From Vectors to Transformers: Decoding Search Engine Performance

Akshat Meena[1†] and Siddharth Singh[1†]

[1]Department of Computer Science and Engineering, IIT Madras, Chennai, Tamil Nadu, India.

Contributing authors: cs19b052@smail.iitm.ac.in; cs19b072@smail.iitm.ac.in;
[†]These authors contributed equally to this work.

Abstract

We conduct a comparative analysis of search engine models, specifically the Vector Space Model (VSM), Latent Semantic Indexing (LSI), and transformer-based models (BERT), on the Cranfield dataset. The performance of the models is evaluated by experimenting with different weighting schemes, including counts, term frequency-inverse document frequency (TF-IDF), normalized TF-IDF, and the Glasgow model. Through an exploration of various hyperparameter values, we aim to optimize the retrieval capabilities of each model. The results obtained from the analysis provide insights into the strengths and weaknesses of the models, aiding in the design and implementation of effective search systems. The study contributes to the field of information retrieval by offering empirical evidence and valuable findings based on the analysis of the Cranfield dataset.

Keywords: Vector Space Model, Latent Semantic Indexing, TF-IDF, Glasgow model

1 Introduction

Search engines play a crucial role in information retrieval, aiding users in finding relevant and meaningful results from vast amounts of data. In this technical report, we present a comprehensive study that compares the performance of different search engine models, namely the Vector Space Model (VSM), Latent Semantic Indexing (LSI), and a transformer-based model. Through experimental analysis and rigorous statistical hypothesis testing, we aim to gain insights into the strengths and weaknesses of each model, shedding light on their effectiveness in delivering accurate and relevant search results.

To evaluate the performance of these models, we employed various weighting schemes, including basic counts, TF-IDF, normalized TF-IDF, and the Glasgow model. By experimenting with different hyperparameter values, we sought to optimize the models' retrieval capabilities and assess their overall effectiveness. The theoretical foundations of our hypotheses were derived from the literature on information retrieval, text mining, and natural language processing.

The Vector Space Model (VSM) is a widely used method that represents documents and queries as vectors in a high-dimensional space. It calculates the similarity between documents and queries based on measures such as cosine similarity. Latent Semantic Indexing (LSI), on the other hand, incorporates a dimensionality reduction technique to capture latent semantic relationships between terms and documents. By leveraging the Singular Value Decomposition (SVD) algorithm, LSI aims to uncover hidden patterns and meaning within the data. Lastly, our study also incorporates a transformer-based model (based on BERT, to be more specific), which has gained prominence in recent years due to its ability to capture contextual relationships and dependencies between words, enhancing the quality of search results.

In our investigation, we conducted an experimental analysis to compare the performance of these models using the Cranfield datasets. The Cranfield dataset is a well-established benchmark dataset

widely used in the field of information retrieval research. It comprises a diverse collection of documents from various disciplines, making it suitable for evaluating search engine models across different domains.

To validate our hypotheses, we employed rigorous statistical hypothesis tests, which allowed us to make informed conclusions about the significance of the observed differences in performance between the models. By establishing statistical significance, we gained confidence in the reliability and generalizability of our findings.

The remainder of this report is organized as follows: In the next section, we provide an overview of the related literature and discuss the theoretical foundations underlying our study. Subsequently, we describe the experimental setup, including the datasets, preprocessing techniques, and implementation details of the search engine models. We present the results obtained from the experiments, discuss the performance metrics and analyse the outcomes based on our hypotheses. Finally, we draw conclusions, highlight the implications of our findings, and suggest avenues for future research.

Overall, this study aims to contribute to the field of information retrieval by providing insights into the comparative performance of different search engine models, helping researchers and practitioners make informed decisions when designing and implementing effective search systems.

2 Models

2.1 Vector Space Model

The Vector Space Model (VSM) [1] is a basic but popular search engine model. It represents documents and queries as vectors in a high-dimensional space. It then makes use of similarity calculations between vectors to find the documents relevant to a given query.

Each term in the corpus is a dimension in the space where the documents and the queries are projected. The coefficient of each dimension represents the weight or importance of the term in that particular document or query. Various weighting schemes can be employed to assign weights to terms, such as basic counts, term frequency-inverse document frequency (TF-IDF), normalized term frequency-inverse document frequency, and the Glasgow Model.

