
The University Of Sheffield.

# COM 6115 Text Processing

## Assignment Report:

## Document Retrieval

Submission Date: 10 December 2021

Course: **MSc Data Analytics**

Name of Student: **Jagat Kiran Alla**

User Name: **acp21jka**

Registration No.: **210116270**

Lecturer: **Prof. Dr. Mark Hepple**

## Introduction

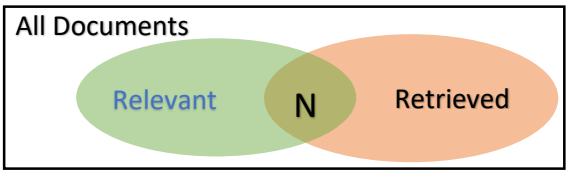
In the assignment, 3 term weighting schemes for the document retrieval system were implemented. The Binary method (Exact matching) was first studied, then the Term Frequency (Similarity to query) method was studied, and finally, the Term frequency- Inverse Document Frequency method (Importance of documents) was studied. This retrieval system was created on Windows 10 using Python 3.8.5 and the JupyterLab IDE. Evaluation results are based on the CACM gold standard result. The gold standard and output results were compared using the 'eval_ir.py' file. A total of 64 queries were passed onto the system for document retrieval on 3204 documents.

## Metrics calculated

For each possible configuration of the term weighting scheme, the following metrics/scores were calculated after comparison of our results with the CACM gold standard (Refer Table 1). The metrics are as follows:

**1. Number of relevant documents retrieved (N)**

After computing the cosine similarity score and ranking the documents using the term weighting scheme, the top 10 documents were chosen as relevant ones for the particular query out of all the ranked documents. For the total 64 queries, that equates to 640 documents retrieved by the system. The CACM in total has 796 relevant documents matched to all the queries.




Relevant=796 docs
Retrieved=640 docs

**2. Precision**

Precision is the proportion of retrieved material which is relevant. It is calculated by dividing the N (Number of relevant documents retrieved) by the actual number of documents retrieved (which in this case in 640)

$$\text{Precision} = {}^{N}/_{Retrieved} = {}^{N}/_{640}$$

**3. Recall**

Recall is the proportion of relevant material retrieved to all the relevant documents. It is calculated by dividing N by the actual number of relevant documents (which is 796 in this case)

$$\text{Recall} = {}^{N}/_{Relevant} = {}^{N}/_{796}$$

**4. F-measure**

F-measure takes both precision and recall into account so that each one cancels out each other's negative attributes. It is given by the (H.M)harmonic mean of recall and precision. Harmonic mean is chosen instead of Arithmetic mean because, for high F-measure, both Recall and Precision must be high.

$$\text{F-measure} = {}^{(2 \times P \times R)}/_{(P+R)}$$

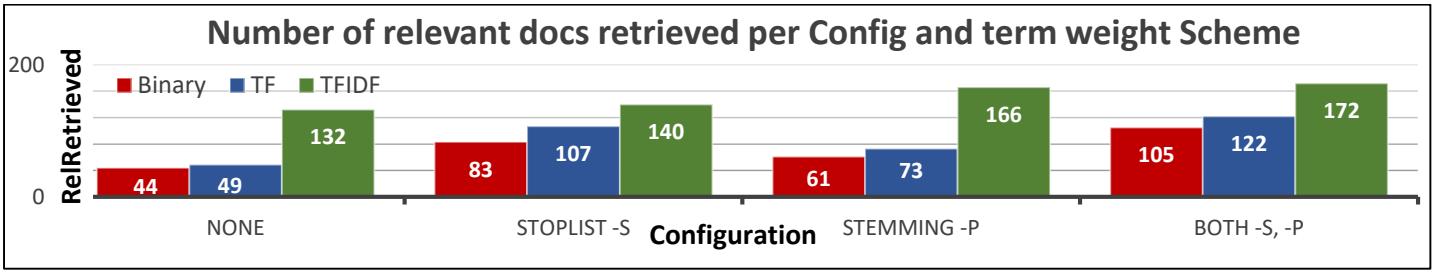
## Term weighting schemes

The retrieval system was implemented for 3 term weighting schemes.
They are - Binary, TF (Term Frequency) and TFIDF (Term Frequency- Inverse Document Frequency)

