Natural Language Processing - Project

V Sai Krishna[EE17B035] and Kommineni Aditya[EE17B047]

Indian Institute of Technology Madras

Abstract. This project involves building an Information Retrieval system employing Natural Processing techniques such as Latent Semantic Analysis(LSA), Query Expansion, Bigrams and modified scoring techniques. The objective of the IR system is to return apt documents from the collection to answer the query. The dataset which has been provided is the Cranfield Dataset.

Keywords: Latent Semantic Analysis · BM25 · Bigram · TF-IDF · Query Expansion · WordNet

1 Introduction

The project primarily involves employing various Natural Language Processing techniques, **a combination of both bottom up and top down methods** in order to develop an Information Retrieval system. Henceforth, the Information Retrieval has been abbreviated as IR through the entire report.

1.1 Structure of the Report

The document hereon has been divided in the following manner. Problem Statement has been described in **Section 2**, Prior and related work in the field of IR systems has been discussed in **Section 3**, the IR model which has been treated as a baseline to compare the results of proposed models has been explained in **Section 4**, the concepts which have been employed to improve the efficacy of the baseline model in terms of evaluation metrics has been enlisted in **Section 5**, the various models which have been tested employing a mix of the concepts enlisted have been mentioned in **Section 6**, the results of the models have been tabulated in **Section 7**, the references for the entire report have been cited in **Section 8** and finally, we have an **Appendix** at the end to report the tabulated results.

2 Problem Statement

As mentioned in Introduction, the objective of the project is to built an IR system which has the ability to return the most relevant documents from the data available to us for the query.

In this case, the dataset is the **Cranfield dataset**. It is composed of **1400 documents** with a title for each of the document followed by a body composed of a string of sentences. In addition, we are provided **225 queries**. Each of the queries have along with them, in ranked order from 1 to 5 the set of most relevant documents among the dataset.

Employing the queries and the corresponding rankings of documents, we can evaluate a model which we design using evaluation metrics. Here, the evaluation metrics which have been employed include **Precision**, **Recall**, **Mean Average Precision**, **nDCG** and **F-score**. Each of the mentioned evaluation metrics were measured for the first 10 positions of the retrieved docuemnts for the given set of queries.

3 Related Work

From literature survey on information retrieval system models, we observed the following:

- There were numerous instances wherein models were employed to utilize **Wordnet and query augmentation** as a tool to improve information retrieval capacity of the system such as in [3], [6], [4]. However, they didn't report a significant improvement by employing Wordnet.
- In terms of Latent Semantic Analysis, **SVD** has been extensively employed to improve vector space based models through **rank lowering** such as in [1].
- We know that **unigrams are unable to capture the context of the sentence**. Hence, attempts have been made to employ **bigrams** in vector space models so as to enable some level of order to sentences as seen in [2].
- In contrast to the above methods wherein the focus is on improving the formalisms, another form of improving the retrieval systems was to **tweak the scoring methods to use metrics other than cosine similarity** which has been depicted in [7]. These scoring metrics were often employed in **early web browsers for search result ranking**.

4 Prior Work - Baseline System

During the course, we were instructed to implement an Information Retrieval system employing the vector space model ideology.

The pre processing of the dataset involved the following steps. Firstly, the documents in the dataset were word **tokenized**. The individual words were then **lemmatized** to their root form. Following this, we get **rid of the stop words** using a list of stopwords in the NLTK package. Once we do this, we have a set of words in each document in their root forms.

Following this, we initialize an array with the dimensions of the number of types in the dataset as the columns and number of documents as the rows. This makes the **term document matrix**. Then, we calculate the **Inverse Document Frequency(IDF) matrix**. This enables us calculate the TF-IDF matrix. We perform a similar operation on the query and calculate the corresponding query vector.

Now, we can calculate the **cosine similarity** between each of the document vector and the corresponding query vector. Following which we **order them in the descending order** and we can calculate the evaluation metrics for the entire query set.

The TF-IDF method in the vector space models produced the following results as depicted in the Fig. 1 and we will be using this system as the baseline system for further comparing the performance to the models we have trained.

