THE UNIVERSITY *of* EDINBURGH

Text technologies for Data Science

Coursework 2

Student ID: S2258327

General discussion about the coursework

This report is divided intro three main sections: Infomation retrieval evaluation, Text analysis and Text classification.

The first section, Information retrieval evaluation, deals with the evaluation of six different systems given a set of queries, the documents retrieved by each system for a specific query, and the relevance of each document. This evaluation is performed according to six evaluation metrics (see Section 1 for more details). To solve this task, for each system I created a dictionary with the format 'query number' : {docid : (relevance, rank, score)}, and simply calculated the required metric for each system. All of this is capsulated onto a class called EVAL on my code.

The second section, Text Analysis, deals with the analysis of three diferent corpora, the Old Testament, the New Testament, and the Quran. To analyze the content of each corpus, the mutual information and chi-squared for all of the tokens in each corpus was calculated. Similarly, and Latent Dirichlet Allocation (LDA) was used to allocate the most probable topic for each corpus. Everything in this section was implemented from scratch, with the exception of the LDA which was implemented using gensim. To calculate the mutual information and the chi-squared, the N-terms count was calculated by starting with an index with format 'word' : {frequency in corpus 1, frequency in corpus 2, frequency in corpus 3}. All of this is encapsulated in the class TextAnalysis.

Lastly, the third section, Text Classification, deals with the classification of text by using a machine learning classifier. The dataset used for this task is the same as the data used in second section. That is, given a verse of any of the three books, predict to which book it is more likely to belong to. For this, a baseline model using a Support Vector Machine (SVM) was trained. Our task was to improve this baseline model's performance on the validation set. My improvement was to use TF-IDF values (with unigrams and bigrams) instead of counts, normalize the values, and to augment the data from the New Testament by duplicating the corpus and changing each word for its synonym. The baseline model's Macro avg F1-score on the validation set was 0.92, while my model achieved 0.96 on the validation set and 0.93 on the test set. Moreover, I changed the classifier into a Linear SVM, reducing the training time from approximately 6 minutes to roughly 1 second.

All the topics analyzed in this coursework were new for me, so I clearly learnt many things. Without any doubt, the most challeging task for me was the calculation of the N-terms (efficiently), and the improvement of the baseline classifier (since the baseline model's performance was already too high).

1. **Information Retrieval Evaluation**

In this section of the report, different evaluation metrics for six Information Retrieval (IR) systems are calculated and compared to select the best system overall. These metrics are calculated based on the documents retrieved by each system when fetched with 10 identical queries. The evaluation metrics used for this evaluation are: Precision, recall, r-precision, average precision (AP) and normalized discounted cumulative gain (nDCG). For the nDCG, a hand-labeled document containing the relevance of each documents for all queries was used. Table 1 shows the mean of the metrics (across all 10 queries) for each system.

*Table 1: Evaluation metric for each system (**bold** indicates best system/s for specific metric, underline indicates the 2nd best)*

	Precision at 10	Recall at 50	r-precision	AP	nDCG at 10	nDCG at 20
System 1	0.39	0.834	0.401	0.4	0.363	0.485
System 2	0.22	**0.867**	0.253	0.3	0.2	0.246
System 3	**0.41**	0.767	**0.449**	**0.451**	**0.42**	**0.511**
System 4	0.08	0.19	0.049	0.075	0.069	0.076
System 5	**0.41**	0.767	0.358	0.364	0.333	0.424
System 6	**0.41**	0.767	**0.449**	0.445	0.4	0.49

Table 1 shows the best system for a specific metric based on the mean value across all 10 queries. However, while this may give a good intuition, it is not clear if one system is statistically better than the others. In order to determine wheter one system is better than other for a specific metric, we can use a 2 tailed t-test with a p value of 0.05. This 2 tailed t-test is performed using all of the values for the system for that metric, not the mean. We will perform this test for the 1st and 2nd best systems for each metric.

For example, for the Precision at 10, we see that Systems 3,5,6 have the same value, 0.41. When the p value is calculated, it turns out to be 1, meaning that the systems are completely identical. However, if we compare these systems with System 1 (the 2nd best on that metric), the p value is 0.888, which is greater than 0.05, indicating that Systems 1,3,5,6 are statistically indistinguishible on that metric. Similarly, for Recall at 50, Systems 1,2 are indistinguishible (p=0.703). For r-precision, Systems 1,3,6 are indistinguishible (p=0.759). For AP, Systems 6,3 are indistinguisible(p=0.967). For nDCG at 10 and 20, Systems 3,6 are indistinguishible, (p=0.883) and (p=0.868), respectively.

Overall, System 3 is the best system since it has the best performance in most of the metrics. However, the t-test suggests that both System 3 and 6 are the best systems since they are statistically indistinguishible. In fact, all of the systems have a relatively decent performance with the exception of System 4.

