# Text Technologies for Data Science Assignment 2 - Report

s1813674

5th December 2021

## 1 Introduction

### 1.1 Tasks completed

#### 1.1.1 IR-Evaluation for 6 Systems

Designed an Eval class that takes care of the following tasks:

- Finds P@10: precision at cutoff 10 (only top 10 retrieved documents in the list are considered for each query), R@50: recall at cutoff 50, R-precision, Average Precision, nDCG@10: normalized discount cumulative gain at cutoff 10 and nDCG@20: normalized discount cumulative gain at cutoff 20.
- Outputs the ir_eval.csv file that includes the evaluation results for the 6 systems, labelled with their system numbers
- Runs a 2-tailed p-test on all systems and metrics.

#### 1.1.2 Text Analysis

Ran a text anlysis on a corpora which contain verses from the Quran and the Bible (split into Old and New Testaments) by:

- Preprocessing the data, including tokenization, stopword removal and stemming
- Computing the Mutual Information and $\chi^2$ scores for all tokens (after preprocessing) for each of the three corpora and generating a ranked list of the results, in the format token,score.
- Running LDA on the entire set of verses from all corpora together (setting k=20 topics) As an additionnal step for Text Analysis I decided I would run LDA multiple times to look for a possible pattern and see if any topics appear to be common in 2 corpora but not in the other corpus.

#### 1.1.3 Text Classification

Applied text classification using the same collection used in the previous section as training data: given a verse, predict which corpus that verse belongs to.

- Extracted BOW features and trained an SVM classifier with c=1000 to predict the label.
- Computed the precision, recall, and f1-score for each of the 3 classes, as well as the macro-averaged precision, recall, and f1-score across all three classes.
- Tried to improve the results of the classifier and outputted scores for both baseline and improved models in the classification.csv file

### 1.2 Things I learned and Challenges

#### 1.2.1 IR Evaluation

Working on IR Evaluation made me realize the importance of having different ways to evaluate systems. As I will describe later on, some results might suggest that a system is way better but evaluating the significance of the performance difference between the systems can quickly determine that it's not necessarily the case. For this part, the challenge was to make the code reusable so there's not too much hard-coding and in order to have efficient reusable code.

### 1.2.2 Text Analysis

This section taught me a lot about LDA models and ways to compare corpora which I found very insightful. The struggle here once again is trying to have code that doesn't take forever to run since we have to go through each verse multiple times. Finding the right data structures to store some information that will be needed later on in the code was an interesting challenge.

### 1.2.3 Text Classification

For this section, I learned more about SVMs but the challenge was to find ways to improve the model. As I will describle in the report, a lot of the things I thought would improve the model ended up doing the contrary.

## 2 IR Evaluation

For the IR Evaluation we use different scores (P@10, R@50, R-precision, AP, nDCG@10 and nDCG@20) on all 6 systems. To do a first comparison on those systems, we will start by comparing the means for each metric and system as illustrated in the table below.

| | P@10 | R@50 | R-Precision | AP | nDCG@10 | nDCG@20 |
|---|---|---|---|---|---|---|
| System 1 | 0.390 | 0.834 | 0.401 | 0.400 | 0.363 | 0.485 |
| System 2 | 0.220 | 0.867 | 0.253 | 0.300 | 0.200 | 0.246 |
| System 3 | 0.410 | 0.767 | 0.448 | 0.451 | 0.420 | 0.511 |
| System 4 | 0.080 | 0.189 | 0.049 | 0.075 | 0.069 | 0.076 |
| System 5 | 0.410 | 0.767 | 0.358 | 0.364 | 0.332 | 0.424 |
| System 6 | 0.410 | 0.767 | 0.448 | 0.445 | 0.400 | 0.491 |

Table 1: Mean of P@10, R@50, R-Precision, AP, nDCG@10 and nDCG@20 scores for each evaluated system

From this table we can extract the highest scoring system per metric and second highest as illustrated in this second table below:

| | P@10 | R@50 | R-Precision | AP | nDCG@10 | nDCG@20 |
|---|---|---|---|---|---|---|
| Highest Scoring System | 3, 5 and 6 | 2 | 3 and 6 | 3 | 3 | 3 |
| Second Highest | 1 | 1 | 1 | 6 | 6 | 6 |

