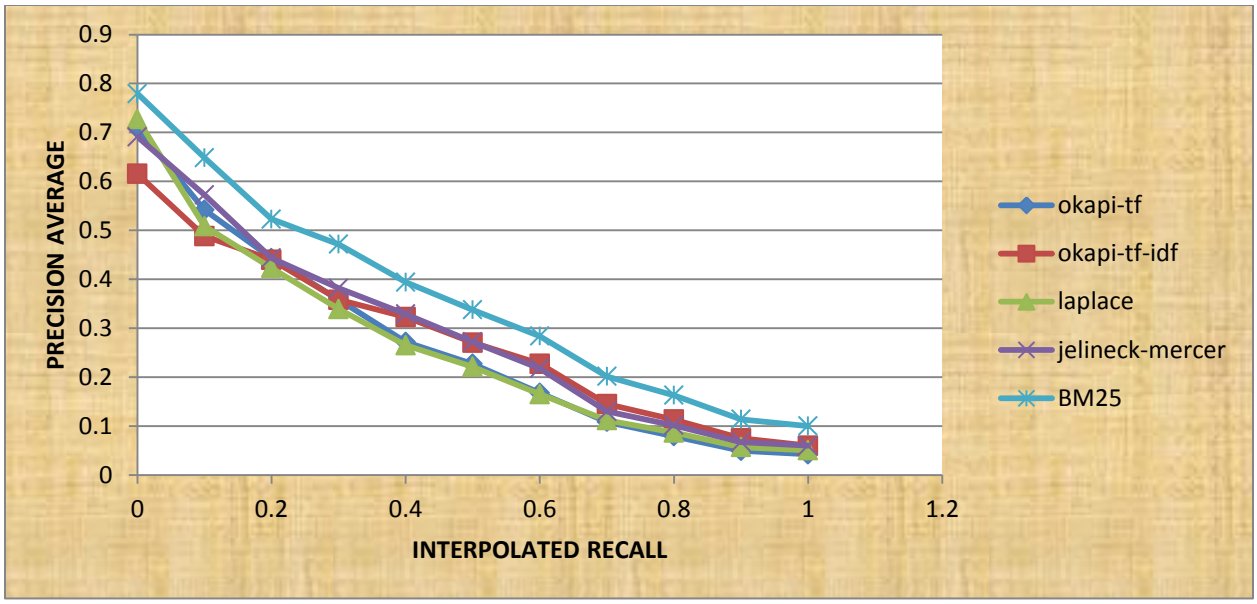
2. For every query :
    1. remove punctuations and extra white spaces.
    2. Create a list of all the words in the query.
    3. Check for each word whether it is present in stopList:
        a. if not present in stop List:
            pass the word to Porter_Strmmer
        b. write the stemmed_word in a file

Result Analysis:

Sr. No	Retrieval Models	MAP	R-Precision	Precision at 10 Documents	Precision at 30 Documents
		Qrel	Qrel	Qrel	Qrel
1	Okapi-tf	0.2512	0.2581	0.5418	0.3581
2	Okapi-tf*idf	0.2599	0.2636	0.2808	0.1878
3	Laplace	0.2471	0.2586	0.2750	0.1590
4	Jelinek-Mercer	0.2805	0.2885	0.3000	0.1878
5	BM25	0.3446	0.3467	0.3442	0.2147

1. Number of document: 3204
2. Number of total terms: 169266
3. Number of unique terms: 11201
4. Average document length: 52.83

Graph for project 3:
Stopping and Stemming applied:



Result Analysis:

1. From the values that I have obtained ,BM25model gives the highest precision among all the five models.
2. From the result obtained it can be seen that it has retrieved highest number of relevant documents.
3. Results of Jelineck-Mercer model were best among all the remaining four model.
4. Precision of okapi-tf model is best as for the 10 and 30 document precision parameter.
5. From the graph we can see that recall values for BM25 model has highest recall values for the entire run. BM25 retrieved highest number relevant documents.
6. okapi-tf*idf model have second highest recall value. After BM25model this model i.e. okapi-tf*idf retrieved most relevant documents.
7. Although Laplace and Jelineck-Mercer both are language model and both implement smoothing Jelineck-Mercer shows high recall values for most of the performance.