The similarity measure mostly used in VSM is cosine similarity between the vector representation of documents and queries. The documents are then retrieved in decreasing order of their cosine similarity with the query vector. Cosine similarity gives a value in the range [0,1]. 0 means no similarity while 1 means perfect similarity.

2.1.1 Advantages

VSM is a simple model to implement as compared to the other modern methods. VSM also gives sufficiently good results when the query is "similar" to the documents. By "similar", we mean that most of the terms present in the query are also present in the documents. Also, the performance of the VSM model can be improved by using sophisticated weighting schemes.

2.1.2 Limitations

There are certain limitations of this model also. One of them is that it doesn't take into account the order in which the words come together to form a sentence. It relied on the "bag-of-words" representation of the queries and documents due to which it is not able to extract the semantic meaning of the text accurately. Also, VSM is very computation-intensive as the space in which the documents and queries are embedded is generally very high-dimensional. All the vectors have to be recomputed once a new term is added to the term space.

2.2 Latent Sematic Indexing

Latent Semantic Indexing [2] is a model used in information retrieval. Its underlying motive is to extract certain "hidden" concepts from the corpus that relate the terms to the documents in which they appear using matrix factorization. In contrast to the Vector Space Model (VSM), where documents and queries are mapped to a high-dimensional space, the Latent Semantic Indexing (LSI) approach involves projecting documents and queries onto a lower-dimensional space.

A term-document matrix is constructed where all the terms present in the corpus form the rows of the matrix, and all the documents form the columns of the matrix. The entries of the matrix signify the relevance of that term to the document. Singular Value Decomposition (SVD) is used to decompose the matrix into three matrices - a term-concept matrix, a singular value matrix, and

the concept-document matrix. The number of concepts to be extracted is a hyperparameter, and its value is to be decided by the user.

The term-concept matrix represents the relationship between terms and "latent" or "hidden" concepts. Each row in the matrix corresponds to a term, and each column represents a latent concept. The entries indicate the strength of association between the term and the concept. Similarly, the concept-document matrix captures the relationship between concepts and documents. The columns correspond to the concepts, while the rows represent the documents, and the entries reflect the strength of the association of the document with the concept.

2.2.1 Advantages

An advantage of the LSI is that it does not require any source of external knowledge. It is an approach that extracts semantic relations between the terms and their usage in documents using computational techniques.

It is not as naive as VSM when the order in which the words appear in a sentence is considered. It is also good at handling synonymy, which means it can capture the relation between two different words that convey the same meaning.

As discussed earlier, the vector space in which the queries and documents are mapped is of lower dimensionality as compared to VSM. Hence, constructing the vectors and calculating the cosine similarities is less computation-intensive. However, performing the SVD is a computationally intensive task itself.

2.2.2 Limitations

LSI also has certain limitations. One is that if we choose the number of hidden dimensions to be too low, then the precision gets affected, whereas if we choose the number of hidden dimensions to be too high, then the recall gets affected. So, the user must choose an appropriate value for the number of hidden dimensions to suit his requirements.

We discussed above that LSI is good at handling synonymy, but it is inefficient at handling polysemy.

2.3 Transformers

Transformer-based models have revolutionized the field of natural language processing (NLP) and information retrieval due to their ability to capture complex dependencies and contextual relationships within textual data.

The transformer architecture [3] is based on the concept of self-attention mechanisms. Recurrent neural networks (RNNs) and Convolutional neural networks (CNNs) perform sequential processing but it is not the case with transformers. They capture global dependencies and contextual information efficiently by attending to all positions or words within a sequence simultaneously.

In this model, input sequences are represented as embeddings, which are then transformed by multiple self-attention layers and feed-forward neural networks. The self-attention mechanism allows each word to attend to all other words in the sequence, capturing their contextual importance. This enables the model to learn intricate relationships between words and improve its understanding of the semantic meaning and structure of the text.

After the encodings of the documents are calculated, they are ranked on the basis of the cosine similarity with the query's encoding.

We have used *multi-qa-MiniLM-L6-cos-v1* model from the Hugging Face library for our experiments. It is based on the MiniLM-L6 architecture, which is a variant of BERT (Bidirectional Encoder Representations from Transformers). BERT is a popular transformer-based model that has achieved state-of-the-art performance on various natural language processing tasks, including question-answering.

