Western Michigan University

Design and Evaluation of an Information Retrieval System with Tokenization and Stemming Techniques

Group Members:
Ahmad K. Khanfar
Mamatha Devi Thalari
Aishwarya Doma
Meghana Vasudeva Murthy

Abstract

In the report, the performances of three popular IR ranking methods-Classic Term Frequency-TF, Term Frequency-Inverse Document Frequency-TF-IDF, and BM25-were studied. We experimented with several configurations using different tokenization techniques-Whitespace and NLTK, stemming (yes/no), and stopword removal (yes/no). Precision, recall, F1-score, Precision at rank 10 (P@10), and Normalized Discounted Cumulative Gain (NDCG) are used to evaluate the methods. Finally, the results are discussed in respect of different scenarios with common keywords like "new," "government," and "technology."

1. Introduction

Information Retrieval, or IR, systems are the heart of every modern search engine and document management system. These systems are designed to retrieve relevant documents from a vast collection of data based on the input query provided by the user. The major goal of IR systems is the ranking of documents in such a way that the most relevant ones would be top listed to improve the user's search experience. Traditional IR models, such as Term Frequency (TF), Term Frequency-Inverse Document Frequency (TF-IDF), and BM25, had been widely adopted because of their simplicity, effectiveness, and high computation efficiency.

The simplest ranking functions are Term Frequency, in which the relevance of a document is determined simply by the frequency of the search terms appearing in the document. There are several problems with using **TF** alone: it does not take into consideration the importance of the terms across the entire dataset and cannot differentiate common from rare terms.

TF-IDF is an extension of TF that weights the importance of the terms not only for their frequency within the document but also for their rarity across the entire corpus. This helps to reduce the influence of common words, which may appear in almost every document, making it more robust for document ranking. TF-IDF is considered the mainstay of IR systems because it relies on the fact that important terms are often rare in a corpus and, therefore, leverages this to identify such terms and assigns them higher weights.

BM25 is a probabilistic model which extends the notions of term frequency and document length and also introduces some notion of saturation, which prevents terms occurring in excessively long documents from being too influential. BM25 has received widespread attention regarding its performance for many retrieval tasks, especially in those cases when a more advanced probabilistic framework is needed for modeling relevance.

Each of these methods has strengths and weaknesses depending on the dataset and query characteristics. One of the most important aspects that influences the results of these models is the pre-processing of text data. Pre-processing techniques such as tokenization, stemming, and stopword removal can change the performance of the retrieval model substantially.

The text is tokenized, meaning it breaks the text down into terms or tokens; tokenization methods can imply different levels of granularity for the terms. Stemming reduces words

to their root form, and it may help in matching various inflections of the same word and result in over-simplification. Stopword removal can be done to filter out commonly used words such as "the," "is," and "in" that do not contribute much to the meaning of text. However, if they are not filtered out, they could affect the performance of the ranking model.

This study investigates the performances of ***TF, TF-IDF, and BM25*** within various preprocessing scenarios: different tokenization methods, such as whitespace and NLTK, stemming, and stopword removal, and how these would affect each model's accuracy and effectiveness. We will evaluate the models based on several performance metrics, including precision, recall, F1-score, precision at rank 10 (P@10), and normalized discounted cumulative gain (NDCG). Additionally, we will examine the impact of keyword selection on ranking, using common terms such as new, government, and business.

This study brings to light how such factors influence IR performance, hence giving insight into the best settings that are most effective for retrieving relevant documents with an improved efficiency of IR systems in real-world applications.

2. Methodology

In this section, we describe the methodology employed to evaluate the effectiveness of different Information Retrieval (IR) models—namely TF, TF-IDF, and BM25—under varying preprocessing conditions. We also present the data, preprocessing steps, evaluation metrics, and the experimental setup.

a. **Dataset**

For this work, we had utilized the BBC News dataset, which contains news articles with pre-identified topics. The dataset is popular for the evaluation of document retrieval systems due to its variety in terms of topics and textual content. The dataset was preprocessed to remove irrelevant metadata, leaving only the text content of the articles, to which we apply the evaluation of the retrieval models.

