Text Technologies for Data Science

s1862323

Chapter 1: Overall Work

Coursework 2 is split into three main components: IR evaluation, Text analysis and text classification.

Part 1: IR evaluation

The code is an implementation of an IR system. This code has various evaluation metrics for IR systems including precision@10, recall@50, r-precision, average precision, and normalized discounted cumulative gain (nDCG) at the cutoffs 10 and 20. Based on the average scores achieved for each system, I obtained the best system based on each score. For each best system with a given score, I found that none of the evaluation values were statistically significantly better than the second system by using a two-tailed t-test with a p-value of 0.05.

As a result of the development, I have better understood the formulas and implemented them in the code. Also, based on the improvements made to the data framework, the IR system has become more efficient. The average time to generate an ir_eval.csv file has been reduced to 5 seconds.

Part 2: Text analysis

I implemented the Quran, New Testament, and Old Testament each to be a separate corpus, and their verses as individual documents. On the WordLevel, I computed the Mutual Information and X2 scores for all tokens (after preprocessing) for each of the three corpora. And generate a ranked list of the results, in the format token score. On the Topic level, I applied LDA method to find the top 10 tokens and their probability.

I took the Quran, NT, OT verses as the verses collection. And the highest average score (3 topics in total). For each of those three topics, find the top 10 tokens with highest probability of belonging to that topic.

As a result of development, I have better understood three corpora and know their similarities and differences. The advantage of LDA model and MI, Chi-Square limitation.

Part 3: Text classification

In this section, I perform text classification by extracting bag-of-words (BOW) features and training a support vector machine (SVM) classifier to predict labels. I evaluate the model's performance on each dataset (training, development, and testing) by calculating precision, recall, and f1-score for each class, as well as macro-averaged precision, recall, and f1-score across all classes. To improve the performance of the baseline system, I experiment with two methods: change the SVM parameters (C=10, kernel='rbf') and change the preprocess way for documents (keep the stopwords).

Chapter 2: IR Evaluation

Table 1 shows a summary of the average scores for each IR system

	P@10	R@50	r-precision	AP	nDCG@10	nDCG@20
S1	0.390	0.834	0.401	0.400	0.363	0.485
S2	0.220	0.867	0.253	0.300	0.200	0.246
S3	0.410	0.767	0.448	0.451	0.420	0.511
S4	0.080	0.189	0.049	0.075	0.069	0.076
S5	0.410	0.767	0.358	0.364	0.332	0.424
S6	0.410	0.767	0.448	0.445	0.400	0.491

Table 2 shows the best performance IR system according to each metric.

	P@10	R@50	r-precision	AP	nDCG@10	nDCG@20
Best	S3,S5,S6	S2	S3,S6	S3	S3	S3
Value	0.41	0.867	0.448	0.451	0.42	0.511

Table 3 shows the two-tailed test results according to each metric.

	P@10	R@50	r-precision	AP	nDCG@10	nDCG@20
p-value	0.888	0.703	0.759	0.967	0.882	0.869

The table 2 shows that S3, S5 and S6 have the best P@10 performance with 0.41. While even a perfect system would have a P@K score of less than 1, our system shows P@10=0.41, which means that our system showing 4 relevant documents in the top 10 documents. The three system have the same score; hence they are not trivially different. The two-tailed test result 0.888 shows the same meaning.

The table 2 shows that system 2 performs best in the R@50 with 0.867. The second-best system S1 with 0.834. According to Table 3 two-tailed test result, the p-value is 0.703. This means the difference is not statistically significant and the difference could have happened by chance.

The table 2 shows the system 3 performs best in the r-precision, AP, nDCG@10, nDCG@20. The S3 and S6 have the same result in r-precision, so it will not pass the two-tailed test. However, according to the table 3 results inn AP, nDCG@10 and nDCG@20, The p-value are 0.967, 0.882 and 0.869 respectively. This means the difference are not statistically significant and the difference could have happened by chance.

The two-tailed test with p-value of 0.05 was used to indicate whether the best system was statistically significantly better than the second system. According to the results in Table 3, the p-values are all greater than the significance level of 0.05. It does not reject the null hypothesis. It means that the observed data is not sufficient to show that there is a significant difference between each system. Therefore, the differences in each system occur by chance and are not genuine differences. This means that for the best system for each given score, that system is not statistically significantly better than the second system for that score.