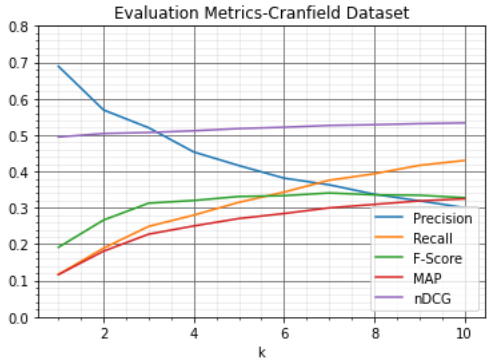

Fig. 1: A figure showing the performance of the baseline model.

5 Proposed Methodology

We propose the following in order to obtain a better information retrieval model over the base model in terms of efficacy.

5.1 Latent Semantic Analysis

Latent Semantic Analysis(LSA) is a statistical Natural Language Processing approach which involves **building similarities in documents through process of rank lowering**. This process of rank lowering is facilitated through Singular Value Decomposition(SVD) as discussed in [1]. Firstly, the term document matrix is computed following which this matrix undergoes SVD.

Let X be the term document matrix. The SVD operation is defined as follows :

$$X = U\Sigma V^T \tag{1}$$

In the above equation, U and V are orthogonal matrices representing the left and right singular vectors respectively whereas Σ is a diagonal matrix composed of singular values. Now, we exclude a few singular values which are small in values along with their corresponding singular vectors. This will provide us with a lower rank approximation of the term document matrix. Following this, we can use this lower rank matrix to compute retrieve documents using cosine similarity.

5.2 Query Expansion using WordNet

In contrast to the above section where we leveraged the information present within the data provided, in this section we employ an external source of information to gain insights and improve our IR model. We will be using **the semantic relatedness between words in WordNet to include more words/terms in our query**. This process is known as query expansion and it will help in context disambiguation and well as including synonyms which are present in the documents but not in the query.

5.3 Okapi BM25

Thus far, we have relied on cosine similarities as the scoring function for finding the similarities between query and document vectors. **However, Okapi BM25 is a modified ranking function which could be used similar to the cosine similarity**. The formula for the similarity score is as below:

$$Score(D,Q) = \Sigma_{i=1}^{N} IDF(q_i).\frac{f(q_i,D).(k_1+1)}{f(q_i,D)+k_1.(1-b+b\frac{|D|}{avgdl})} \tag{2}$$

In the equation above, N refers to the number of keywords in query Q. IDF refers to the inverse document frequency, $f(q_i, D)$ refers to the number of times keyword q_i occurs in document D and avgdl refers to the average document length in the dataset. k_1 and b are hyperparameters which can be learnt while optimizing or be tuned to obtain the best results on a validation set. k_1 is a measure of "when do we think a term is likely to be saturated?" and b is a measure of "when do we think a document is likely to be very long, and when should that hinder its relevance to a term?" [5]

We replace the cosine similarity score with the scoring formula seen above and employ it on our processed documents.

5.4 Bigrams

For the baseline model, we have employed using unigrams with the stop words removed for the TF-IDF matrices. However, we can clearly see that **unigrams**

do not capture any sense of word order in the sentences. Therefore, we propose the use of bigrams instead of use of unigrams alone.

However, we need to note the fact that in order to prevent the size of the term document matrix of bigrams from exploding, we will consider only the most frequent bigrams in the term frequency matrix.

6 Experiments

6.1 Experiment 1

In this Experiment the following IR systems are compared (Refer to section 7.1 for the graphical results and the Appendix at the end of the document for tabular results):

- **TF-IDF**: This is our Baseline model. No additions/variations. The tabular results are in Appendix A.
- **TF-IDF+LSA**: In this IR system, we will incorporate LSA into our baseline model. The TF-IDF matrix from our Baseline model will first undergo SVD. After this smaller Eigenvalues will be omitted such that 80% of the variance is preserved. The tabular results are in Appendix B.
- **TF-IDF+Wordnet**: In this IR system, we shall use wordnet for query expansion and follow it up with TF-IDF. The tabular results are in Appendix C.
- **TF-IDF+Bigram**: In this IR system, we shall first create a list of common Bigrams. (The top **15000 Bigram words** are considered). Then similarities are measured using both Unigram and Bigram model, and a weighted average is taken (60% for Unigrams and 40% for Bigrams). The tabular results are as shown in Appendix D
- **BM25**: This model doesn't modify our baseline model like the above systems. Instead it uses the Okapi BM25 scoring metric. The hyper-parameters used are: b=1.2 and k_1=0.75. The tabular results are in Appendix E.