Table 2: Highest and Second Highest Scoring System per Metric

According to these results we could think that system 3 is the best system as it appears to be the highest scoring system for 5 out of the 6 metrics found. However, we want to find out if there is a significant difference between the highest scoring system and second highest. For this, we conducted a 2-tailed t-test on the systems with a significance level of 0.05. The objective is to compare, for each metric, the sample mean (highest system) with the population mean (second highest system). Our null hypothesis here is that the means are equal.

| | P@10 | R@50 | R-Precision | AP | nDCG@10 | nDCG@20 |
|---|---|---|---|---|---|---|
| Systems compared (Sample, Population) | (3, 1) (5, 1) (6, 1) | (2, 1) | (3, 1) | (3, 6) | (3, 6) | (3, 6) |
| p-value | 0.888 | 0.703 | 0.759 | 0.967 | 0.882 | 0.869 |

Table 3: Results of Two-Tailed P-Test for each Metric

The table above shows the results of the conducted two tailed p-test. Because our significance level is 0.05, the p-values are only significant if:

$$p_{value} < 0.5 \times 0.05 = 0.025 \text{ or } p_{value} > 1 - 0.5 \times (0.05) = 0.975$$

The table shows that none of the p-values obtained fit the criteria. Hence, we cannot reject the null hypothesis as there is not enough evidence showing that the second best system is significantly different to the first system. This however suggests that system 6 might also be just as good as system 3 since it also appears in 5 of the 6 metrics from Table 2.

## 3 Text Analysis

### 3.1 Token Analysis

For the Text Analysis Section, we consider the Quran, New Testament, and Old Testament each to be a separate corpus, and their verses as individual documents. After preprocessing (Stemming, Stopping, CaseFolding and Tokenization), we computed different scores starting with Mutual Information and Chi Squared $\chi^2$. For these metrics we used one corpus as the target corpus and compared it the two other corpora together. We've used the formulae given in the slides to compute those scores. The Mutual Information I(X,Y) tells us how much we can learn about Y by observing X where Y is the target corpus and X is the 2 other corpora concatenated. We use the fomula below:

$$I(U;C) = \frac{N_{11}}{N}\log_2\frac{NN_{11}}{N_1 \cdot N_{.1}} + \frac{N_{01}}{N}\log_2\frac{NN_{01}}{N_0 N_{.1}} + \frac{N_{10}}{N}\log_2\frac{NN_{10}}{N_1.N_{.0}} + \frac{N_{00}}{N}\log_2\frac{NN_{00}}{N_0.N_{.0}}$$

Where $N_{11}$ is the number of documents a term appears in for the target corpus, $N_{10}$ is the number of documents this term appears in for the other corpora combined, $N_{01}$ is the number of documents in the target corpus minus$N_{11}$ and $N_{00}$ is the number of documents in the combined corpora minus $N_{10}$.
The Chi Squared score, is a hypothesis testing approach where the null hypothesis is that a term's appearance is independent from the document's class. To compute it, we use the following formula:

$$X^2(D,t,c) = \frac{(N_{11}+N_{10}+N_{01}+N_{00}) \times (N_{11}N_{00} - N_{10}N_{01})^2}{(N_{11}+N_{01}) \times (N_{11}+N_{10}) \times (N_{10}+N_{00}) \times (N_{01}+N_{00})}$$

Where $N_{11}$, $N_{10}$, $N_{01}$ and $N_{00}$ have the same values as the ones stated above.

| Old Testament | | | | New Testament | | | | Quran | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Mutual Information | | Chi Squared | | Mutual Information | | Chi Squared | | Mutual Information | | Chi Squared | |
| Term | Score | Term | Score | Term | Score | Term | Score | Term | Score | Term | Score |
| jesus | 0.0372 | jesus | 1464.155 | jesus | 0.0524 | jesus | 3026.683 | god | 0.0314 | muhammad | 1852.132 |
| israel | 0.0311 | lord | 1122.996 | christ | 0.0319 | christ | 1772.096 | muhammad | 0.0288 | god | 1710.674 |
| king | 0.0261 | israel | 1095.935 | lord | 0.0221 | lord | 874.182 | believ | 0.0197 | believ | 1344.233 |
| lord | 0.0259 | king | 946.774 | discipl | 0.0143 | discipl | 819.833 | torment | 0.0196 | torment | 1332.812 |
| christ | 0.0199 | god | 788.884 | israel | 0.0138 | paul | 528.992 | messeng | 0.0156 | messeng | 1062.254 |
| believ | 0.0180 | christ | 779.407 | peopl | 0.0099 | peter | 528.992 | revel | 0.0139 | revel | 941.556 |
| god | 0.0165 | believ | 750.066 | king | 0.0099 | thing | 503.197 | king | 0.0131 | unbeliev | 848.865 |
| muhammad | 0.0155 | muhammad | 608.738 | paul | 0.0095 | israel | 451.222 | israel | 0.0130 | guidanc | 810.932 |
| son | 0.0129 | faith | 543.938 | peter | 0.0095 | spirit | 436.662 | unbeliev | 0.0125 | disbeliev | 786.117 |
| torment | 0.0127 | son | 538.667 | land | 0.0095 | john | 408.176 | guidanc | 0.0122 | unjust | 765.187 |