2. Text Analysis

For this section, three corpora will be evaluated. The corpuses are: the Old Testament (OT), the New Testament (NT) and the Quran. More specifically, this section will focus on the comparison of the three corpora by using the Mutual Information (MI) and X^2 of each corpus (token analysis). Similarly, the Latent Dirichlet Allocation (LDA) will be run in order to understand the corpuses at the topic level (topic analysis).

2.1. Token Analysis

Table 2 and 3 contain the top 10 tokens obtained for each corpus using MI and X^2, respectively (sorted from highest value to lowest value).

Table 2: Top tokens for each corpus using MI

	Top tokens using mutual information									
Corpus	**Token 1**	**Token 2**	**Token 3**	**Token 4**	**Token 5**	**Token 6**	**Token 7**	**Token 8**	**Token 9**	**Token 10**
OT	jesu	israel	king	lord	christ	believ	god	muhammad	son	torment
NT	jesu	christ	lord	discipl	israel	king	paul	peter	land	thing
Quran	god	muhammad	believ	torment	messeng	revel	king	israel	unbeliev	guidanc

Table 3: Top tokens for each corpus using X^2

	Top tokens using X^2									
Corpus	**Token 1**	**Token 2**	**Token 3**	**Token 4**	**Token 5**	**Token 6**	**Token 7**	**Token 8**	**Token 9**	**Token 10**
OT	jesu	lord	israel	king	god	christ	believ	muhammad	faith	son
NT	jesu	christ	lord	discipl	paul	peter	thing	israel	spirit	john
Quran	muhammad	god	believ	torment	messeng	revel	unbeliev	guidanc	diesbeliev	quran

2.1.1 What differences do you observe between the rankings produced by the two methods (MI and X^2)?

The tokens found using MI and X^2 are very similar. However, it seems that with X^2, the tokens for each corpus depend more on the documents of that corpus rather than the others. For example, the token ‘King’ is predominant in the OT, but the MI of NT and Quran includes it while the X^2 does not. Similarly, the X^2 includes the token ‘quran’, while the MI also does, but at a lower ranking than the X^2.

2.1.2 What can you learn about the three corpora from these rankings?

The tokens obtained by using MI and X^2 give a decent intuition of the content of each corpus. Just by looking at the top tokens for each corpora, one can deduce that the three correspond to Abrahamic religions books. However, if one were to predict to which religious book each corpus corresponds to based on the tokens, that person would have a hard time distinguishing between the OT and NT, but not for the Quran. In the case of the OT and NT, the tokens are very similar. For example, the first token for the OT and NT is jesu, however, in the OT, this token does not appear on the whole corpus, but it is still a token that gives much information about that corpus. On the other hand, for the Quran, tokens like: ‘muhammad’, ‘messeng’ and ‘quran’ are a clear indicative that the corpus is the Quran, the Islam sacred’s book.

2.2. Topic analysis

To analyse the topics of each corpus, an LDA with k=20 topics was run using all of the corpuses as training data. Then, for each corpus, the same LDA was used to compute the average score for each topic by summing the document-topic probability for each document in that corpus and dividing by the total number of documents in the corpus. Then, for each corpus, the topic with the highest probability was identified. For each of those three topics, the top 10 tokens with highest probability of belonging to that topic were obtained. The LDA is an algorithm that involves much randomness, for this reason, I used the random_state=25, and set the number of iterations to 50,000 (this will ensure that the algorithm converges). Table 4 shows the result.

Table 4: Most probable topic for each corpus

Corpus	Topic
OT	19
NT	14
Quran	14

Top 10 tokens for topic 19: '0.129*"god" + 0.103*"believ" + 0.052*"lord" + 0.047*"king" + 0.044*"command" + 0.041*"judgment" + 0.039*"peopl" + 0.034*"law" + 0.033*"mose" + 0.031*"day"'

Top 10 tokens for topic 14: '0.122*"god" + 0.058*"lord" + 0.037*"heart" + 0.034*"peopl" + 0.033*"forgiv" + 0.032*"word" + 0.030*"thing" + 0.029*"reward" + 0.024*"power" + 0.021*"hear"'

2.2.1. Your own labels for the 3 topics. That is, in 1-3 words, what title would you give to each of the three topics?

In my own words, topic 19 refers to: ‘Moses and Pharaoh’, and topic 14 refers to: ‘Mercy of God’.

2.2.2. What does the LDA model tell you about the corpus? Are there any topics that appear to be common in 2 corpora but not the other? ...