6.2 Experiment 2

In this Experiment the following IR systems are compared (Refer to section 7.2 for graphical results and the Appendix for tabular results):

- **BM25**: Same BM25 model used in Experiment 1. The tabular results are in Appendix E.
- **BM25+LSA**: In this IR system, we will incorporate LSA into the BM25 model, specifically modifying the F-Matrix (Frequency Matrix) using SVD. Smaller Eigenvalues will be omitted during reconstruction such that 80% of the variance is preserved. The hyper-parameters used are: b=1.2 and k_1=0.75. The tabular results are in Appendix F.

- **BM25+Wordnet**: In this IR system, we shall use wordnet for query expansion and follow it up with BM25. The hyper-parameters used are: b=1.2 and k_1=0.75. The tabular results are in Appendix G.
- **BM+Bigram**: In this IR system, we shall first create a list of common Bigrams (The top **15000 Bigram words** are considered). Then similarities are measured using both Unigram and Bigram model along with the BM25 metric, after which a weighted average is taken (60% for Unigrams and 40% for Bigrams). The hyper-parameters used are: b=1.2 and k_1=0.75. The tabular results are in Appendix H.

6.3 Experiment 3

In this Experiment the following IR systems are compared (Refer to section 7.3 for graphical results and Appendix for tabular results):

- **BM25**: Same BM25 model used in Experiment 1. The tabular results are as listed in Appendix E.
- **BM25+LSA**: In this IR system, we will incorporate LSA into the BM25 model, specifically modifying the F-Matrix (Frequency Matrix) using SVD. Smaller Eigenvalues will be omitted during reconstruction such that 80% of the variance is preserved. The hyper-parameters used are: b=1.2 and k_1=0.75. The tabular results are in Appendix F.
- **BM+Bigram**: In this IR system, we shall first create a list of common Bigrams.(The top **15000 Bigram words** are considered). Then similarities are measured using both Unigram and Bigram model along with the BM25 metric, after which a weighted average is taken (60% for Unigrams and 40% for Bigrams). The hyper-parameters used are: b=1.2 and k_1=0.75. The tabular results are in Appendix H.
- **BM+Bigram+LSA**: We shall incorporate LSA into the Unigram Frequency matrix nd not the Bigram Frequency matrix (as Bigrams aren't significantly related to each other). Then similarities are measured and combined using a weighted average (60% for Unigrams and 40% for Bigrams). The hyper-parameters used are: b=1.2 and k_1=0.75. The tabular results are in Appendix I.

7 Results

The following subsections are composed of the evaluation metric plots for the experiments mentioned in the section above. In order to refer to exact values for each model, refer to the Appendix section tables.

7.1 Experiment 1

The results of Experiment 1 is shown graphically in Figure 2. One can observe that **BM25 produces better result over all variations of TF-IDF**. Among the TF-IDF variations, including Bigrams and LSA into our system seems to improve the performance. However, using Wordnet has led to poorer results than the baseline model.