Table 4: Mutual Information and $\chi^2$ scores of the 10 highest scoring terms for each corpus

The table above shows the 10 highest scoring terms for each metric and for each corpus. Firstly, it is important that we mention why the value scale between $\chi^2$ and MI are different: Mutual Information is a sum of joint probabilities so is bounded to a lower number than $\chi^2$ that is evaluating raw counts. We could convert the $\chi^2$ to p-values but for our use we do not need to do that and can simply compare with the mutual information scores as it is.

We can notice from the table below that both methods retrieve very similar words. In the table we've highlighted in blue cells the terms that come up for both $\chi^2$ and MI in each corpus. This is to be expected as the two quantities are reparametrizations of each other. We can also notice a lot of common words in the Old Testament and New Testament columns which could be explained with the fact that both corpora make up the Holy Bible.

As expected, "Jesus" Is the top scoring terms for all metrics in both the Old and New Testament whereas God (swt) and Muhammad (pbuh) are the top terms for the Quran. Surprisingly, these words also appear in the Old testament. However, they appear with a low Chi Squared and Mutual Information score suggesting that the terms are not closely related to the corpus.

We can observe that the new testament and Quran have more outliers, i.e elements that appear in one metric but not in the other (respectively 6 and 4), than the Old testament that only has one. This makes sense as the Old testament is the largest corpus available (over 20 000 verses against 7000+ verses for the New testament and 5000+ verses for the Quran: as the sample sizes increase, $\chi^2$ and MI will tend to give similar results.

Overall, the top 10 words for the corpora all have the same religious connotation and are very similar probably due to the fact that they're all texts from Abrahamic religions.

## 3.2 Topic Analysis

For this section, we have ran LDA on the entire set of verses (not stemmed) from all corpora together. We've extracted the top topic for each corpus as well as the top 10 tokens with highest probability of belonging to that topic. Note that, unless we set a random state, each run will give different results. The table below displays the output of one run:

| | Topic | Label | Top 10 tokens |
|---|---|---|---|
| Old Testament | **5** | Moses and the tribes of Israel | lord * 0.132 + god * 0.079 + moses * 0.071 + spirit * 0.059 + children * 0.051 + israel * 0.039 + fact * 0.032 + speak * 0.029 + people * 0.025 + words * 0.023 |
| New Testament | **9** | Soul Righteousness | god * 0.073 + good * 0.054 + lord * 0.053 + evil * 0.040 + reward * 0.037 + word * 0.036 + created * 0.034 + knowledge * 0.033 + soul * 0.025 + find * 0.025 |
| Quran | **7** | Faithfulness And Sincerity | god * 0.344 + lord * 0.046 + people * 0.046 + fear * 0.040 + peace * 0.021 + wise * 0.020 + living * 0.016 + hear * 0.016 + hearts * 0.014 + righteousness * 0.012 |

Table 5: Top topic for each corpus and top 10 tokens with highest probability of belonging to that topic

This run didn't show a common topic. However, I've decided to run LDA multiple times an analyze the result to see if there's a pattern.

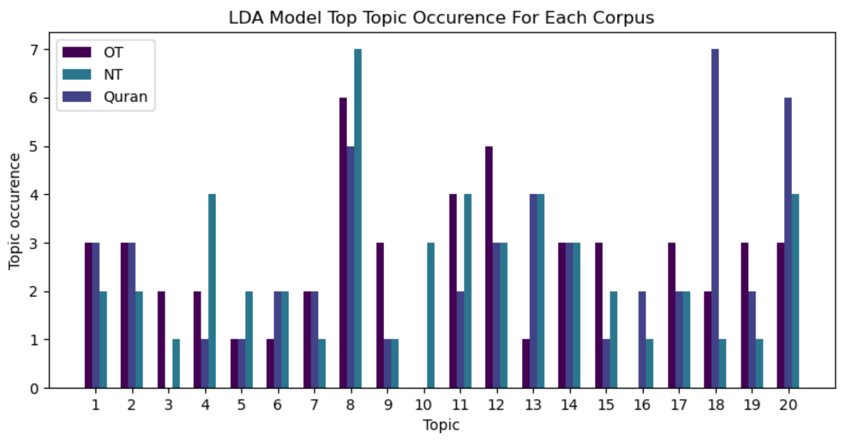

Figure 1: Bar Plot showing the number of occurrences of top topics (indexed starting at 1) for each corpus