The corpus is a combination of different genres, ranging from business to politics, entertainment, and technology. This was based on assessing the capability of retrieval models with regard to ranking documents relevant to a given query in terms of their textual content.

b. **Preprocessing Techniques**

Prior to evaluating various retrieval models, several preprocessing techniques were applied to the dataset. These techniques are necessary to properly clean up the text and enhance the model's ranking performance.
Tokenization: This is a form of processing in which text, generally broken into words or phrases, is segmented into smaller units. Two methods in this regard were used in this study:
Whitespace Tokenization: This treats text as a sequence of words separated by spaces and hence tokenizes on whitespace.
NLTK Tokenization: This uses the Natural Language Toolkit library for tokenization, which can handle punctuation and more complex word boundaries.
Stemming: It reduces words into their base or root form. "Running" would be stemmed to "run." This assists in grouping various words that come under one same word with different inflections. This paper made use of Porter Stemmer algorithm for stemming.

Stemmed Applied: The words are reduced into their stem form.
No Stemming: The words remain in their original form without any modification.
Stopword Removal: The removal of stopwords, which are those words that are common enough not to add much significance to document retrieval tasks, such as "the," "is," and "in." Removing these can help the model perform better by reducing noise.
Stopwords Out: The stopwords were filtered out using the NLTK stopword list.
No Stopword Removal: No stops were removed; hence, all words from the text remained.

c. **Information Retrieval Models**

We assessed three popular retrieval models, namely, TF, TF-IDF, and BM25. Each model was tested under the following conditions:
Tokenization: Whitespace or NLTK-based tokenization
Stemming: Applied or not applied
Stopword Removal: Applied or not applied
Term Frequency (TF): TF model determines the frequency of a term within a document. This ranks the documents based on term frequency, with those having more frequent occurrences of the query term ranked as more relevant.
Term Frequency-Inverse Document Frequency TF-IDF: This is an improvement of TF. In TF-IDF, the term frequency is modified by the inverse frequency of the term across all documents in the corpus. This helps to give more weight to terms that are rare in the corpus and less weight to common terms, hence improving the

model's ranking ability for more relevant documents.
BM25: The BM25 is an improved probabilistic model over the TF-IDF approach, introducing a factor for document length normalization and term saturation. Indeed, it is considered one of the best performing ranking algorithms, especially on longer documents.

d. **Evaluation Metrics**

Several evaluation metrics were used to assess the performance of the retrieval models

i. **Precision:** Precision is the proportion of retrieved documents that are relevant to the query. It is defined as:

$$\text{Precision} = \frac{\text{Relevant Documents Retrieved}}{\text{Total Documents Retrieved}}$$

ii. **Recall:** Recall measures the proportion of relevant documents that were successfully retrieved. It is defined as:

$$\text{Recall} = \frac{\text{Relevant Documents Retrieved}}{\text{Total Relevant Documents}}$$

iii. **F1-Score**: The F1-Score is the harmonic mean of precision and recall. It is used to evaluate the balance between precision and recall, especially when there is an uneven class distribution. The F1-score is defined as:

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

iv. **Precision at Rank 10 (P@10):** P@10 measures the precision of the top 10 documents returned by the system. It is particularly useful for evaluating the relevance of documents in the first 10 results.

v. **Normalized Discounted Cumulative Gain (NDCG):** NDCG is a measure of ranking quality that accounts for the position of the relevant documents in the result list. It assigns higher relevance to documents that appear earlier in the ranking. NDCG is normalized to ensure that values are between 0 and 1.

$$NDCG_k = \frac{DCG_k}{IDCG_k}$$

Where DCGk is the Discounted Cumulative Gain at rank k, and IDCGk is the Ideal Discounted Cumulative Gain at rank k.

e. **Experimental Setup:**
The experiment was run in the following way: The BBC News dataset was preprocessed with the chosen tokenization, stemming, and stopword removal techniques.
All three retrieval models (TF, TF-IDF, and BM25) were then applied to the preprocessed dataset.
In the end, the rankings from every model were evaluated using these metrics. Results regarding various tokenization techniques, stemming applications, and stopword removal were compared.