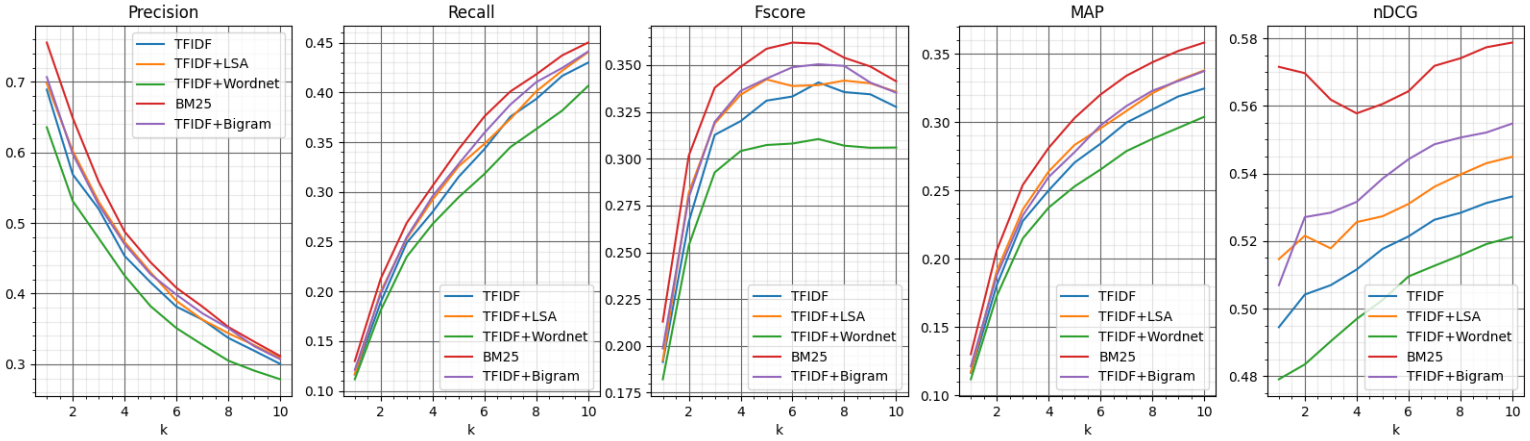

Fig. 2: A plot of the various evaluation metrics for the models as mentioned in the legend of the plot for Experiment 1

7.2 Experiment 2

The results of Experiment 2 is shown graphically in Figure 3. One can observe that **BM25+Bigram produces the best result**. BM25+LSA doesn't improve the performance much. However, using Wordnet once again has led to poorer results.

7.3 Experiment 3

Since BM25 always performs better than TF-IDF and Wordnet always leads to poor performance, we have compared all variations of BM25 excluding wordnet in Experiment 3 (Refer Figure 4). As you can see, **BM25+Bigram+LSA, i.e the combination of all three models performs the best**. BM25+Bigram comes second, followed by BM25+LSA and BM25 (The last two perform almost the same).

8 Conclusion

Firstly, let us enlist the drawbacks of our baseline model which will enable us better understand the reasoning for employing the methods used in the experiments above. The drawbacks of the baseline model are as below.

– Being a bag of words unigram model, **the baseline model doesn't take into account context and order** in which the words of the query or document occur.

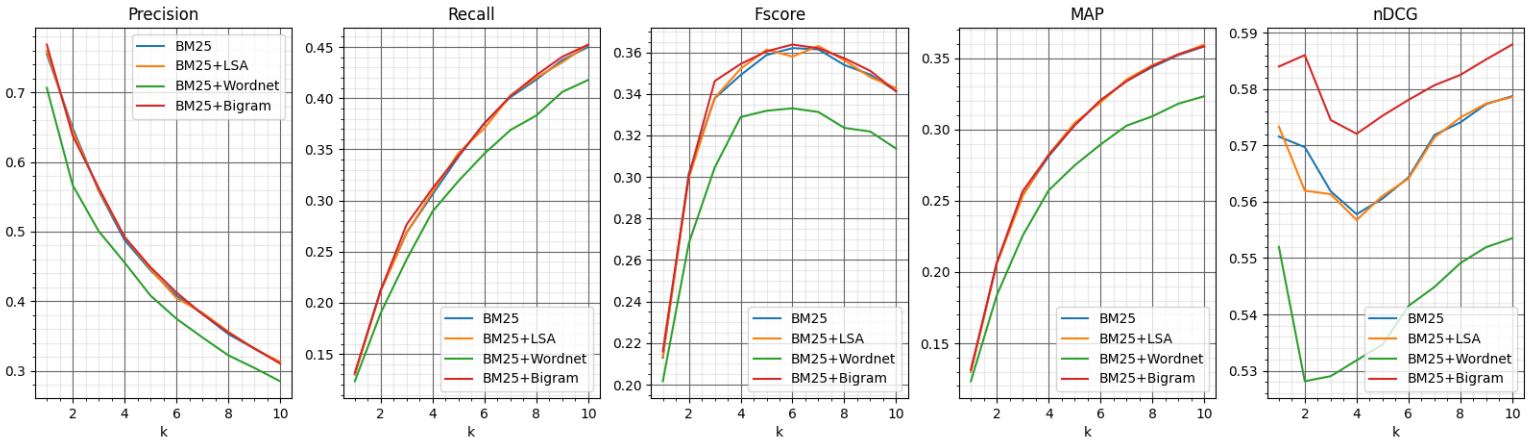

Fig. 3: A plot of the various evaluation metrics for the models as mentioned in the legend of the plot for Experiment 2

