1 Introduction

In this assignment, I conducted IR Evaluation, Text Analysis, and Text Classification tasks. I implemented custom evaluation metrics for IR systems, performed token and topic analysis on given texts, and developed a sentiment classifier for tweets. Using Python and libraries like Pandas, NLTK, and Scikit-learn, I structured my code into classes for modularity. This report will be compartmentalized into three main modules: IR Evaluation, Text Analysis, and Text Classification. One challenge I faced was accurately implementing the text classification, which required the evaluation of a larger selection of models and the refinement of their respective parameters to establish an optimal solution. Through this assignment, I deepened my understanding of text analysis techniques and improved my coding proficiency.

2 IR Evaluation

In this assignment, we evaluated six information retrieval systems using evaluation metrics such as P@10, R@50, r-precision, AP, nDCG@10, and nDCG@20. Here are the average scores for each system on these metrics:

SystemID	P@10	R@50	r-precision	AP	nDCG@10	nDCG@20
1	0.390	0.834	0.401	0.400	0.350	0.487
2	0.220	0.867	0.252	0.300	0.212	0.265
3	0.410	0.767	0.448	0.451	0.416	0.515
4	0.080	0.189	0.049	0.075	0.071	0.080
5	0.410	0.767	0.358	0.364	0.330	0.432
6	0.410	0.767	0.448	0.445	0.390	0.489

Tabell 1: Retrieve the performance indicators of the system

Based on the average score, we identified the best-performing system under each metric and performed a two-tailed T-test for the best system versus the second-place system to determine whether the difference was statistically significant.

2.1 P@10

Best systems: Systems 3, 5, and 6 (average P@10 score of 0.410). Second-Best system: System 1 (average P@10 score 0.390). Statistical significance test: Compared with the best systems, the p value of system 1 is 0.7509, and the difference is not significant. Conclusions: Although systems 3, 5, and 6 have the highest scores on P@10, the difference from system 1 is not statistically significant.

2.2 R@50

Best System: System 2 (with an average R@50 score of 0.867). Second-Best System: System 1 (with an average R@50 score of 0.834). Statistical Significance Test: The comparison between System 2 and System 1 yields a p-value of 0.3434, indicating no statistically significant difference. Conclusion: System 2 achieves the highest R@50 score; however, the difference compared to System 1 is not statistically significant.

2.3 r-precision

Best Systems: Systems 3 and 6 (with an average r-precision score of 0.448). Second-Best System: System 1 (with an average r-precision score of 0.401). Statistical Significance Test: The comparison between the best systems and the second best system yields a p-value of 0.5911, indicating no statistically significant difference. Conclusion: Systems 3 and 6 achieve the highest r-precision scores; however, the differences compared to System 1 are not statistically significant.

2.4 Average Precision (AP)

Best System: System 3 (with an average AP score of 0.451). Second-Best System: System 6 (with an average AP score of 0.445). Statistical Significance Test: The comparison between System 3 and System 6 yields a p-value of 0.6757, indicating no statistically significant difference. Conclusion: System 3 achieves the highest AP score; however, the difference compared to System 6 is not statistically significant.

2.5 nDCG@10

Best System: System 3 (with an average nDCG@10 score of 0.416). Second-Best System: System 6 (with an average nDCG@10 score of 0.390). Statistical Significance Test: The comparison between System 3 and System 6 yields a p-value of 0.3065, indicating no statistically significant difference. Conclusion: System 3 achieves the highest nDCG@10 score; however, the difference compared to System 6 is not statistically significant.

2.6 nDCG@20

Best System: System 3 (with an average nDCG@20 score of 0.515). Second-Best System: System 6 (with an average nDCG@20 score of 0.489). Statistical Significance Test: The comparison between System 3 and System 6 yields a p-value of 0.2823, indicating no statistically significant difference. Conclusion: System 3 achieves the highest nDCG@20 score; however, the difference compared to System 6 is not statistically significant.

