COSC - 488

IR - Search Engine P3

Venkata Sai Varun Uppala

Status : Complete

Time : 25 hours

Things I wish I'd known earlier : N/A

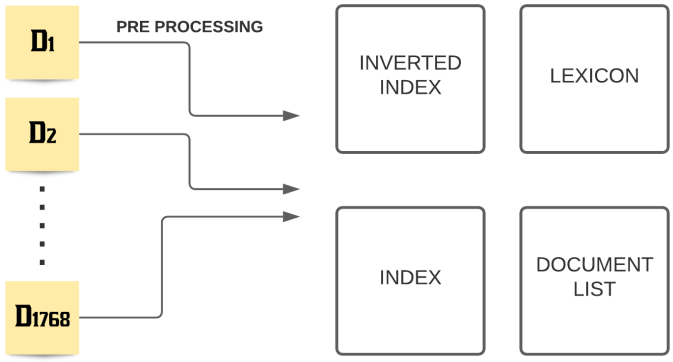
Contents Page No.

1) Recap

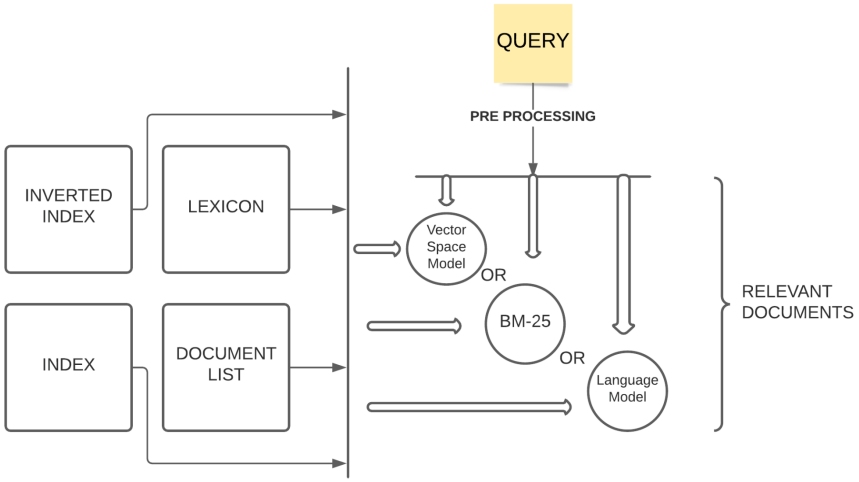
Part - 1: Document Preprocessing and Indexing

4 indices were built :

1. Single Term Index
2. Stem Term Index
3. Phrase Index
4. Proximity Index



Part 2: Query Processing and Document Retrieval



2) System Design

We consider a newly created query during the query reformulation process and as we apply a few strategies to optimize the MAP and/or retrieval processing time

2.a) Query Expansion:

The purpose of query expansion is to improve the precision of the retrieval process by adding terms to the query that appear frequently in previously retrieved documents and then rerunning the procedure to discover the relevant documents. The query was expanded using the **pseudo relevance feedback** method during this procedure.

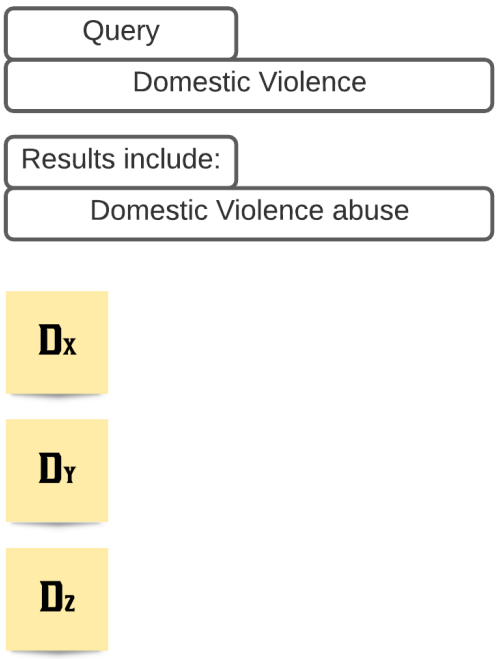


Figure : Query Expansion

The query expansion method involves executing the query against a group of documents and retrieving the documents that we believe are relevant. When the BM25 approach is used to the single term index**, the relevancy of the documents is determined.

*** NOTE : Highest precision was obtained when the Single Term Index was issued against the BM25 retrieval model in P2.*

Our goal after identifying the relevant documents was to expand the query using terms in order to retrieve more relevant documents utilizing the results. This was accomplished by determining the most likely relevant term to the query and attaching it to the original query.