Chapter 3: Text analysis

Part 1: Word Level analysis

Table 4 shows the top 10 highest score words for Mutual information and Chi-squared(X^2)

	Quran		OT		NT	
	MI	X^2	MI	X^2	MI	X^2
1	god 0.0311	muhammad 1667.179	jesus 0.0386	jesus 1334.869	jesus 0.0566	jesus 2908.463
2	muhammad 0.0302	god 1506.601	israel 0.0363	lord 1214.012	christ 0.0344	christ 1697.684
3	torment 0.0205	torment 1204.042	king 0.0314	israel 1177.843	lord 0.0237	lord 857.818
4	believ 0.0202	believ 1197.830	lord 0.0307	king 1045.074	israel 0.0153	discipl 778.895
5	messeng 0.0160	messeng 944.798	ot 0.0227	christ 709.808	discipl 0.0152	nt 539.668
6	king 0.0158	revel 846.744	christ 0.0206	god 695.797	peopl 0.0115	paul 507.351
7	israel 0.0155	quran 814.918	believ 0.0185	believ 682.372	king 0.0114	peter 507.351
8	quran 0.0147	unbeliev 763.421	son 0.0163	ot 631.651	nt 0.0109	thing 461.758
9	revel 0.0144	guidanc 730.740	god 0.0162	son 620.278	ot 0.0109	israel 458.498
10	unbeliev 0.0130	disbeliev 708.902	muhammad 0.0160	muhammad 553.875	land 0.0103	spirit 406.494

Chi-squared and mutual information are both measures that can be used to assess the relationship between two variables. Chi-squared is a measure of association between two variables. It is commonly used in statistical hypothesis testing to assess whether there is a significant association between the two variables. Unlike mutual information, chi-squared can only detect linear relationships between variables. Mutual information is a measure of the mutual dependence between two variables. It quantifies the amount of information that one variable provides about the other. In other words, it measures how much knowing the value of one variable can tell us about the value of the other variable. Mutual information can detect any type of relationship between two variables, including non-linear relationships.

In the Quran, the first 5 words are almost the same in MI and X^2. After that, the ranked words became different. In the OT and NT, the first 10 words are mostly in the same rank or adjacent rank. I observed the god, muhammad and torment are the most popular words in Quran. In contrast, jesus, israel, lord, king and christ are the most popular words in OT and NT. Based on the background of Quran, OT and NT, The Quran is the central religious text of Islam and is revered by Muslims as the word of God. The OT and NT are part of the Christian Bible and are revered by Christians as the word of God. This difference also shows in our top 10 words.

In summary, the main difference between chi-squared ranked data and mutual information ranked data is the type of relationship they can detect. Chi-squared can only detect linear relationships, while mutual information can detect any type of relationship.

Part 2: Topic level analysis

Table 5 shows the top 10 tokens with highest probability of belonging to that topic.

	1st	2nd	3rd	4th	5th	6th	7th	8th	9th	10th
Quran Topic	angel 0.158	god 0.068	come 0.049	lord 0.049	bodi 0.0436	grace 0.0346	sound 0.031	soul 0.031	worship 0.030	stand 0.022
OT Topic	israel 0.061	children 0.061	put 0.053	tribe 0.052	gate 0.033	mind 0.032	end 0.031	lord 0.028	number 0.025	cloud 0.023
NT Topic	jesus 0.162	son 0.104	written 0.043	book 0.035	day 0.034	jew 0.032	elder 0.030	prophet 0.027	hour 0.025	belov 0.025

I wanted to give each corpus the same religious label, while in the second part it was different. The label as shown below.

- Quran label: religion and spirituality.
- OT label: religion and history.
- NT label: religion and inspiration.

LDA-Quran: Quran is the religious text. It contains the teachings and beliefs including its laws and moral principles. Quran is a sacred text that contains the exact words of God and is a source of guidance and wisdom for all aspects of life.

LDA-OT: OT contains the history of God's relationship with the people of Israel and the teachings of ancient Israel. These teachings include moral and ethical guidance

LDA-NT: The NT is revered by Christians as the word of God and is a source of spiritual guidance and inspiration. It contains Jesus' life and ministry. The NT also includes the Book of Revelation, which is a vision of the end of the world and the coming of the Kingdom of God.

There is only one token occur in both Quran and OT topic: lord. The rest of the tokens are all different. In contrast, MI and Chi-square word-level analysis had many similar tokens in the top 10. Because MI and X2 can be used to identify relationships between words in text. Therefore, high frequency tokens and its related tokens are more likely to be bundled together. Thus, the high frequency of related words would allow Quran, OT and NT to all produce the adjacent Top10 Token results for MI and Chi-square word-level analysis. However, for the LDA method, the tokens are mostly different.