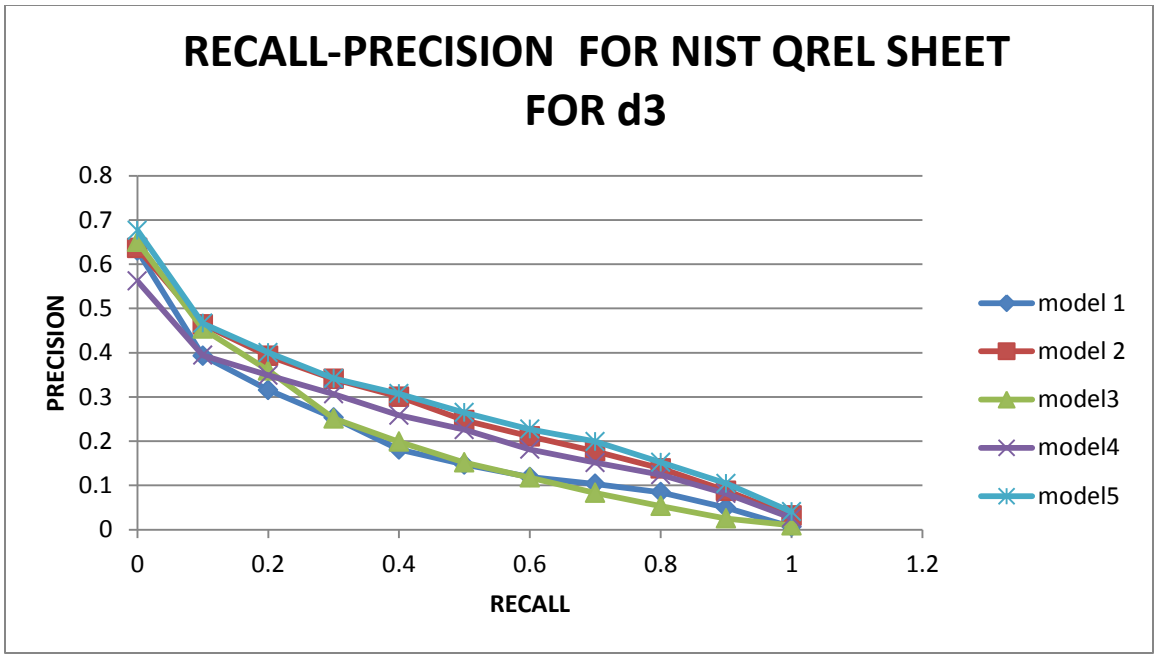
Results obtained for project 2:

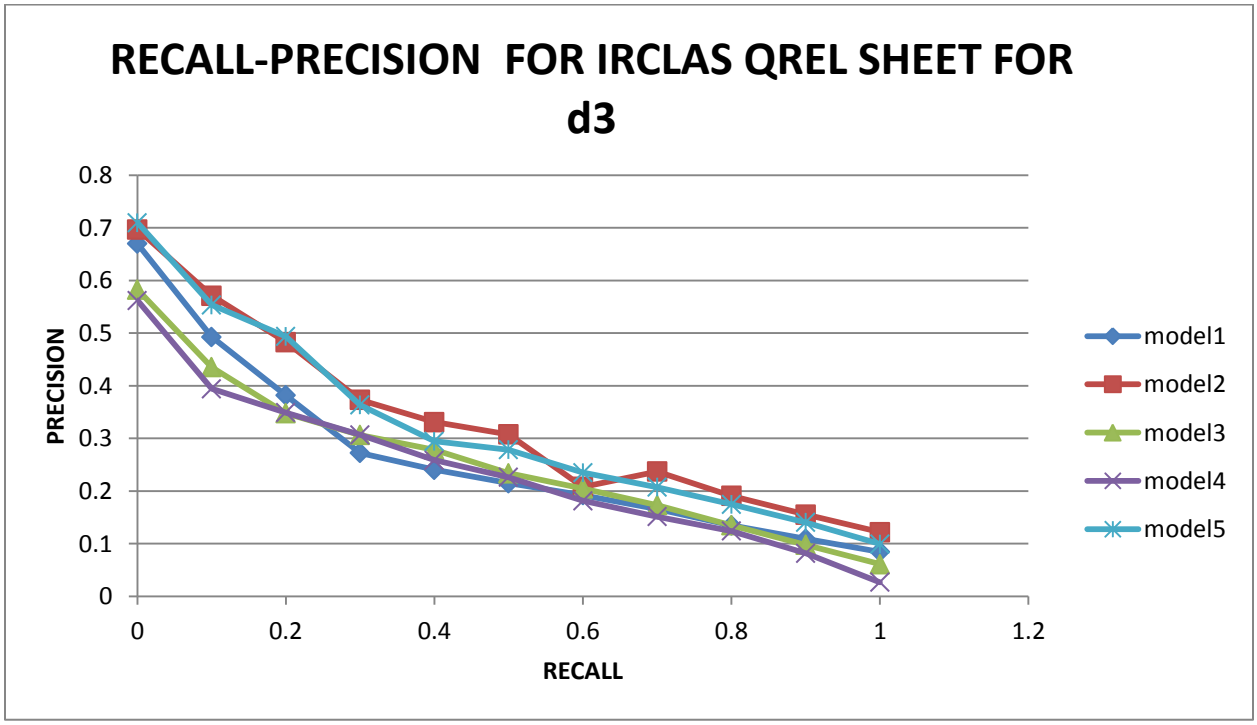
1. Result obtained for running queries on d3 database (stopping and stemming applied)