After running the LDA multiple times, we have more significant results as shown in the bar plot above. Some topics appear mainly in one corpus and not in the others : Topic 18 appears mostly in Quran, Topic 10 appears only in the New Testament and topic 9 appears mostly in the Old Testament. We can also notice that some topics appear in all corpora equally like topic 14 with top words: 0.103*"son" + 0.077*"time" + 0.075*"created" + 0.051*"god" + 0.050*"evidence" + 0.046*"peace" + 0.030*"drink" + 0.023*"save" + 0.018*"garden" + 0.018*"lord". This makes sense as they're all Abrahamic religion texts.

One other interesting to mention is that the New Testament and Quran seem to be sharing top topics quite often. In multiple runs, they had the same top topic and it differed from the Old Testament. Example topics are topic 12, topic 13, topic 14, topic 9 and topic 17. This can be explained by looking back at Table 5. While the topic IDs differ, the theme seems to be similar when we look at top 10 tokens and when we read the verses. Both seem to be talking about righteousness, soul and moral high ground.

Some other topics are mainly shared between Old testament and Quran like topic 1, topic 2 and topic 7. If we take topic's 2 top tokens: 0.060*"land" + 0.049*"live" + 0.045*"follow" + 0.037*"people" + 0.034*"pharaoh" + 0.031*"turned" + 0.027*"hand" + 0.025*"sea" + 0.024*"pray" + 0.021*"back" we can notice words like pharaoh which explain it as pharaohs are mentionned in the OT (Exodus) and in the Quran.

Finally, and as expected, some topics are mostly shared between OT and NT as they are what constitutes the Holy Bible. For example we can cite topic 11, topic 15 and topic 3 with top words including gentiles (non-Jews), priests and dwelling which explain that they're shared between OT and NT but not with Quran as they're more related to Judaism and Christianity than Islam.

# 4 Text Classification

## 4.1 Baseline Model

For this part, we apply text classification using the same collection used in the previous section as our training data. The goal is: given a verse, predict which corpus that verse belongs to. To do that we follow the steps below

- We start by shuffling the order of the data and splitting the dataset into a training set and a separate development set. We choose to have 90% for training and 10% for development.
- We then preprocess the corpora but without stopping and stemming. We only apply case-folding and tokenization.
- We create an ID mapping where each unique term is mapped to an integer. We also make sure to maintain consistency across the sets. Meaning a word in the training set will have the same mapping in the test set. New words that appear in the test set are mapped to other IDs so that the classifier can handle out of vocabulary terms.
- We create an identical numberic mapping for the categories such that each corpus is mapped to an integer ID.
- We then create a count matrix where each row is a document and each column corresponds to a word in the vocabulary. The value of element (i,j) is the count of the number of times the word j appears in document i. To do that we use a Sparse Matrix because a lot of entries will be 0. So to optimize on storage it is better to use a Sparse Matrix.
- For each set we will obtain an X and a y. Where X is the Sparse Matrix and each element of y is the correct classification for the given verse.
- Lastly, we train a classifier and test the performance. For this task, we use an SVM multiclass Classifier that we first instantiate with c=1000. This gives us our baseline model. We fit the model on the training data only.

| system | split | p-quran | r-quran | f-quran | p-ot | r-ot | f-ot | p-nt | r-nt | f-nt | p-macro | r-macro | f-macro |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| baseline | train | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 |
| baseline | dev | 0.913 | 0.906 | 0.909 | 0.931 | 0.955 | 0.943 | 0.884 | 0.821 | 0.851 | 0.909 | 0.894 | 0.901 |
| baseline | test | 0.918 | 0.890 | 0.904 | 0.927 | 0.952 | 0.939 | 0.881 | 0.834 | 0.857 | 0.909 | 0.892 | 0.900 |

Table 6: Baseline Model: Precision, recall, and f1-score for each of the 3 classes, as well as the macro-averaged precision, recall, and f1-score across all three classes for training, development and testing set

The table above shows the performance scores for our basline model on the training, development and test set. The scores are realtively high for all metrics with the lowest values encountered for the New Testament.

