TTDS Coursework2

s2020314

1 General Information

I have implemented all required modules from scratch, except for using an existing tool for LDA-based topic modeling. The code can be divided into three main modules — **EVAL** for IR evaluation, **ANALYSIS** for text analysis and **CLASSIFICATION** (including **CLASSIFICATION_BASELINE**, **NEWANALYSIS** and **CLASSIFICATION_IMPROVED**) for document(verse) classification.

In the **EVAL** module, I used six required evaluation methods (**P_10** (precision at cutoff 10), **R_50** (recall at cutoff 50), **R_precision**, **AP** (average precision) and **nDCG_k** (normalized discount cumulative gain at cutoff k of 10 or 20)) for system evaluation based on their retrieved results. **evaluation_combo()** leads to each one of them. During the implementation, I feel enlightened for that the process enhanced my understanding of their theory, especially **nDCG_k**, which I'm not familiar with and had a struggle about its calculation.

In the **ANALYSIS** module, I calculated the **MI** (mutual information) and χ^2 score for all tokens under each corpus, outputting three files corresponding to each corpus and its list of score-sorted token and score pair. I used **LatentDirichletAllocation** under sklearn.decomposition to run LDA on the entire set of verses (including all three corpuses). The most challenging part here is the use of LDA. I used a sparse matrix — **csr_matrix** to save space for corpus-term appearance table, which I have never used before. After getting the result of best topic and its corresponding score, I output a file of topic-related most likely terms and their probability scores for all three topics (each for a corpus). I struggled in **what's the actual topic_term_probability_matrix** (probability score for each topic-term pair). The problem is — is it the probability score for each term under each topic or for each topic under each term. I finalized that it should be for each topic under each term, which means the summation of probabilities for a term under all topics should be one. It's only then when we could say to decide the topic by observing a term.

In the **CLASSIFICATION** module, I split the data into 90% for training and 10% for evaluation. In the baseline model, I only preprocess with tokenization. All steps of **BOW extraction** and **training SVM classifier** are strictly in line with the instructions under lab7 and assignment2. I set random seed of my student number (2020314) in dataset shuffling and SVM model to make all my experiments repeatable and comparable. To improve the model, I took action in **preprocessing**, **feature selection** and **SVM parameters selection**. I compared preprocessing with or without stemming and stopping. I utilized the mi score output in task2 to choose the top N features and its best N. I changed parameters in SVM like C, decision_function_shape, gamma and tol. What's troublesome is using mi score-related feature selection especially when I want to keep the preprocessing simple, which means mi score should be recalculated to keep the preprocessing before mi score simple, so to be consistent.

In a nutshell, I mostly made the implementation from scratch. This assignment enriches me with concrete theory, practices and critical thinking.

2 IR Evaluation

Evaluation Method	Best System	Score	Runner-up	Score	Pvalue
P@10	3, 5, 6	0.410	1	0.390	0.888
R@50	2	0.867	1	0.834	0.703
R-precision	3, 6	0.448	1	0.401	0.759
AP	3	0.451	6	0.445	0.967
nDCG@10	3	0.420	6	0.400	0.882
nDCG@20	3	0.511	6	0.491	0.868

Table 1: The best performing IR system and runner-up for each score.

The best systems and its runner-up under each score is listed in Table 1. The ranking is based on the average scores achieved for each system. For all systems achieving the same highest average score, we identify them the best regardless.

Now we use 2-tailed t-test to decide whether the best system is statistically significantly better than the runner-up under each score (query level). The p-value threshold is set as 0.05. All corresponding p-values calculated are also listed in Table 1. Notably, in the tie of best systems in P@10 and R-precision, the scores for each query are mostly identical. Therefore, we consider best systems in a tie not significantly different and calculate the t-test value between the runner-up and the first best system. It is seen that the p-values under R@50 and R-precision are around 0.75 and all others are around 0.9. The test measures whether the average (expected) value differs significantly across samples. In observing high p-values larger than 0.05, we cannot reject the null hypothesis of identical average scores, meaning that the best system is not statistically significantly better than the second system.

Interestingly, we observe that system 3 is almost in every best system set, while system 6 is the second best purely by observing its appearance in the best and runner-up list. I'd like to add another t-test (score level) to check their performance. The p-value calculated is 0.924, which shows that system 3 is not significantly better than system 6 in a general level.