Hence, LDA is a generative probabilistic model that provides detailed information about the topics present in the text and the words that are associated with each topic, which can be useful for understanding the themes and topics present in the text.

Chapter 3: Text classification

In this section, I perform text classification by extracting bag-of-words (BOW) features and training a support vector machine (SVM) classifier to predict labels. I evaluate the model's performance on each dataset (training, development, and testing) by calculating precision, recall, and f1-score for each class, as well as macro-averaged precision, recall, and f1-score across all classes. To improve the performance of the baseline system, I experiment with two methods: change the SVM parameters (C=10, kernel='rbf') and change the preprocess way for documents (keep the stopwords).

Improvement of baseline system

1. To improve the performance of the baseline system, I tried to change the C value to a smaller value. As C = 1000, which is too high for this dataset, I tried different C values from 100 to 10. This had no effect on the performance of the model. Because a lower C value would have reduced resource costs and time spent, adjusting the C value to a smaller value had all positive effects on the current model. If the dataset is adjusted later, consider adding the Early Stopping setting to the model.
2. In order to improve the model performance, I have tweaked the previous pre-processing of the text. After my research and observation of STOPWORDS, I found that keeping STOPWORDS would have been a better choice. The actual results also show that keeping STOPWORDS has a positive effect on the performance of the model
3. To improve the model performance, I changed the kernel of the model to Radial Basis Function (RBF). The advantage of using the Radial Basis Function (RBF) kernel in support vector machines (SVM) is that it can model more complex, non-linear decision boundaries compared to linear SVMs. The RBF kernel is defined as the similarity between the input vectors (x and z) measured by the Euclidean distance between them. This allows the SVM to learn and model non-linear decision boundaries. Another advantage of the RBF kernel is that it can automatically learn the best value for the hyperparameter gamma, which controls how much influence a single training example has on the model. In contrast, the gamma parameter must be manually set when using a linear SVM. In summary, the RBF kernel allows for more flexible and complex decision boundaries and can automatically learn the best value for the gamma parameter, making it a good choice for many classification tasks.

Experiment result and summary

According to the previous improvement method, the comparison between baseline system Macro-F1 and the improved system Macro-F1 as shown below.

System_split	Macro-F1_precision	Macro-F1_recall	Macro-F1_f1-score
Baseline_dev	0.506	0.514	0.508
Baseline_test	0.524	0.539	0.529
Improved_dev	0.634	0.574	0.590
Improved _test	**0.647**	**0.593**	**0.609**

For the baseline dev and improved dev, the improvement of precision was from 0.506 to 0.634. A 25.3% increase in precision and the growth is good. Correspondingly, the recall increased from 0.514 to 0.574, 11.6% growth. And the F1-score increased from 0.508 to 0.590, 16.1% growth. Overall, it showed a steady improvement.

In addition, for the baseline test and improved test, the improvement of precision was from 0.524 to 0.647. A 23.5% increase in precision and the growth is good. Accordingly, the recall increased from 0.539 to 0.593, 10.1% growth. And the F1-score increased from 0. 529 to 0.609, 15.1% growth. Overall, it also showed a steady improvement.

Future work and potential improvement methodologies

1. **TF-IDF**: TF-IDF is better than BOW in text classification because it weights the importance of each word in a document, allowing the model to focus on more relevant words and make more accurate predictions. It also helps reduce the impact of common words that may not be useful for classification.
2. **CNN model**: The main advantage of using a CNN for text classification over linear or RBF-SVM is its ability to capture complex patterns and interactions between words and their contexts, improving accuracy and performance. Linear and RBF-SVM use a linear model to classify documents based on a linear combination of features, which may not capture non-linear relationships or complex patterns in natural language. CNNs can learn to extract and combine features from input text, adapting to new patterns and relationships through training on large datasets. This allows the model to improve performance on unseen data and better generalize to different text types and tasks. Reference: (Y. Luan and S. Lin, 2019) (Z. Wang and Z. Qu, 2017) The paper raised a great improvement on precision, recall and F1-score. Which are 99% and 93.4% respectively.

References

Y. Luan and S. Lin. (2019). Research on Text Classification Based on CNN and LSTM,. *2019 IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA)*, pp. 352-355. doi: doi: 10.1109/ICAICA.2019.8873454.

Z. Wang and Z. Qu. (2017). Research on Web text classification algorithm based on improved CNN and SVM. *2017 IEEE 17th International Conference on Communication Technology (ICCT), 2017,*, pp. 1958-1961,. doi: doi: 10.1109/ICCT.2017.8359971.