Based on the above analysis, System 3 performs the best across multiple metrics; however, the differences between System 3 and other systems are not statistically significant, making it inconclusive that System 3 is significantly better on these metrics. Additionally, the performance of System 4 is significantly lower than that of other systems, which is statistically validated.

3 Text Analysis

In this section, we analyze the tokens from three religious corpora: the Old Testament (OT), the New Testament (NT), and the Quran. We computed the Mutual Information (MI) and χ^2 (Chi-Squared) scores for all tokens after preprocessing, generating ranked lists of tokens for each corpus.

3.1 Token Analysis

Tabell 2: New Testament (NT)

Rank	**Token**	**MI Score**	**Token**	χ^2 **Score**
1	jesu	0.05208	jesu	3011.11
2	christ	0.03177	christ	1760.65
3	lord	0.02203	lord	870.33
4	shall	0.01647	discipl	819.83
5	discipl	0.01429	shall	620.68
6	israel	0.01376	peter	528.99
7	peopl	0.00991	paul	521.51
8	king	0.00987	thing	501.12
9	peter	0.00953	israel	450.33
10	paul	0.00951	spirit	436.66

From Table 2, both MI and χ^2 identify **jesu** and **christ** as the most significant tokens, reflecting the centrality of Jesus Christ in the NT. MI ranks **shall** and **disciples** slightly lower than χ^2, suggesting these words are informative but not uniquely characteristic (**shall** and **disciples** also frequently appear in other corpora). Tokens like **peter** and **paul** have higher χ^2 scores due to their significant frequency differences compared to other corpora.

Tabell 3: Old Testament (OT)

Rank	**Token**	**MI Score**	**Token**	χ^2 **Score**
1	shall	0.03771	shall	1504.94
2	jesu	0.03708	jesu	1457.28
3	israel	0.03108	lord	1115.94
4	king	0.02600	believ	1109.85
5	lord	0.02576	israel	1093.89
6	believ	0.02575	king	944.77
7	christ	0.01974	god	789.55
8	god	0.01654	christ	774.37
9	muhammad	0.01552	muhammad	608.74
10	certainli	0.01299	certainli	569.63

From Table 3, MI emphasizes **israel**, **king**, and **lord**, reflecting the OT's focus on the nation of Israel, monarchy, and divine authority. χ^2 highlights **believ** and **certainli**, indicating these words have notable frequency differences.

Tabell 4: Quran

Rank	**Token**	**MI Score**	**Token**	χ^2 **Score**
1	god	0.03170	muhammad	1852.13

(Continued on next page)

Rank	Token	MI Score	Token	χ^2 Score
2	muhammad	0.02883	god	1727.87
3	certainli	0.02467	certainli	1682.74
4	believ	0.02284	believ	1526.51
5	torment	0.01964	torment	1332.81
6	messeng	0.01550	messeng	1057.61
7	shall	0.01479	revel	941.56
8	revel	0.01389	unbeliev	848.87
9	king	0.01308	want	833.37
10	israel	0.01301	guidanc	810.93

From Table 4, **god** and **muhammad** are consistently top-ranked, underscoring their importance in the Quran. MI brings attention to **torment**, **messeng**, and **revel**, tokens central to Quranic themes. χ^2 ranks **unbeliev**, want, and **guidanc** highly, pointing to themes of faith and guidance.

Mutual Information (MI) measures the informativeness of a token for a particular corpus, favoring tokens that are unique or highly characteristic. Chi-Squared (χ^2) measures the statistical significance of the difference in frequency of a token between corpora, highlighting tokens with large frequency disparities.

3.2 Topic Analysis

Using Latent Dirichlet Allocation (LDA) with 20 topics on the combined corpus, we identified the most associated topic for each corpus by calculating the average topic distribution across documents.