Sr. No	Retrieval Models	MAP		R-Precision		Precision at 10 Documents		Precision at 30 Documents	
		NIST	IR_Class	NIST	IR_Class	NIST	IR_Class	NIST	IR_Class
1	Okapi-tf	0.1838	0.2517	0.2214	0.2588	0.3360	0.2720	0.2680	0.1947
2	Okapi-tf*idf	0.2566	0.3170	0.2808	0.3111	0.400	0.3400	0.3228	0.2480
3	Laplace	0.1886	0.2417	0.2296	0.2343	0.3640	0.3160	0.2973	0.2427
4	Jelinek-Mercer	0.2267	0.2371	0.2506	0.2300	0.3320	0.2800	0.2800	0.1907
5	BM25	0.2685	0.3017	0.2826	0.2796	0.3720	0.3360	0.3213	0.2373

'd3' database have following properties:

1. Number of document: 84678
2. Number of terms: 24401877
3. Number of unique terms: 166054
4. Average document length: 288


RECALL-PRECISION FOR NIST QREL SHEET FOR d3
PRECISION
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
0.2
0.4
0.6
0.8
1
1.2
RECALL
model 1
model 2
model3
model4
model5


RECALL-PRECISION FOR IRCLAS QREL SHEET FOR d3
PRECISION
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
0.2
0.4
0.6
0.8
1
1.2
RECALL
model1
model2
model3
model4
model5

Observation and analysis of project2 and project 3:

1. From the values obtained for project 2 it can be seen that model 2 i.e. okapi-tf*idf model performed best among all the models while for project 3 its BM25 model which gave best result.
2.BM25 model performed consistent among all the variation as it gave precision value of 0.30 and 0.34 for first and second variation.
3. Jelineck-Mercer model also performed also performed in consistent way in both the variations . As it was third best performer for the previous variation and it is second best performer for this variation.
4. For both variation it can be seen that okapi-tf*idf model performs exceptionally well for 10 and 30 document precision parameter.
5. BM25 model have highest recall values for both the variations.
6. okatf-tf*idf model has second highest recall values for both the variations. So after BM25 this model retrieved most relevant documents.
7. Although Laplace and Jelineck-Mercer both are language model and both implement smoothing Jelineck-Mercer shows high recall values for most of the performance. For both the variations i.e. project2 and project 3 jelineck-mercer model performed best among language models.

EXTRA RUN:

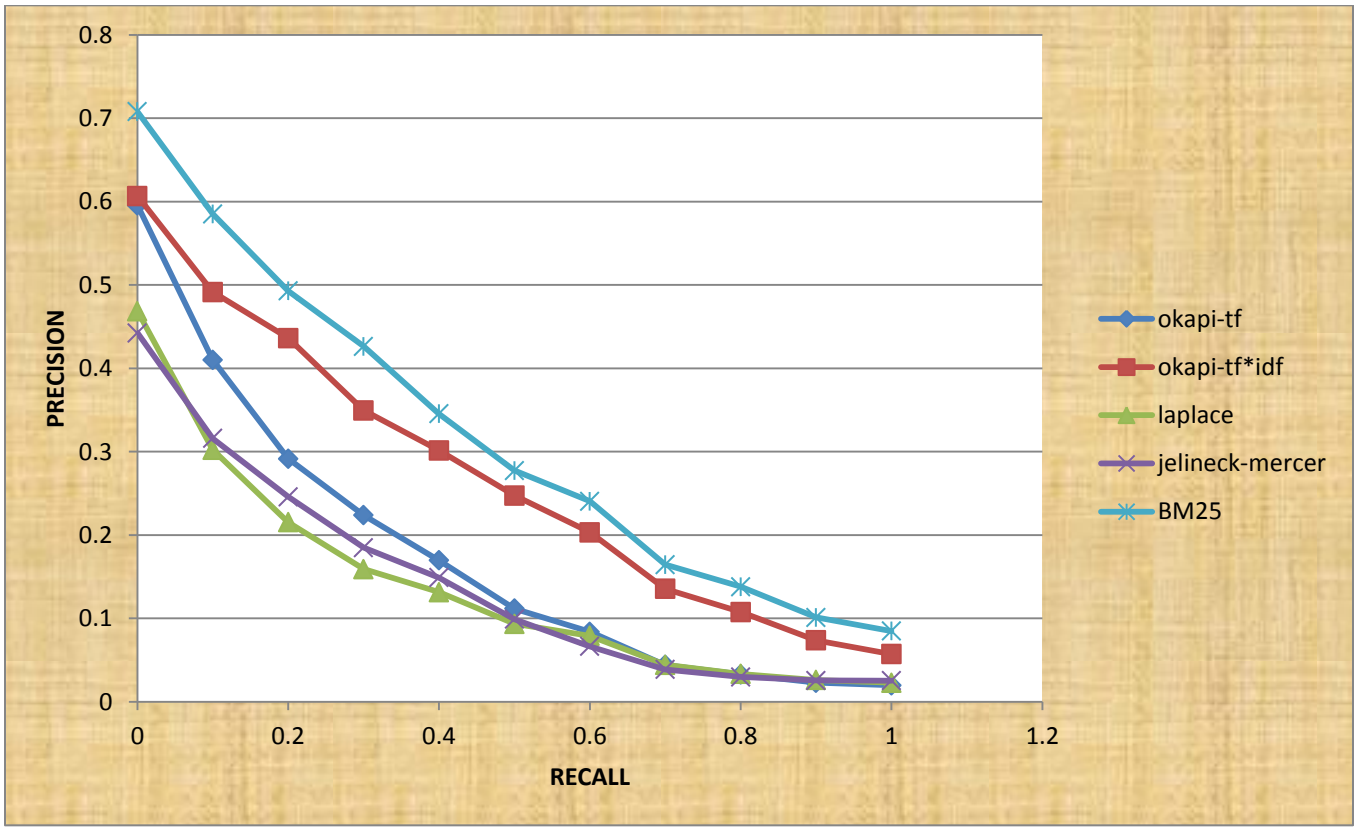
1. If index is created after removing stopping words:
details:
1. Number of document: 3204
2. Number of total terms: 169266
3. Number of unique terms: 11201
4. Average document length: 52.83

2. If index is created without removing words from stoplist following are the details:
Number of unique words: 11418
Number of total words : 246731
Number of documents: 3204
Average document length: 77.00
Time to create the index: approximately extra 10-15 seconds are required to create

Index without removing stopwords.

Result:

Sr. No	Retrieval Models	MAP	R-Precision	Precision at 10 Documents	Precision at 30 Documents
		Qrel	Qrel	Qrel	Qrel
1	Okapi-tf	0.1649	0.1732	0.2269	0.1269
2	Okapi-tf*idf	0.2504	0.2514	0.2673	0.1897
3	Laplace	0.1265	0.1419	0.1558	0.1013
4	Jelinek-Mercer	0.1343	0.1691	0.1942	0.1186
5	BM25	0.3053	0.3149	0.3288	0.1962



Observation &Analysis:

1. if stopwords are not removed ,from the graph and table we can see that performance of okapi-tf model decreased by a very large margin.
2. Performance of BM25 still remains best among all the model.
3.BM25 still has highest recall values among all the model
4. Okapi-tf*idf still remains the second best performer among all the five models
5. Performance of laplace and jelineck-mercer model also varied with a very large margin.
6. But as as like previous variation among language models jelineck-mercer model still performs best.

References & Courtesy:

While developing this project I have collaborated with Srikar Demangu Reddy.