2.3.1 Advantages

One of the primary advantages of transformer-based models is that they are able to handle long-range dependencies and capture contextual information effectively. This becomes very important in the field of information retrieval, where understanding the context of a query and document is very important to find the most relevant documents. Transformers have beaten the state-of-the-art models in various NLP tasks, such as machine translation, text summarization, and question-answering.

2.3.2 Limitations

Transformer-based models also have their own limitations. They require substantial computational resources for training due to their large number of parameters. Fine-tuning transformer models on domain-specific data can be time-consuming and may necessitate a significant amount of annotated data. Additionally, transformers are sensitive to the quality and diversity of the training data, which can affect their generalization capabilities.

3 Evaluation

We analyzed the VSM and LSI models with different weighting schemes, namely simple counts, term frequency-inverse document frequency (TF-IDF), normalized term frequency-inverse document frequency (TF-IDF) and the Glasgow model.

We experimented with the LSI model with different values of the hyperparameter, which in this case was the number of hidden dimensions.

Note that in both the models, the preprocessing was done on the queries as well as the documents. Segmentation, tokenisation, stemming, and stopword removal were performed in order. Then, the preprocessed queries and documents were passed to the models for information retrieval.

We ran the multi-qa-MiniLM-L6-cos-v1 model on the documents and queries to find their encodings. After this, cosine similarity was used as a similarity measure to rank the documents in order of their relevance to a given query.

In the case of the *multi-qa-MiniLM-L6-cos-v1* model, we did not preprocess the queries or the documents.

We calculated different evaluation metrics like precision, recall, f-score, Mean Average Precision (MAP), and normalized Discounted Cumulative Gain (nDCG) for each model at different ranks.

We compared the values of these evaluation metrics for different models and weighting schemes and made some observations based on them.

We then gave sound theoretical arguments backing our observations and then finally, performed statistical hypothesis testing to verify them.

4 Observations

4.1 Evaluation Results

4.1.1 Vector Space Model (VSM)

k	Precision	Recall	F-score	MAP	nDCG
1	0.5644	0.0938	0.1544	0.0938	0.4544
2	0.4578	0.1494	0.2124	0.1401	0.4136
3	0.3837	0.1854	0.2336	0.1648	0.3815
4	0.3511	0.2175	0.2495	0.1841	0.3787
5	0.3191	0.2435	0.2556	0.1978	0.3735
6	0.2904	0.2607	0.2535	0.2072	0.3697
7	0.2705	0.2787	0.2534	0.2156	0.3703
8	0.2522	0.2941	0.2505	0.222	0.3736
9	0.2375	0.3103	0.2484	0.2275	0.3777
10	0.2236	0.3235	0.2445	0.2313	0.3795

(a) Basic Count as weighting scheme

k	Precision	Recall	F-score	MAP	nDCG
1	0.5644	0.0938	0.1544	0.0938	0.4544
2	0.4578	0.1494	0.2124	0.1401	0.4136
3	0.3837	0.1854	0.2336	0.1648	0.3815
4	0.3511	0.2175	0.2495	0.1841	0.3787
5	0.3191	0.2435	0.2556	0.1978	0.3735
6	0.2904	0.2607	0.2535	0.2072	0.3697
7	0.2705	0.2787	0.2534	0.2156	0.3703
8	0.2522	0.2941	0.2505	0.222	0.3736
9	0.2375	0.3103	0.2484	0.2275	0.3777
10	0.2236	0.3235	0.2445	0.2313	0.3795

(b) TF-IDF as weighting scheme

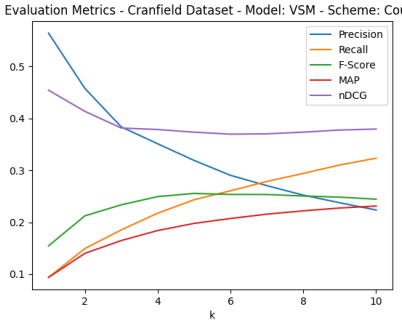
k	Precision	Recall	F-score	MAP	nDCG
1	0.6489	0.1091	0.1799	0.1091	0.52
2	0.5578	0.1839	0.2601	0.1756	0.4881
3	0.4859	0.2296	0.2901	0.2111	0.4592
4	0.4378	0.2662	0.3064	0.2365	0.4476
5	0.3884	0.2918	0.3079	0.2514	0.4378
6	0.3622	0.3268	0.317	0.2678	0.442
7	0.3378	0.3534	0.3189	0.2798	0.4454
8	0.3217	0.38	0.3219	0.2926	0.4546
9	0.3042	0.3998	0.3196	0.301	0.4603
10	0.2871	0.4156	0.3143	0.3068	0.4639