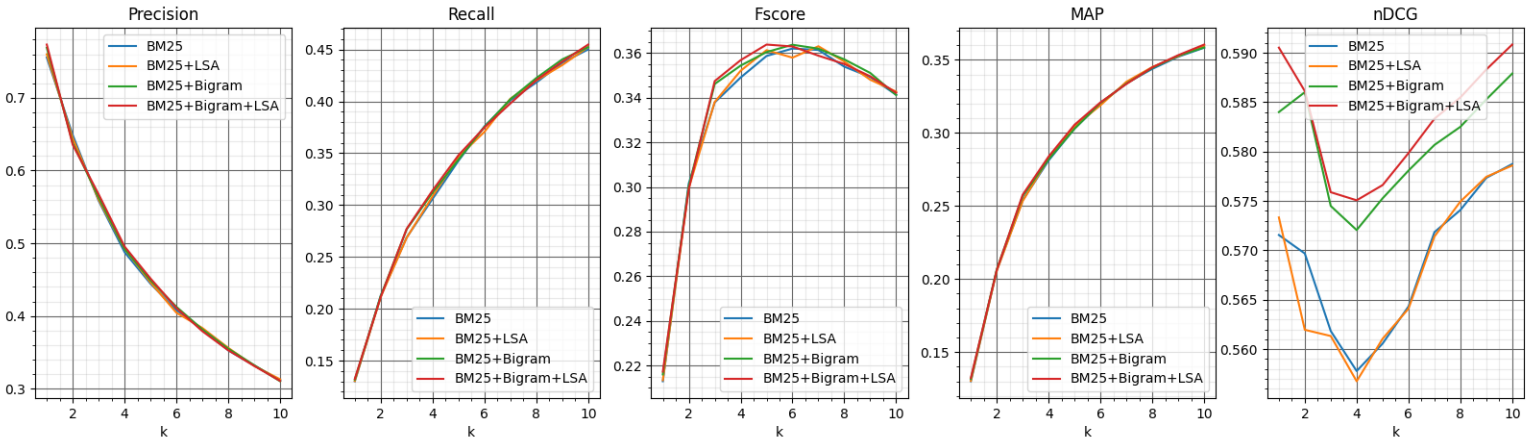

Fig. 4: A plot of the various evaluation metrics for the models as mentioned in the legend of the plot for Experiment 3

- While using the vector model in term frequency matrix, we **do not have an innate method by which we could account for synonyms** of a word.
- As a scoring metric, we employ cosine similarity in baseline model. However, considering that the query is often very small in size compared to the documents, we may not always get a very representative estimate of the similarity between documents and query since in the query often a single word may have a very large emphasis which could dictate the meaning of the query.

In order to account for the above mentioned drawbacks, we have proposed an array of methods. The manner in which each of the methods alleviate one or more of the above concerns are as follows:

- The use of LSA and Wordnet is an effort to account for similarities between words and make sure we account for synonyms of a word in the term frequency matrix when the word occurs in the document or query. We observe that LSA improves the efficacy of the model as depicted in Figure.2. This could be attributed to the fact that **LSA is able to identify underlying concepts** in the dataset. However, we see that employing Wordnet doesnt account for any increase in performance. This could be owing to the fact that the dataset is from a restricted domain i.e. mostly scientific. Owing to this, the queries have quite a few words which have proper nouns which arent classified in wordnet. Moreover, being a single domain, **most words are used in a single sense alone** here thereby reducing the need for word sense disambiguation.
- **Using bigrams along with unigrams in bag of words model helps us establish some degree of context and order** in the words. We observe that employing bigrams improves the evaluation metrics over the baseline models as shown in Figure.2. However, moving any further over bigrams to use trigrams doesn't result in an improvement. This is understandable owing to the sparsity of the trigram matrix.
- Using the BM25 scoring metric as opposed to cosine similarity used in baseline model gives an observable improvement in efficacy of the model as shown in Figure.3. This could be owing to the fact that the **BM25 scoring metric iterates over every keyword of the query and normalizes the importance** of the specific document in terms of length of document against the average document length.

From the experiments, we can see that in individual models, BM25 method performs the best and when combined together, effects of LSA, Bigrams and BM25 add up to give the best results on evaluation metrics. Although the best method (BM25+LSA+Bigram) performs better across all metrics, it is more pronounced in nDCG, as it is the only Evaluation metric in this report that takes the ranking of documents into account.