3.2.1 Old Testament (OT)

Most Associated Topic: Topic 13 (Divine Commands)
Average Topic Score: 0.0781
Top 10 Tokens:

Tabell 5: Divine Commands

Token	Probability
say	0.1217
lord	0.1106
peopl	0.0641
messeng	0.0542
god	0.0525
sent	0.0393
thou	0.0378
mose	0.0344
command	0.0278
israel	0.0253

The top tokens suggest themes of divine communication (“say”, “lord”, “messeng”, “god”), prophecy (“sent”, “command”, “mose”), and the people of Israel (“people”, “israel""). The assigned label reflects the OT’s focus on laws, commandments, and prophetic messages delivered to the Israelites.

3.2.2 New Testament (NT)

Most Associated Topic: Topic 7 (Teachings and Dialogue)
Average Topic Score: 0.1107
Top 10 Tokens:

Tabell 6: Teachings and Dialogue

Token	Probability
said	0.1117
us	0.0787
let	0.0531
truth	0.0390
book	0.0300
god	0.0287
come	0.0272
one	0.0266
ask	0.0266
know	0.0228

Tokens like “said”, “let”, “truth”, “ask”, and “know” indicate dialogues and teachings central to the NT, particularly the interactions of Jesus with his disciples and others. The presence of “book”and “come” also aligns with the themes of new covenant and the spreading of teachings.

3.2.3 Quran

Most Associated Topic: Topic 8 (Divine Knowledge and Creation)
Average Topic Score: 0.1271
Top 10 Tokens:

Tabell 7: Divine Knowledge and Creation

Token	Probability
god	0.1293
earth	0.0484
know	0.0461
thing	0.0459
one	0.0417
heaven	0.0351
judgment	0.0272
day	0.0266

Token	Probability
deed	0.0261
people	0.0254

The dominant tokens revolve around monotheism (“god”, “one”), cosmology (“earth”, “heaven”), eschatology (“judgment”, “day”, “deed”), and divine knowledge (“know”, “thing”). The topic captures the Quran’s emphasis on God’s creation, knowledge, and the accountability of humanity.

4 Text Classification

In this section, we develop a sentiment analyzer to classify tweets into three categories: positive, negative, or neutral. We begin by creating a baseline model using Bag-of-Words (BOW) features and then attempt to improve upon it. Below, we detail the steps taken, the challenges faced, the results obtained, and the analysis of these results.

4.1 Preprocessing

1. Converted all text to lowercase.
2. Removed URLs, mentions (@usernames), and hashtags (keeping the word but removing the ’#’).
3. Removed punctuation and numerical characters.
4. Tokenized the text.
5. Removed stopwords using NLTK’s English stopword list.
6. Applied stemming or lemmatization using the Porter Stemmer or Word Net Lemmatizer.

4.2 Feature Extraction

We used the CountVectorizer from scikit-learn to convert the preprocessed text into BOW features. When trying to improve the performance of the model, we modified the CountVectorizer to include unigrams and bigrams (ngram_range=(1, 2)). However, this does not always work. Another option is to use TfidfVectorizer to convert the clean text into TFIDF features. Contrary to expectations, using TF-IDF features resulted in lower performance compared to BOW.

In an attempt to further improve the performance of the sentiment classifier, we tried to incorporate non-textual features alongside the textual ones. For example, tweets with hashtags might express strong emotions or trends, mentions could indicate direct interactions that carry specific sentiments, and the presence of URLs might be associated with promotional or informational content, potentially affecting the sentiment.

But these extra features don’t always work. It can indeed improve the performance of the SVM model, but it will cause the Logistic model to fail to converge.