(c) Normalized TF-IDF as weighting scheme

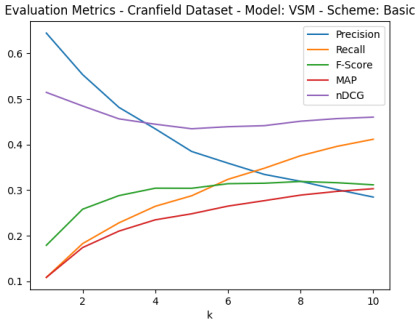
k	Precision	Recall	F-score	MAP	nDCG
1	0.6444	0.1088	0.1792	0.1088	0.5144
2	0.5533	0.1829	0.2583	0.1743	0.4846
3	0.4815	0.2284	0.2881	0.2103	0.4564
4	0.4344	0.2647	0.3043	0.235	0.4446
5	0.3849	0.2877	0.3042	0.2482	0.4347
6	0.3593	0.3238	0.3142	0.2651	0.4392
7	0.3346	0.3481	0.3153	0.277	0.4414
8	0.3194	0.3756	0.3191	0.2892	0.4512
9	0.3012	0.396	0.3164	0.2974	0.457
10	0.2849	0.4117	0.3118	0.3032	0.4602

(d) Glasgow model as weighting scheme

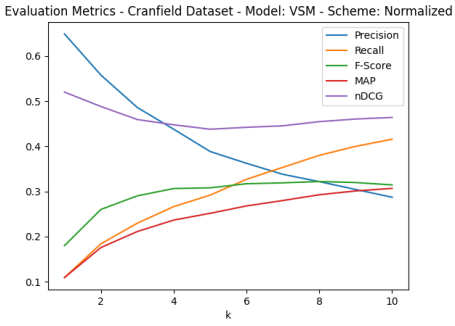
Table 1: Evaluation results for VSM model



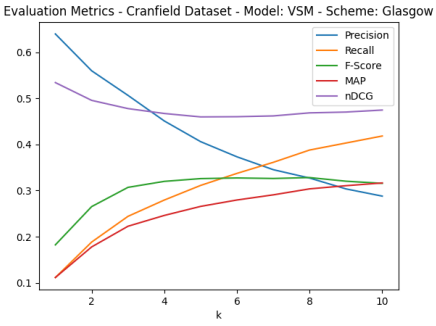
(a) Basic Count



(b) TF-IDF



(c) Normalized TF-IDF



(d) Glasgow model

Fig. 1: VSM Evaluation plots

4.1.2 Latent Sementic Indexinig (LSI)

k	Precision	Recall	F-score	MAP	nDCG
1	0.0222	0.0038	0.0064	0.0038	0.0122
2	0.0178	0.0053	0.0079	0.0046	0.0128
3	0.0178	0.008	0.0107	0.0056	0.014
4	0.0167	0.0104	0.0121	0.0062	0.0145
5	0.0169	0.0126	0.0137	0.0067	0.0153
6	0.0163	0.0155	0.0151	0.0071	0.0164
7	0.0165	0.0178	0.0162	0.0075	0.0171
8	0.0167	0.0213	0.0175	0.0079	0.0184
9	0.0158	0.0217	0.0169	0.008	0.0187
10	0.0147	0.0224	0.0164	0.008	0.0188

(a) Basic Count as weighting scheme(k=100)

k	Precision	Recall	F-score	MAP	nDCG
1	0.1689	0.0246	0.0416	0.0246	0.0978
2	0.1444	0.0396	0.0594	0.0345	0.0934
3	0.1437	0.0597	0.0798	0.0433	0.0994
4	0.1467	0.0798	0.0973	0.0511	0.106
5	0.1378	0.094	0.1052	0.0554	0.1079
6	0.1341	0.1092	0.1132	0.0603	0.1138
7	0.1257	0.1202	0.1153	0.064	0.1159
8	0.1211	0.1315	0.1183	0.0675	0.1192
9	0.1185	0.142	0.1216	0.0714	0.1235
10	0.1129	0.1472	0.12	0.0732	0.1256