As an example, we can list 3 instances from the development set that the baseline system labels incorrectly. We can also hypothesize that words with similar counts in both the actual corpus and the predicted corpus, , words that do not appear in the training set and included stop words (that appear often in all the corpora) are the reason why some verses get misclassified. I ran a few experiments to confirm or reject the hypothesis:

The table above highlights the fact that theses misclassified verses contain mostly stop words (67% of the verse is made of stop words) and the rest of the words have similar counts (If we take into account the corpora sizes) which will make it hard for the classifier to predict a correct output.

Another interesting thing in this example is that the second verse "So I swore in my wrath they shall not enter my rest" is present in both the New testament and the Old Testament which makes it hard to classify especially if in the training set the verse from OT is present. Hence we can confirm our hypothesis.

| Acutal Label | Predicted Label | Verses | Stop-words count & percentage | Non stop-words counts per corpus |
|---|---|---|---|---|
| OT | NT | ['yet', 'i', 'will', 'also', 'make', 'a', 'nation', 'of', 'the', 'son', 'of', 'the', 'bondwoman', 'because', 'he', 'is', 'your', 'seed'] | 12 out of 18<br>67 % | make: OT: 827, NT: 145, Quran: 197<br>nation: OT: 436, NT: 59, Quran: 42<br>son: OT: 2178, NT: 343, Quran: 99<br>bondwoman: OT: 2, NT: 4, Quran: 0<br>seed: OT: 56, NT: 44, Quran: 7 |
| NT | OT | ['so', 'i', 'swore', 'in', 'my', 'wrath', 'they', 'shall', 'not', 'enter', 'my', 'rest'] | 8 out of 12<br>67% | swore: OT: 64, NT: 5, Quran: 3<br>wrath: OT: 141, NT: 41, Quran: 28<br>enter: OT: 134, NT: 122, Quran: 59<br>rest: OT: 279, NT: 51, Quran: 34 |
| NT | Quran | ['pursue', 'peace', 'with', 'all', 'people', 'and', 'holiness', 'without', 'which', 'no', 'one', 'will', 'see', 'the', 'lord'] | 10 out of 15<br>67% | pursue: OT: 83, NT: 9, Quran: 8<br>peace: OT: 262, NT: 81, Quran: 58<br>holiness: OT: 371, NT: 155, Quran: 12<br>lord: OT: 5473, NT: 568, Quran: 846 |

Table 7: Results of Experiments Ran on Misclassified Verses

## 4.2 Improved Model

After getting these results the goal was to improve the model and get higher scores across all metrics. Note that during experiments and tuning, a random state was used to ensure reproducible results however, the scores displayed here do not use a fixed random state. To optimize and improve the model I had multiple ideas some of which worked and some that didn't:

- Pre-process the text (stopping, stemming, stopping and stemming): This made the scores decrease and didn't help improve the model so I kept the tokenizing and case folding only pre-processing instead.
- Use TF-IDF instead of count: This approach also made the scores decrease and didn't help improve the model.
- I decided to try multiple c values and noticed that decreasing the c value improved the scores. So, I lowered to c=40
- I decided to add TF-IDF as a new feature instead of completely getting rid of the count and using TF-IDF instead. This also improved the final scores.

| system | split | p-quran | r-quran | f-quran | p-ot | r-ot | f-ot | p-nt | r-nt | f-nt | p-macro | r-macro | f-macro |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| improved | train | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 |
| improved | dev | 0.914 | 0.915 | 0.914 | 0.932 | 0.965 | 0.948 | 0.908 | 0.812 | 0.857 | 0.918 | 0.897 | 0.907 |
| improved | test | 0.921 | 0.906 | 0.914 | 0.929 | 0.958 | 0.943 | 0.904 | 0.833 | 0.867 | 0.918 | 0.899 | 0.908 |

Table 8: Improved Model: Precision, recall, and f1-score for each of the 3 classes, as well as the macro-averaged precision, recall, and f1-score across all three classes for training, development and testing set

Overall, as shown in the table above we've accomplished some improvement that can be seen by comparing the scores in the test set for both the baseline and the improved model. However, more generally, we can see an increase of +0.009 in macro precision, +0.007 in macro recall and +0.008 in macro f1.

We could probably increase the scores even more with an even lower C. We could also consider using a completely different model like Convolutional Neural Network (CNN) that are known to outperform Support Vector Machines.