While the LDA also gives an understanding of the corporas, the LDA analyzes the corpus at a topic level rather than at a word level (like MI and X^2). Apart from the NT and quran sharing the same topic, it is worth mentioning that the topic 14 is also common on the OT (2nd most probable topic). Similarly, topic 19 is also common in the Quran (2nd most probable topic). On the other hand, topic 17 is common in the OT and NT, but not in the quran. Common words for topic 17 are: ‘son’, ‘promis’, ‘brother’, ‘destroy’, ‘wrong’, etc… Possibly indicating that this topic is related to Cain and Abel, something that is mentioned many times in the OT and NT, but only once in the Quran (according to the dataset).

3. Text Classification

This section deals with the task of text classification. The objective of this task is to train a classifier that should be able to assign a class/corpus for a given document. For example, given a verse, to determine if it fits more in the OT, NT or Quran. To create this model, the combined data from all of the corpuses is fed into a Support Vector Machine (SVM). This section will focus on the improvement of a baseline model by using different techniques discussed during the TTDS course.

The baseline model consists of a SVM classifier with a rbf kernel and a regularization parameter of one. For reproducability and consistence, all of the experiments will use a random_state of 25 and a train-val split of 90%-10%.

Table 5: Results for baseline model

Set	Corpus	Precision	Recall	F1-score	Support
Training	Quran	1.00	1.00	1.00	5034
	NT	1.00	1.00	1.00	6434
	OT	1.00	1.00	1.00	18676
	Macro avg	1.00	1.00	1.00	30144
	Weighted avg	1.00	1.00	1.00	30144
Validation	Quran	0.93	0.91	0.92	582
	NT	0.91	0.84	0.87	678
	OT	0.94	0.97	0.95	2090
	Macro avg	0.93	0.91	0.92	3350
	Weighted avg	0.93	0.93	0.93	3350
Test	Quran	0.94	0.92	0.93	620
	NT	0.89	0.86	0.87	844
	OT	0.94	0.96	0.95	2379
	Macro avg	0.92	0.91	0.92	3843
	Weighted avg	0.93	0.93	0.93	3843

3.1. Classification

The following are examples of misslabeled predictions from the validation set:

- ['So', 'the', 'evening', 'and', 'the', 'morning', 'were', 'the', 'fifth', 'day'] - labeled as NT but true label is OT
- ['And', 'you', 'know', 'that', 'with', 'all', 'my', 'might', 'I', 'have', 'served', 'your', 'father'] - labeled as OT but true label is NT
- ['So', 'Moses', 'said', 'You', 'have', 'spoken', 'well', 'I', 'will', 'never', 'see', 'your', 'face', 'again'] - labeled as OT but true label is NT

Clearly, the classifier is having a hard time distinguishing between verses from the OT and NT. This is expected since both books are very similar, but the main reason is because there is much more training data corresponding to OT verses than to NT verses, so, when a NT verse appears, sometimes, the model thinks that it belongs to the OT rather than to the NT.

3.2. Improving the baseline model

The first observation that I had about the baseline model is that it was taking too long to train, around 6 minutes. My first step was to try to reduce the training time while mantaining a decent performance on the validation set. Hence, I trained a linear SVM with the same regularization as the baseline (C=1000), and used TF-IDF (with unigrams and bigrams) instead of the counts of words. Also, I normalized the values of the TF-IDF to the range 0-1. By doing this, I achieved a Macro average F1-score of 1 and 0.93 in the training and validation sets, respectively, in only 9 seconds.

What about the problem of classifying NT verses as OT? What if we add more NT verses to the corpus used to train the model? Lets duplicate the whole NT corpus but replace each word on each document by its synonym (if there is one); to do this, I used wordnet from NLTK. By doing this, with C=1, the training and validation Macro avg F1-scores on the training and validation set is 1 and 0.97, respectively, in only 1 second (due to C=1).

In summary, I used a linear SVM with C=1, TF-IDF with bigrams and unigrams, data normalization, and data augmentation for NT verses by using synonyms. The train time was reduced from ~6 minutes to ~1 second. Table 6 summarizes the results obtained.

Table 6: Results for improved model (training scores are the same as baseline, ignored due to space)

Set	**Corpus**	**Precision**	**Recall**	**F1-score**	**Support**
Validation	Quran	0.99	0.94	0.96	556
	NT	0.96	0.94	0.95	1395
	OT	0.95	0.98	0.97	2110
	Macro avg	0.97	0.95	0.96	4061
	Weighted avg	0.96	0.96	0.96	4061
Test	Quran	0.98	0.92	0.95	620
	NT	0.92	0.86	0.89	844
	OT	0.94	0.98	0.96	2379
	Macro avg	0.95	0.92	0.93	3843
	Weighted avg	0.94	0.94	0.94	3843

While the performance on the test set is not as good as in the validation set (small difference), bear in mind that I’m using a linear SVM and reduced train time from 6 min to 1 second. Also, I tried stemming, lemmatization, stopword removal and dimensionality reduction using MI/X^2/PCA. But in all the cases, the results were either equal or less than the baseline model.