(b) TF-IDF as weighting scheme(k=200)

k	Precision	Recall	F-score	MAP	nDCG
1	0.1778	0.026	0.0439	0.026	0.1122
2	0.1689	0.0472	0.0706	0.0384	0.1125
3	0.1615	0.0696	0.092	0.0493	0.1146
4	0.1489	0.0839	0.1014	0.0554	0.1146
5	0.1342	0.0925	0.1035	0.0592	0.1138
6	0.1274	0.1031	0.1077	0.0628	0.116
7	0.1194	0.1124	0.1092	0.0663	0.1165
8	0.1206	0.1283	0.1169	0.0704	0.1232
9	0.1156	0.1359	0.1176	0.0727	0.1254
10	0.112	0.1457	0.1194	0.0756	0.1285

(c) Normalized TF-IDF as weighting scheme(k=400)

k	Precision	Recall	F-score	MAP	nDCG
1	0.16	0.0266	0.044	0.0266	0.0989
2	0.1667	0.0511	0.075	0.0422	0.1074
3	0.163	0.0728	0.0953	0.0518	0.1142
4	0.1544	0.0878	0.1054	0.0596	0.1168
5	0.1467	0.1013	0.1127	0.0653	0.1182
6	0.1348	0.1116	0.115	0.0687	0.1193
7	0.1308	0.1219	0.1186	0.072	0.1227
8	0.1267	0.132	0.1216	0.0754	0.1264
9	0.1215	0.1419	0.1231	0.0784	0.1295
10	0.1169	0.1534	0.1246	0.081	0.1333

(d) Glasgow model as weighting scheme(k=100)

Table 2: Evaluation results for LSI model for k with best results

k	Precision	Recall	F-score	MAP	nDCG
1	0.0044	0.0004	0.0008	0.0004	0.0022
2	0.0067	0.0014	0.0024	0.0009	0.0042
3	0.0074	0.002	0.0032	0.0011	0.0049
4	0.0067	0.0023	0.0034	0.0012	0.0045
5	0.008	0.005	0.0057	0.0018	0.0053
6	0.0081	0.0068	0.0067	0.0021	0.0061
7	0.0083	0.0076	0.0071	0.0022	0.0067
8	0.0089	0.0103	0.0086	0.0026	0.0081
9	0.0089	0.0116	0.0092	0.0027	0.0084
10	0.0098	0.0153	0.0109	0.0031	0.0101

(a) Basic Count as weighting scheme

k	Precision	Recall	F-score	MAP	nDCG
1	0.1467	0.0188	0.0322	0.0188	0.0911
2	0.1444	0.037	0.0562	0.0298	0.0942
3	0.1333	0.0545	0.0729	0.0365	0.0955
4	0.1322	0.0719	0.0879	0.0434	0.0998
5	0.1227	0.0827	0.0932	0.048	0.0986
6	0.1244	0.0989	0.1041	0.0535	0.1046
7	0.1187	0.1088	0.1072	0.0574	0.1074
8	0.1144	0.1197	0.1105	0.0604	0.11
9	0.1131	0.1318	0.1148	0.0635	0.1145
10	0.1098	0.1434	0.1172	0.066	0.1184

(b) TF-IDF as weighting scheme

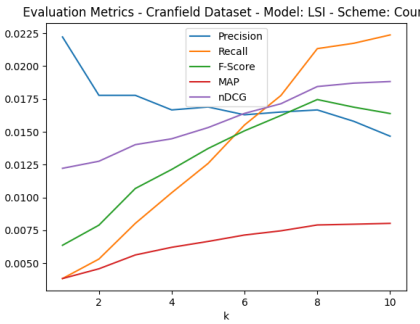
k	Precision	Recall	F-score	MAP	nDCG
1	0.1511	0.021	0.036	0.021	0.0911
2	0.1356	0.0359	0.0545	0.0309	0.0898
3	0.1378	0.0564	0.0763	0.0395	0.095
4	0.1322	0.0731	0.0892	0.0458	0.098
5	0.1307	0.0889	0.1	0.0514	0.102
6	0.1319	0.1041	0.1098	0.0573	0.1093
7	0.1257	0.1168	0.1141	0.0612	0.1127
8	0.1194	0.125	0.1151	0.064	0.1145
9	0.118	0.1377	0.1198	0.0674	0.1191
10	0.1124	0.1452	0.1196	0.069	0.1216