4.3 Model selection

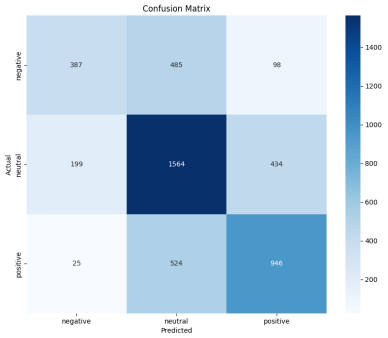
4.3.1 Multinomial Naive Bayes

Accuracy: 62.14%
Macro F1-Score: 59.39%

Figur 1: table

Classification Report

Class	Precision	Recall	F1-Score	Support
Negative	0.63	0.40	0.49	970
Neutral	0.61	0.71	0.66	2197
Positive	0.64	0.63	0.64	1495



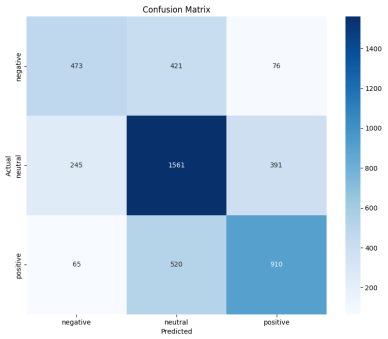
Figur 2: Confusion matrix: MultinomialNB

4.3.2 Logistic Regression

Accuracy: 63.15%
Macro F1-Score: 61.26%

Tabell 8: Classification Report

Class	Precision	Recall	F1-Score	Support
Negative	0.60	0.49	0.54	970
Neutral	0.62	0.71	0.66	2197
Positive	0.66	0.61	0.63	1495



Figur 3: Confusion matrix: LogisticRegression

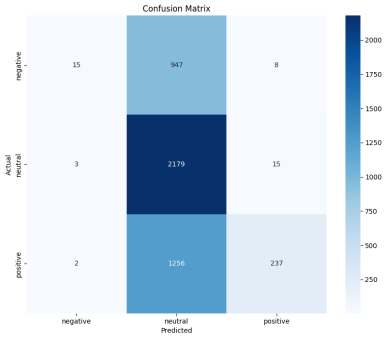
4.3.3 SGDClassifier

Accuracy: 52.15%
Macro F1-Score: 32.09%

Figur 4: table

Classification Report

Class	Precision	Recall	F1-Score	Support
Negative	0.75	0.02	0.03	970
Neutral	0.50	0.99	0.66	2197
Positive	0.91	0.16	0.27	1495



Figur 5: Confusion matrix: SGDClassifier

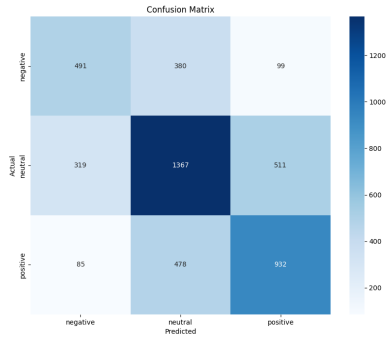
4.3.4 LinearSVC

Accuracy: 59.85%
Macro F1-Score: 58.62%

Classification Report

Figur 6: table
Classification Report

Class	Precision	Recall	F1-Score	Support
Negative	0.55	0.51	0.53	970
Neutral	0.61	0.62	0.62	2197
Positive	0.60	0.62	0.61	1495



Figur 7: Confusion matrix: LinearSVC

5 Conclusion

In this assignment, we completed three tasks: information retrieval evaluation, text analysis and text classification.

Information retrieval Evaluation: We evaluated the performance of six retrieval systems. Although some systems performed better on average scores, the difference was not significant when tested statistically.

Text analysis: By calculating mutual information and Chi-square tests, we identify representative words in the Old Testament, the New Testament and the Qur ’an. Using the LDA model, we find the themes of each corpus and the connections between them, and deepen the understanding of the text content.

Text classification: We built an emotion classifier to analyze tweets. Despite the addition of non-text features (such as text length, whether hash tags are included, etc.), model performance is not significantly improved. The macro average F1 score of some models is improved by introducing the methods of double word and handling negative words.

Overall, the assignment made us realize the importance of evaluation methods, feature selection, and model improvement in text technology. Through practice, we deepen our understanding of textual data processing and analysis, laying the foundation for future research and application.