3. Results and Discussion

This section will present the results of the experiments conducted on the different IR models, such as TF, TF-IDF, and BM25. These models were compared under various preprocessing conditions: tokenization methods (whitespace and NLTK), stemming-applied or not-and stopword removal-applied or not. We discuss each model's performance with several evaluation metrics: precision, recall, F1-score, precision at 10 (P@10), and normalized discounted cumulative gain (NDCG).

3.1 Performance Overview:

We present the results of the three IR models, namely TF, TF-IDF, and BM25, under the different preprocessing conditions. The metrics used for evaluation are as follows:

a. **Precision:** Proportion of relevant documents retrieved out of all the retrieved documents.
b. **Recall:** Proportion of relevant documents retrieved out of all relevant documents.
c. F1-score: Harmonic mean of precision and recall, giving a single measure of performance.
d. **Precision at 10 (P@10):** Precision of the top 10 retrieved documents.

e. **NDCG**: A ranking metric considering the position of relevant documents in the ranked list.

Below is a summary of the results based on different combinations of preprocessing methods.

i. **TF Model ResultsWith Stopword Removal and Stemming:**
A low performance of this combination is confirmed by the TF model. Precision, recall, and F1-score are similarly low, which indicates that, even after stopword removal and stemming, the model was not able to retrieve a high number of relevant documents.

ii. **TF-IDF Model Results With Stopword Removal and No Stemming:**
Here, where stemming is not applied but stopword removal is performed, the performance of the model goes way higher. That means that stemming in this case introduced some noise, which was reduced with the removal of stopwords.

iii. **BM25 Model ResultsWith Stopword Removal and Stemming:**
BM25 performed worse when stemming was applied with stopword removal. This suggests low precision and recall, implying this setup didn't lead to an effective ranking of the relevant documents. However, NDCG, which takes into consideration the position of relevant documents, is higher when compared to TF; this would suggest that even though relevant documents are retrieved, they are ranked better.

iv. **With Stopword Removal and No Stemming:**
When stemming is not applied, BM25 shows a slight improvement, at least in recall. However, the precision remains very low, meaning that this model retrieved many irrelevant documents. The higher NDCG score suggests that the ranking in BM25 for relevant documents is better, although the general performance is still not as good as in the case of TF-IDF.

Method	Precision	Recall	F1-Score	Tokenization Method	Stopwords Removal	Stemming
TF	0.561644	0.398058252	0.465909091	whitespace	Yes	Yes
TF	0.561644	0.398058252	0.465909091	whitespace	Yes	No
TF	0.561644	0.398058252	0.465909091	whitespace	No	Yes
TF	0.561644	0.398058252	0.465909091	whitespace	No	No
TF	0.561644	0.398058252	0.465909091	whitespace	Yes	No
TF	0.561644	0.398058252	0.465909091	whitespace	Yes	Yes
TF	0.561644	0.398058252	0.465909091	nltk	No	Yes
TF	0.561644	0.398058252	0.465909091	nltk	No	No
TF	0.561644	0.398058252	0.465909091	nltk	Yes	No
BM25	0.561644	0.398058252	0.465909091	nltk	Yes	No
BM25	0.561644	0.398058252	0.465909091	whitespace	Yes	No
BM25	0.561644	0.398058252	0.465909091	whitespace	No	Yes
BM25	0.561644	0.398058252	0.465909091	whitespace	No	No
BM25	0.561644	0.398058252	0.465909091	whitespace	Yes	No
BM25	0.561644	0.398058252	0.465909091	whitespace	Yes	Yes
BM25	0.561644	0.398058252	0.465909091	nltk	Yes	Yes
BM25	0.561644	0.398058252	0.465909091	nltk	Yes	No
BM25	0.561644	0.398058252	0.465909091	nltk	No	No
BM25	0.561644	0.398058252	0.465909091	whitespace	No	No
TF-IDF	1	0.718446602	0.836158192	whitespace	No	No
TF-IDF	1	0.718446602	0.836158192	whitespace	Yes	No
TF-IDF	1	0.718446602	0.836158192	whitespace	Yes	Yes
TF-IDF	1	0.718446602	0.836158192	whitespace	No	Yes
TF-IDF	1	0.718446602	0.836158192	whitespace	No	No
TF-IDF	1	0.718446602	0.836158192	whitespace	Yes	No
TF-IDF	1	0.718446602	0.836158192	nltk	Yes	No
TF-IDF	1	0.718446602	0.836158192	nltk	No	No
TF-IDF	1	0.718446602	0.836158192	nltk	Yes	No
TF-IDF	1	0.718446602	0.836158192	nltk	No	Yes