(c) Normalized TF-IDF as weighting scheme

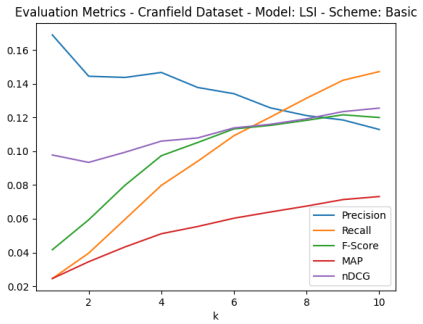
k	Precision	Recall	F-score	MAP	nDCG
1	0.1289	0.0166	0.0287	0.0166	0.0756
2	0.16	0.0441	0.0662	0.0323	0.0981
3	0.1556	0.0654	0.0871	0.0416	0.1008
4	0.1478	0.082	0.0995	0.0489	0.1043
5	0.1467	0.0994	0.1119	0.0561	0.1105
6	0.137	0.11	0.1152	0.0601	0.1127
7	0.1333	0.1222	0.1203	0.0638	0.1168
8	0.1256	0.1301	0.1204	0.0664	0.1181
9	0.118	0.1369	0.1195	0.0683	0.1192
10	0.1124	0.1452	0.1196	0.0703	0.1214

(d) Glasgow model as weighting scheme

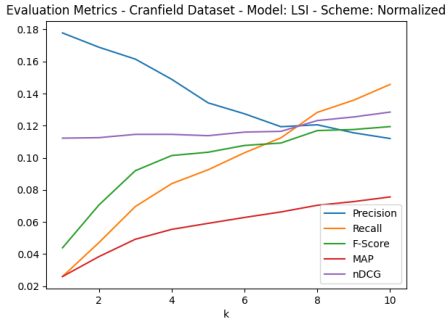
Table 3: Evaluation results for LSI model for k = 1400



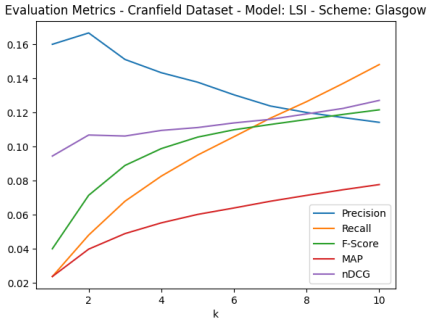
(a) Basic Count



(b) TF-IDF



(c) Normalized TF-IDF



(d) Glasgow model

Fig. 2: LSI Evaluation plots

4.1.3 Bidirectional Encoder Representations from Transformers (BERT)

k	Precision	Recall	F-score	MAP	nDCG
1	0.6667	0.1132	0.1857	0.1132	0.5333
2	0.5889	0.1958	0.2764	0.1858	0.5118
3	0.5022	0.2402	0.3028	0.2222	0.4784
4	0.4444	0.2759	0.3157	0.2468	0.4655
5	0.4089	0.3129	0.3276	0.269	0.4628
6	0.3778	0.3403	0.3305	0.2849	0.4635
7	0.3422	0.3553	0.3218	0.2938	0.4595
8	0.3261	0.3813	0.3243	0.3057	0.4653
9	0.3101	0.4045	0.3242	0.3149	0.4711
10	0.2933	0.4202	0.3198	0.3215	0.4741

Table 4: Evaluation results for BERT model

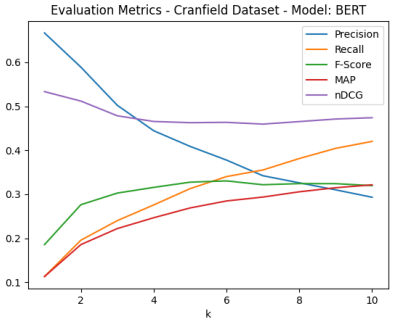


Fig. 3: Basic Count

Fig. 4: BERT Evaluation plots

4.2 Inferences

1. VSM performs surprisingly better than LSI for Cranfield Dataset and almost matches the efficiency results of BERT.
2. VSM takes less time to run than LSI.
3. LSI performs very poorly in all evaluation metrics.
4. LSI's performance increases in all metrics as we increase the number of hidden concepts (a hyperparameter) until a certain value. After which, performance starts declining.
5. Comparitively good performance is obtained even on fewer hidden concepts (such as 40, 50).
6. Count weighting scheme is the least efficient among all the weighting schemes.
7. Normalized TF-IDF and Glasgow model are the best weighting schemes.
8. BERT performs the best of all three models.