References

1. Deerwester, S., Dumais, S.T., Furnas, G.W., Landauer, T.K., Harshman, R.: Indexing by latent semantic analysis. JOURNAL OF THE AMERICAN SOCIETY FOR

INFORMATION SCIENCE **41**(6), 391–407 (1990)
2. Jiang, M., Jensen, E., Beitzel, S., Argamon, S.: Choosing the right bigrams for information retrieval. In: Classification, Clustering, and Data Mining Applications, pp. 531–540. Springer (2004)
3. Mandala, R., Tokunaga, T., Tanaka, H.: The use of wordnet in information retrieval. In: Usage of WordNet in Natural Language Processing Systems (1998)
4. Smeaton, A.F., Berrut, C.: Running trec-4 experiments: A chronological report of query expansion experiments carried out as part of trec-4. In: Proceedings of The Fourth Text REtrieval Conference (TREC-4) (1995)
5. Trotman, A., Puurula, A., Burgess, B.: Improvements to bm25 and language models examined. In: Proceedings of the 2014 Australasian Document Computing Symposium. pp. 58–65 (2014)
6. Voorhees, E.M.: Query expansion using lexical-semantic relations. In: SIGIR'94. pp. 61–69. Springer (1994)
7. Whissell, J.S., Clarke, C.L.: Improving document clustering using okapi bm25 feature weighting. Information retrieval **14**(5), 466–487 (2011)

APPENDIX

A Baseline Model

Table corresponding to the results for baseline model is as below:

Index	Precision	Recall	F-Score	MAP	nDCG
1	0.69	0.12	0.19	0.12	0.49
2	0.57	0.19	0.27	0.18	0.5
3	0.52	0.25	0.31	0.23	0.51
4	0.45	0.28	0.32	0.25	0.51
5	0.42	0.32	0.33	0.27	0.52
6	0.38	0.34	0.33	0.28	0.52
7	0.36	0.38	0.34	0.3	0.53
8	0.34	0.39	0.34	0.31	0.53
9	0.32	0.42	0.33	0.32	0.53
10	0.3	0.43	0.33	0.32	0.53

Table 1: Table comprising of the evaluation metrics for TF-IDF model

B TFIDF & LSA

Table corresponding to the results for model combining TF-IDF and LSA is as below:

Index	Precision	Recall	F-Score	MAP	nDCG
1	0.7	0.12	0.19	0.12	0.51
2	0.6	0.2	0.28	0.19	0.52
3	0.53	0.25	0.32	0.24	0.52
4	0.47	0.29	0.33	0.26	0.53
5	0.43	0.33	0.34	0.28	0.53
6	0.39	0.35	0.34	0.3	0.53
7	0.36	0.37	0.34	0.31	0.54
8	0.34	0.4	0.34	0.32	0.54
9	0.33	0.42	0.34	0.33	0.54
10	0.31	0.44	0.34	0.34	0.54

Table 2: Table comprising of the evaluation metrics for TF-IDF combined with LSA model

C TFIDF & Wordnet

Table corresponding to the results for model combining TF-IDF and Wordnet for query augmentation is as below:

Index	Precision	Recall	F-Score	MAP	nDCG
1	0.64	0.11	0.18	0.11	0.48
2	0.53	0.18	0.25	0.17	0.48
3	0.48	0.23	0.29	0.21	0.49
4	0.43	0.27	0.3	0.24	0.5
5	0.38	0.29	0.31	0.25	0.5
6	0.35	0.32	0.31	0.27	0.51
7	0.33	0.35	0.31	0.28	0.51
8	0.3	0.36	0.31	0.29	0.52
9	0.29	0.38	0.31	0.3	0.52
10	0.28	0.41	0.31	0.3	0.52

Table 3: Table comprising of the evaluation metrics for TF-IDF combined with Wordnet model

D TFIDF & Bigrams

Table corresponding to the results for model combining TF-IDF and Bigrams is as below:

Index	Precision	Recall	F-Score	MAP	nDCG
1	0.71	0.12	0.2	0.12	0.51
2	0.6	0.2	0.28	0.19	0.53
3	0.53	0.25	0.32	0.23	0.53
4	0.47	0.3	0.34	0.26	0.53
5	0.43	0.33	0.34	0.28	0.54
6	0.4	0.36	0.35	0.3	0.54
7	0.37	0.39	0.35	0.31	0.55
8	0.35	0.41	0.35	0.32	0.55
9	0.33	0.42	0.34	0.33	0.55
10	0.31	0.44	0.34	0.34	0.55