Once the relevant documents have been found, we select m documents and form a pool that contains all the words from these documents that are not already present in the query, followed by their term frequency within these documents. Each of the term frequencies is multiplied by the document frequency of the associated word to weight these words. That is to say,

Example : *If Domestic Violence is the query and abuse has the term frequency of 5 and it appears in 2 documents from the top 3 documents selected, then the weight of abuse would be*

$$w_{abuse} = tf \times df$$

$$w_{abuse} = 2 \times 3$$

$$= 6$$

tf = *Term Frequency of the word*

df = *Document Frequency of the word*

Each term's weights are determined in this manner, and they are listed in increasing order. After that, the top 'n' terms are chosen and added to the original query. The new query is then executed on the documents again, resulting in the retrieval of new relevant documents.

Different combinations of top **documents (m)** and top **terms (n)** were tried out to evaluate the procedure, and their **Mean Average Precision (MAP)** was presented in the results and analysis section.

2.b) Query Reduction:

The purpose of query reduction is to reduce the size of the query by a particular proportion in order to improve precision and reduce the time it takes to get relevant data. The narrative portion of the queries was used to fetch the query's relevant documents.

Example : *<narr> Narrative: A discussion of domestic violence and the amount of sanctioned violence in American culture. Spousal abuse. Repeat offenders. It has been noted that immigrant women are especially vulnerable to domestic violence. The cultures of their home countries often prevent these women from seeking help.*

Suppose, if we cut the query by 10%, we will consider 90% of the query terms to be part of the new query. That is, if the query is "I am the only one" and the threshold is 10%, then 5 * (0.9) = 4 terms are taken into account. (Fractional number terms aren't allowed.)

Terms are assessed based on their document frequencies, which means that the most often appearing terms are excluded from the query, resulting in a more efficient process.

The results table, which can be found in the Results and Analysis section, shows what proportion of terms provide better **Mean Average Precision**.

2.c) Query Expansion and Reduction:

During the query expansion and reduction procedure, both of the current strategies are taken into account. This procedure was carried out on the query's narrative sections. That is, the query is first evaluated using the narrative, then it is reduced by a particular percentage, and finally it is expanded further by adding the most likely relevant terms as part of the query expansion. {For more information, see 2.a.) Query Expansion}

Example : *<narr> Narrative: I am the only one.*

Reduction = 10%

documents = 3

Terms = 2

Query after query reduction : I am the only

Query after expansion : I am the only x . (When x is the most probable relevant term)

So, the final query would be "I am the only x".

Please refer to the Results and Analysis section for results regarding the above mentioned process.

3) Results and Analysis

Below are the results for different methods we have taken into account to analyse our techniques.

1) Query Expansion:

From the results table, the best results are obtained when the combination 3 X 2 was used, moreover the average time taken to obtain better results is 0.055 seconds. During the process of query expansion, the time taken to retrieve better MAP is increased but due to this better MAP has been achieved. It is also clear that better results are being obtained when only 1 term is being appended to the query. Also, higher precision is observed at P@2, P@3. All the combinations with better results are in **bold** and the highest combination has been highlighted.

Combination	MAP	Expanded Query Processing Time
Original Query	0.4361	0.016s

Combination (MXN)	MAP	Expanded Query Processing Time(in seconds)
2 X 1	**0.4708**	0.032s
2 X 2	**0.4443**	0.044s
2 X 3	**0.4505**	0.051s
3 X 1	**0.4778**	0.004s
3 X 2	**0.4979**	**0.053s**
3 X 3	**0.4765**	0.064s
4 X 1	**0.4433**	0.043s
4 X 2	0.4271	0.059s
4 X 3	**0.4375**	0.072s
5 X 1	0.4347	0.046s
5 X 2	**0.4387**	0.059s
5 X 3	0.3905	0.074s
10 X 1	**0.4437**	0.090s
10 X 2	0..4351	0.106s
10 X 3	0.3871	0.160s

2)Query Reduction:

From the results table, the best results have appeared when the query has been reduced by 40%. The application has shown good performance in terms of reducing the query processing time to retrieve better precise results. The average time taken to retrieve better results is 0.023 seconds. The graphs below show a comparison of the MAP and Query processing times. All the combinations with better results are in **bold** and the highest combination has been highlighted.

Reduction (x%)	MAP	Expanded Query Processing Time
Original Query (Narr)	0.3644	0.10s

Reduction (x%)	MAP	Reduced Query Processing Time(in seconds)
10	0.3560	0.06s
20	**0.3785**	0.04s
30	**0.3787**	0.03s
40	**0.4317**	**0.02s**
50	**0.3946**	0.01s
60	**0.3719**	0.008s