5 Hypothesis Testing

Statistical hypothesis tests are an important tool in statistics and machine learning. They provide a framework for making inferences about population parameters based on sample data and help determine whether observed differences or relationships in the data are statistically significant or due to random chance.

A statistical hypothesis test involves formulating a null hypothesis (H_0) and an alternative hypothesis (H_1). The null hypothesis typically assumes no effect, no difference, or no relationship between variables, while the alternative hypothesis proposes a specific effect, difference, or relationship.

In machine learning, hypothesis testing is frequently used to compare the performance of two models or algorithms. Based on the results obtained after hypothesis testing, a decision is made on which model or algorithm to proceed with, in the production.

5.1 Wilcoxon signed-rank test

The Wilcoxon signed-rank test is a non-parametric statistical test used to compare two related or paired samples. It is an alternative to the paired t-test when the assumptions of the t-test, such as the normality of the differences or interval-level data, are not met. The test is particularly useful when dealing with ordinal or non-normally distributed data.

The Wilcoxon signed-rank test assesses whether there is a significant difference between the medians of the paired differences in the samples. The null hypothesis (H_0) assumes that the median difference is zero, indicating no significant difference between the paired observations. The alternative hypothesis (H_1) suggests that the median difference is not zero.

It is worth noting that the Wilcoxon signed-rank test does not provide information about the direction of the difference. It simply determines whether a significant difference exists. If the alternative hypothesis suggests a specific direction, such as one sample being greater than the other, additional statistical tests or analyses may be needed.

However, if we have a specific expectation or hypothesis about the direction of the difference, we can instead use a directional alternative hypothesis. The directional alternative hypothesis can be one-sided, either specifying that the median difference is greater than zero (one-tailed upper), or that the median difference is less than zero (one-tailed lower).

Using a one-sided alternative hypothesis, we can test specifically for a positive or negative difference between the paired samples.

The Wilcoxon signed-rank test is widely used in various fields, particularly in studies with small sample sizes or non-normal or ordinal data. Its non-parametric nature and ability to handle paired observations make it a valuable tool for comparing related samples and drawing conclusions based on the observed data.

5.2 Stating the hypotheses

Hypothesis. M_1 does better than M_2 in the task of finding relevant documents given a query on the nDCG measure at rank 10 over the Cranfield dataset under the assumption that the nDCGs of both models are independent and identically distributed (iid), nDCGs of each model can be ranked, and paired.

In the above statement, M_1 and M_2 will take all possible pairs of a total of 9 models - (4 models of VSM, 4 models of LSI, and 1 model of BERT). So, there would be $\binom{9}{2} = 36$ such hypotheses in total. We show the results of the hypothesis tests for each of these hypotheses in the below subsection.