Table 4: Table comprising of the evaluation metrics for TF-IDF combined with Bigram model

E Okapi BM25

Table corresponding to the results for model using BM25 as the scoring metric and baseline TF-IDF matrix is as below:

Index	Precision	Recall	F-Score	MAP	nDCG
1	0.76	0.13	0.21	0.13	0.57
2	0.65	0.21	0.3	0.21	0.57
3	0.56	0.27	0.34	0.25	0.56
4	0.49	0.31	0.35	0.28	0.56
5	0.44	0.34	0.36	0.3	0.56
6	0.41	0.38	0.36	0.32	0.56
7	0.38	0.4	0.36	0.33	0.57
8	0.35	0.42	0.35	0.34	0.57
9	0.33	0.44	0.35	0.35	0.58
10	0.31	0.45	0.34	0.36	0.58

Table 5: Table comprising of the evaluation metrics for BM25 model

F Okapi BM25 & LSA

Table corresponding to the results for model employing BM25 as the scoring metric and TF-IDF matrix along with LSA is as below:

Index	Precision	Recall	F-Score	MAP	nDCG
1	0.76	0.13	0.21	0.13	0.57
2	0.64	0.21	0.3	0.21	0.56
3	0.56	0.27	0.34	0.25	0.56
4	0.49	0.31	0.35	0.28	0.56
5	0.45	0.35	0.36	0.3	0.56
6	0.4	0.37	0.36	0.32	0.56
7	0.38	0.4	0.36	0.34	0.57
8	0.36	0.42	0.36	0.35	0.57
9	0.33	0.44	0.35	0.35	0.58
10	0.31	0.45	0.34	0.36	0.58

Table 6: Table comprised of evaluation metrics for BM25 scoring metric combined with LSA

G Okapi BM25 & Wordnet

Table corresponding to the results for model employing BM25 as the scoring metric and TF-IDF matrix along with Wordnet for query augmentation is as below:

Index	Precision	Recall	F-Score	MAP	nDCG
1	0.71	0.12	0.2	0.12	0.55
2	0.57	0.19	0.27	0.18	0.53
3	0.5	0.24	0.3	0.23	0.53
4	0.46	0.29	0.33	0.26	0.53
5	0.41	0.32	0.33	0.27	0.53
6	0.37	0.35	0.33	0.29	0.54
7	0.35	0.37	0.33	0.3	0.54
8	0.32	0.38	0.32	0.31	0.55
9	0.3	0.41	0.32	0.32	0.55
10	0.28	0.42	0.31	0.32	0.55

Table 7: Table comprised of evaluation metrics for BM25 scoring metric combined with Wordnet

H Okapi BM25 & Bigram

Table corresponding to the results for model employing BM25 as the scoring metric and TF-IDF matrix along with Bigrams is as below:

Index	Precision	Recall	F-Score	MAP	nDCG
1	0.77	0.13	0.22	0.13	0.58
2	0.64	0.21	0.3	0.21	0.59
3	0.56	0.28	0.35	0.26	0.57
4	0.49	0.31	0.35	0.28	0.57
5	0.45	0.34	0.36	0.3	0.58
6	0.41	0.38	0.36	0.32	0.58
7	0.38	0.4	0.36	0.33	0.58
8	0.36	0.42	0.36	0.34	0.58
9	0.33	0.44	0.35	0.35	0.59
10	0.31	0.45	0.34	0.36	0.59

Table 8: Table comprised of evaluation metrics for BM25 scoring metric combined with Bigrams

I Okapi BM25 & Bigram & LSA

Table corresponding to the results for model employing BM25 as the scoring metric and TF-IDF matrix along with LSA and Bigrams is as below:

Index	Precision	Recall	F-Score	MAP	nDCG
1	0.77	0.13	0.22	0.13	0.59
2	0.64	0.21	0.3	0.21	0.59
3	0.57	0.28	0.35	0.26	0.58
4	0.5	0.31	0.36	0.28	0.58
5	0.45	0.35	0.36	0.31	0.58
6	0.41	0.37	0.36	0.32	0.58
7	0.38	0.4	0.36	0.33	0.58
8	0.35	0.42	0.36	0.34	0.59
9	0.33	0.44	0.35	0.35	0.59
10	0.31	0.45	0.34	0.36	0.59

Table 9: Table comprised of evaluation metrics for BM25 scoring metric combined with LSA and Bigrams