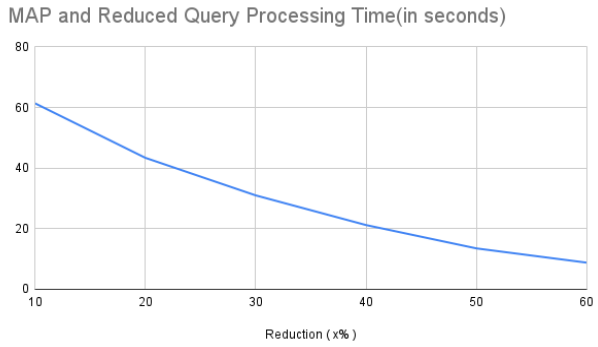


Fig : Reduction vs Seconds

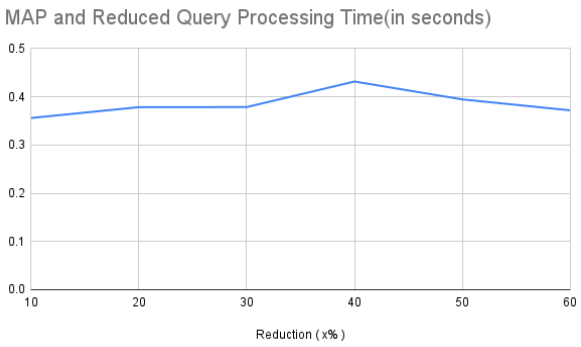


Fig : Reduction vs MAP

3) Query Expansion and Reduction :

The results table shows that the best results have been achieved when the query was reduced with 40% and 2X3 combination was used. The average time taken to expand and reduce the query for better results than the baseline model is 0.0805 seconds which is clearly less than the baseline query processing time. Also, better results are observed when the query is reduced by 30% and 40%. All the combinations with better results are in **bold** and the highest combination has been highlighted.

Type of Query	MAP	Query Processing Time
Original Query (Narr)	0.3644	0.102s

Reduction (x%)	Combination (MXN)	MAP	New Query Processing Time(in seconds)
10	2 X 1	0.3641	0.089s
	2 X 2	**0.3846**	0.096s
	2 X 3	**0.3775**	0.106s
	3 X 1	0.3636	0.101s
	3 X 2	0.3450	0.113s
	3 X 3	0.3458	0.121s
	4 X 1	0.3456	0.105s
	4 X 2	**0.3661**	0.122s
	4 X 3	0.3632	0.132s
20	2 X 1	**0.3781**	0.072s
	2 X 2	**0.3825**	0.078s

	2 X 3	**0.3867**	0.087s
	3 X 1	**0.3941**	0.081s
	3 X 2	**0.3767**	0.093s
	3 X 3	0.3508	0.103s
	4 X 1	**0.3662**	0.086s
	4 X 2	**0.3737**	0.104s
	4 X 3	**0.3689**	0.114s
30	2 X 1	**0.3947**	0.056s
	2 X 2	**0.3858**	0.065s
	2 X 3	**0.4015**	0.073s
	3 X 1	**0.3998**	0.069s
	3 X 2	**0.3952**	0.078s
	3 X 3	**0.3651**	0.088s
	4 X 1	**0.3727**	0.072s
	4 X 2	**0.3846**	0.087s
	4 X 3	**0.3799**	0.099s
40	2 X 1	**0.4113**	0.047s
	2 X 2	**0.3900**	0.055s
	2 X 3	**0.4279**	**0.064s**
	3 X 1	**0.4203**	0.055s
	3 X 2	**0.3858**	0.070s
	3 X 3	**0.3646**	0.079s
	4 X 1	**0.3744**	0.062s
	4 X 2	**0.3743**	0.076s
	4 X 3	**0.3833**	0.087s

4) How to run the program?

Please have the packages listed below installed in order to execute the program.

1. ***warnings*** - ignore warnings
2. ***json*** - read and write index
3. ***re*** - identify special tokens
4. ***math*** - calculate scores
5. ***nltk*** - porter stemmer
6. ***itertools*** - iter through dictionaries
7. ***os*** - create directories
8. ***string*** - operations on strings
9. ***time*** - calculate time
10. ***argparse*** - pass command line arguments
11. ***statistics*** - statistics during building index
12. ***sys*** - include path to folders

Command to run :

python build.py documents_path

python query.py -i INDEX_PATH -q QUERY_PATH -r RESULT_PATH -rt {a,b,c}
[-d NUMBER_DOCUMENTS] [-tc TERM_COUNT] [-qp QUERY_PERC]

-i INDEX_PATH,

-q QUERY_PATH,

-r RESULT_PATH,

-rt {a,b,c}, a) Expansion b) Reduction c) Expansion and Reduction

-d NUMBER_DOCUMENTS,

-tc TERM_COUNT,

-qp QUERY_PERC,

Note : You can use run.sh executable to run the program for all the combinations as mentioned above.

Please use :

create_res_script.py -q QRELS_PATH -r RESULTS_PATH

To create a runnable file for all the combinations and use chmod +x run_results.sh to make it a runnable file.

Use ./run_results to find the MAP's for all the combinations experimented on.