5.3 Results

Model-1	Model-2	Statistics	p_value	Result
VSM(Count)	VSM(TF-IDF)	234369.5	1.0	VSM(Count) < VSM(TF-IDF)
VSM(Count)	VSM(Normalized)	226398.5	1.0	VSM(Count) < VSM(Normalized)
VSM(Count)	VSM(Glasgow)	292834.5	1.0	VSM(Count) < VSM(Glasgow)
VSM(TF-IDF)	VSM(Normalized)	4256.5	0.9999999999999969	VSM(TF-IDF) < VSM(Normalized)
VSM(TF-IDF)	VSM(Glasgow)	405318.5	0.02604514319052293	VSM(TF-IDF) > VSM(Glasgow)
VSM(Normalized)	VSM(Glasgow)	426779.5	3.863999557477115e-06	VSM(Normalized) > VSM(Glasgow)
VSM(Count)	LSI(Count)	786617.0	6.782067003284755e-200	VSM(Count) > LSI(Count)
VSM(Count)	LSI(TF-IDF)	755537.5	3.910396034462176e-199	VSM(Count) > LSI(TF-IDF)
VSM(Count)	LSI(Normalized)	757416.5	4.1659331233333206e-201	VSM(Count) > LSI(Normalized)
VSM(Count)	LSI(Glasgow)	784160.0	7.545044072758376e-202	VSM(Count) > LSI(Glasgow)
VSM(TF-IDF)	LSI(Count)	1059150.5	3.308633090556822e-236	VSM(TF-IDF) > LSI(Count)
VSM(TF-IDF)	LSI(TF-IDF)	1041033.0	7.3028379339915765e-236	VSM(TF-IDF) > LSI(TF-IDF)
VSM(TF-IDF)	LSI(Normalized)	1048515.5	6.601989930358003e-237	VSM(TF-IDF) > LSI(Normalized)
VSM(TF-IDF)	LSI(Glasgow)	1050835.0	1.0264669038789418e-236	VSM(TF-IDF) > LSI(Glasgow)
VSM(Normalized)	LSI(Count)	1063693.0	7.564022391738553e-237	VSM(Normalized) > LSI(Count)
VSM(Normalized)	LSI(TF-IDF)	1045411.5	2.157370804895872e-236	VSM(Normalized) > LSI(TF-IDF)
VSM(Normalized)	LSI(Normalized)	1052973.0	1.711771400738323e-237	VSM(Normalized) > LSI(Normalized)
VSM(Normalized)	LSI(Glasgow)	1055280.5	2.7564895324700993e-237	VSM(Normalized) > LSI(Glasgow)
VSM(Glasgow)	LSI(Count)	992762.0	2.207375631211992e-229	VSM(Glasgow) > LSI(Count)
VSM(Glasgow)	LSI(TF-IDF)	984184.5	2.089316555727617e-229	VSM(Glasgow) > LSI(TF-IDF)
VSM(Glasgow)	LSI(Normalized)	991328.5	2.4976840037167333e-230	VSM(Glasgow) > LSI(Normalized)
VSM(Glasgow)	LSI(Glasgow)	979637.0	1.3388571539099098e-228	VSM(Glasgow) > LSI(Glasgow)
LSI(Count)	LSI(TF-IDF)	8571.0	0.05648201175585025	LSI(Count) = LSI(TF-IDF)
LSI(Count)	LSI(Normalized)	6837.0	3.796760205362756e-05	LSI(Count) > LSI(Normalized)
LSI(Count)	LSI(Glasgow)	12770.0	0.47654527618489856	LSI(Count) = LSI(Glasgow)
LSI(TF-IDF)	LSI(Normalized)	3284.5	0.002353270188034162	LSI(TF-IDF) > LSI(Normalized)
LSI(TF-IDF)	LSI(Glasgow)	1696.5	0.9999998757775934	LSI(TF-IDF) < LSI(Glasgow)
LSI(Normalized)	LSI(Glasgow)	3043.0	0.9999995569952309	LSI(Normalized) < LSI(Glasgow)
BERT	VSM(Count)	780212.0	4.563794838551923e-47	BERT > VSM(Count)
BERT	VSM(TF-IDF)	625865.0	0.0029741836504165262	BERT > VSM(TF-IDF)
BERT	VSM(Normalized)	614633.5	0.028088796288138165	BERT > VSM(Normalized)
BERT	VSM(Glasgow)	646536.0	2.04469495172852e-05	BERT > VSM(Glasgow)
BERT	LSI(Count)	1033320.0	1.8572741014943806e-233	BERT > LSI(Count)
BERT	LSI(TF-IDF)	1032377.5	7.16890743846183e-235	BERT > LSI(TF-IDF)
BERT	LSI(Normalized)	1034088.5	2.789988902740237e-235	BERT > LSI(Normalized)
BERT	LSI(Glasgow)	1033767.5	5.435260469758578e-235	BERT > LSI(Glasgow)

Table 5: Wilcoxon signed-rank test Results

6 Conclusion

In the end, we conclude that BERT performs the best among all three discussed models (VSM, LSI, BERT). Our results are supported both by theoretical arguments as well as statistical arguments (i.e., hypothesis testing).

VSM performs way better than LSI and even reaches very close to BERT.

LSI's performance is surprisingly the least among all three. In fact, it is very low compared to the other two models in all the evaluation metrics used.

Among the different weighting schemes used, the normalized TF-IDF and the Glasgow model are supposed to perform better (both for VSM and LSI) than the simple TF-IDF weights, which in turn perform better than the simple count weighting scheme.

References

[1] Salton, G., Wong, A., & Yang, C. S. (1975). A vector space model for automatic indexing. Communications of the ACM, 18(11), 613-620.

[2] Deerwester, Scott, et al. "Improving information-retrieval with latent semantic indexing." Proceedings of the ASIS annual meeting. Vol. 25. 143 Old Marlton Pike, Medford, NJ 08055-8750: Information Today Inc, 1988.

[3] Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. "Attention is all you need." Advances in neural information processing systems 30 (2017).