| Term weighting schemes | Configuration choice | Relevant documents retrieved (N) | Precision (out of 640) | Recall(out of 796) | F-measure | Runtime (seconds) |
|---|---|---|---|---|---|---|
| Binary | No stemming, No stop list | 44 | 0.07 | 0.06 | 0.06 | 0.91s |
| | Only stop list | 83 | 0.13 | 0.10 | 0.12 | 0.28s |
| | Only stemming | 61 | 0.10 | 0.08 | 0.08 | 0.96s |
| | Both stemming and stop list | 105 | 0.16 | 0.13 | 0.15 | 0.38s |
| Term frequency (TF) | No stemming, No stop list | 49 | 0.08 | 0.06 | 0.07 | 1.03s |
| | Only stop list | 107 | 0.17 | 0.13 | 0.15 | 0.32s |
| | Only stemming | 73 | 0.11 | 0.09 | 0.10 | 1.12s |
| | Both stemming and stop list | 122 | 0.19 | 0.15 | 0.17 | 0.43s |
| Term frequency –Inverse Document Frequency (TFIDF) | No stemming, No stop list | 132 | 0.21 | 0.17 | 0.18 | 1.07s |
| | Only stop list | 140 | 0.22 | 0.18 | 0.19 | 0.33s |
| | Only stemming | 166 | 0.26 | 0.21 | 0.23 | 1.15s |
| | Both stemming and stop list | 172 | 0.27 | 0.22 | 0.24 | 0.44s |

***Table.1**- Comparison of the metric scores for each change in the configuration*

**Note:** Only the metric N (relevant retrieved) is used for observations in the bar chart instead of other 3 metrics. It is because, all the other 3 metrics are directly proportional to N and don't rely on any other factor, so examination of this 1 metric alone is sufficient to draw observations about other metrics.




***Bar Chart. 1-*** *Number of relevant docs retrieved per Configuration and term weight Scheme*

## Configurations

Two different kinds of configurations were used in this system for pre-processing of data to filter out useless data. They are- Stop List and Stemming.

## Results/Observations

1. From Table.1 and Bar Chart.1, it can be concluded that the TFIDF scheme provides us with the most relevant retrieval of documents across all the Configurations. This is because TFIDF eliminates the reliance of most common/frequent terms in query and corpus by having a less IDF value and the relevant retrieved docs do not have those common terms and are much more focused on the important terms and terms of interest, thus outputting a better document in any configuration. Hence the rank of documents with more common words matching is lower. Whereas, the most significant terms have a higher IDF score because they are more exclusive to certain documents rather than being present in all documents, and hence are ranked higher up in the document ranking.
2. Looking at the runtimes from the Table.1, we can see that for all schemes, the configuration with only stoplist takes the least amount of time and the configuration with only stemming takes the most amount of time (4 times more than stoplist time). This is because, stoplist eliminates most of the common words, and hence only a few terms that are more significant are left for the similarity score calculation. However, for only stemming, many more terms are taken back into account for calculation as only the root/stem of a term is considered and this adds many terms which had tenses, prefix, suffix into the pool and this makes the term count increase and thus increasing the runtime.
3. It can also be seen from the Graph that for TFIDF, only stemming did give better results than the only stoplist and basic one (without any pre-processing). This is because stoplist does not give many benefits over basic one as all the frequent and insignificant terms are already taken care of by the IDF values. Whereas, after stemming, many new opportunities open up for the system because stemming allows for the same instance of words to be taken into account even though they were represented with tenses, plural, or even prefix/suffix which is very common in documents (E.g. impute, imputed, imputation, etc. are converted to 'imput' after stemming).
4. We can also infer that the TF scheme did better than binary in all the configuration. This is because the TF scheme takes the occurrence of a term into account, and hence, if a document has more of the query terms, it would get a higher similarity score and it would be higher in the retrieval ranking and in turn, produces a better retrieval. Whereas, the Binary scheme just takes exact words as a match and it doesn't provide satisfactory results as most of the documents will have many of the same query terms and this makes the similarity score to be nearly the same for all the matching documents.
5. It can be seen from the table and graph that binary scheme and TF scheme with no pre-processing produced very less number of relevant documents. It is because, in binary and TF, the common words are aplenty in the query and document, and hence during similarity scoring, the relevant documents retrieved are based on these common words. That is, higher the common words, the document will be ranked higher even though it might not be relevant. Thus there is less number of relevant documents retrieved (44 and 49 respectively).
6. One can infer from the graph that only stoplist produced much better results than only stemming in binary and TF scheme whereas in TFIDF, stemming did better than stoplist. This happened because when stoplist is applied, these most common terms are not considered, and hence, the score is based on the relevant/exact terms, and hence the relevant documents retrieved nearly double from the no pre-processing implementation (83 and 107 respectively). However, when it comes to TFIDF, the common terms are already not weighed much, and hence for stoplist, the score does not improve much from the basic configuration.

## Conclusion

In conclusion, the TFIDF term weighting scheme and configuration with stemming and stop list has the best scores for each metric when compared to the gold standard.