Figure 1 – The Models Value.

3.2 Key Observations:

From the results, the following can be observed:

Stemming effect: Steaming had a relatively minor effect on the performance of TF-IDF, but it caused a significant drop in the performance of BM25. In some cases, the retrieval accuracy and relevant documents retrieved were higher when no stemming was performed.

Stopword Removal Effect: The TF and BM25 models' results improved a lot after the removal of stopwords. Thus, the presence of stopwords likely resulted in irrelevant matches and thereby decreased the retrieval accuracy. Without stopwords, the observed improvement in both precision and recall shows this as an important preprocessing step in the effectiveness of document retrieval.

BM25 Performance: Although very effective in ranking relevant documents, BM25 did not do as well as TF-IDF on the basis of precision and recall. The model was better at ranking the documents based on NDCG but struggled at retrieving highly relevant documents, since that hampered its precision.

TF-IDF Outperforms: The scores of TF-IDF were the highest among the three models, with perfect scores in some. This will say that this model is very strong concerning ranking relevant documents, which may suggest the high strength of IDF in filtering out irrelevant documents.

3.3 Discussion of Limitations

While the performances under most situations were good, there are several limitations. First of all, the specificity or relevance of the query in concern affects model performance. For instance, common terms might lead to high precision but low recall, while more specific terms may result in low precision.

Data Variability: Although diverse, the BBC News data may not reflect the complete range of document types that might arise in real IR tasks. Other document corpora may yield different results.

Performance Metrics: While NDCG is a strong ranking metric, it might not truly indicate the quality of the retrieval if highly relevant documents are not retrieved at the top ranks.

4. Conclusion

This report has presented a comprehensive comparison of TF, TF-IDF, and BM25 in the context of information retrieval from the BBC News dataset. By experimenting with different preprocessing techniques and analyzing the results across various evaluation metrics, we were able to draw meaningful conclusions about the effectiveness of each model under different conditions. TF-IDF emerged as the most robust performer, especially when paired with stopword removal. However, BM25 showed its value in ranking documents effectively, which is a critical feature for modern search engines. The study also highlights the importance of preprocessing, with stopword removal and stemming playing significant roles in improving retrieval quality.The results from this study provide a solid foundation for further research into optimizing IR systems and exploring more advanced techniques for document retrieval and ranking.

5. Project References:

1. ***Zhang, X., & Wang, X. (2018). Text Preprocessing for Text Mining and Information Retrieval. Springer.***
2. ***Croft, W. B., Metzler, D., & Strohman, T. (2009). Search Engines: Information Retrieval in Practice. Addison-Wesley.***

3. ***Course Slides***
4. ***BM25 Implementation in Python - https://github.com/dorianbrown/bm25***
5. ***Kaggle Tutorials - Text Preprocessing and Feature Extraction - https://www.kaggle.com/learn***
6. ***Python's Regular Expression (Regex) for Text Processing - https://docs.python.org/3/library/re.html***
7. ***Scikit-learn Documentation - TF-IDF Vectorizer - https://scikit-learn.org/stable/modules/feature_extraction.html#tfidf-term-weighting***