3 Text Analysis

3.1 Token Analysis

The top ten highest scoring words and their score by MI and χ^2 in three corpus is listed in the Table 2 (Quran), Table 3 (OT) and Table 4 (NT).

Between the rankings produced by the two methods, I have several observations. Firstly, several words are consistent in their prominent place in ranking. For example, in Quran, word 'allah', 'thou' and 'god' ranks high in top 10. In OT and NT, the intersection is even more obvious. Along with this consistency, there exists novel word high up in the ranking but not visible in the other. For example, word 'believ' in Quran. Conventionally we expect to see word 'jesus' high in OT and NT rankings. However, it dominates in NT but not OT. In OT, the first word is 'allah', which does not occur in the corpus but impose a big negative support. The word 'allah' also appears high in the MI ranking of NT but not in the top ten of χ^2 in NT. This difference may indicate an overall preference that χ^2 weighs positive supporting words more.

MI_WORD	MI_SCORE	CHI_WORD	CHI_SCORE
ALLAH	0.153192	ALLAH	7058.784144
THOU	0.039320	PUNISH	917.836598
THI	0.031260	THOU	889.245457
YE	0.028490	BELIEV	856.011939
THEE	0.028215	UNBELIEV	811.821543
GOD	0.025082	MESSENG	769.740994
MAN	0.019547	GOD	704.641573
KING	0.019299	THI	699.436193
HATH	0.019037	BELI	683.328190
PUNISH	0.018013	GUID	677.282408

Table 2: The top 10 highest scoring words for MI and Chi in Quran.

MI_WORD	MI_SCORE	CHI_WORD	CHI_SCORE
ALLAH	0.087108	ALLAH	2778.575055
JESUS	0.040865	JESUS	1296.972790
ISRAEL	0.036133	LORD	1119.329010
LORD	0.031195	ISRAEL	1070.162576
THI	0.029506	THI	953.891134
KING	0.029204	KING	884.373899
THOU	0.022686	THOU	776.968632
CHRIST	0.020500	CHRIST	649.053579
THEE	0.018858	THEE	633.996601
BELIEV	0.017335	BELIEV	600.444336

Table 3: The top 10 highest scoring words for MI and Chi in OT.

MI_WORD	MI_SCORE	CHI_WORD	CHI_SCORE
JESUS	0.064584	JESUS	3268.988777
CHRIST	0.036766	CHRIST	1795.001002
ALLAH	0.019346	DISCIPL	909.799816
DISCIPL	0.018020	FAITH	669.145461
LORD	0.016077	PAUL	588.945044
YE	0.013014	YE	586.429244
ISRAEL	0.012860	PETER	560.751035
FAITH	0.012670	LORD	538.896370
PAUL	0.011853	THING	525.049576
PETER	0.011449	RECEIV	490.808610

Table 4: The top 10 highest scoring words for MI and Chi in NT.

From those rankings and my intuition, I learn that if a word 'allah' appears, it's strongly suggested that it could be corpus Quran but not OT. The god in Quran is 'allah' and 'jesus' in OT and NT. Their relevant words like 'israel' for Quran and 'peter', 'paul', 'christ' for OT and NT are strong supports in distinguishment. In Quran, we hardly refer to people with 'thou' and 'thi'. With these words appearing high in rank, it indicates that the nominative form greatly helps in distinguishing three corpuses. The word 'punish' high in Quran indicates a strong sense of law enforcement and a rip-what-you-sow idea. The word 'hate', 'discipl' and 'believ' indicate emotional guidance and self-restriction in OT and NT.

3.2 Topic Analysis

The top 10 tokens and their probability scores for each of the three topics that are identified as being most associated with each corpus in noted and shown in Table 5.

My own labels for each topic are: 1) for topic 16 (Quran): Arabian & kindness, 2) for topic 14 (OT): love & hope, 3) for topic 11 (NT): Christian & mercy.

Quran_word	Quran_score	OT_word	OT_score	NT_word	NT_score
harm	0.985	goest	0.979	preach	0.992
benefit	0.977	imagin	0.969	endureth	0.982
term	0.977	precept	0.963	elia	0.961
secur	0.973	prolong	0.960	whereunto	0.960
jinn	0.972	lovingkind	0.960	spiritu	0.958
stern	0.967	medit	0.954	christ	0.955
alik	0.967	speakest	0.940	abound	0.952
sorcer	0.966	trusteth	0.940	offenc	0.950
orphan	0.958	degrad	0.926	baptism	0.949
thamood	0.957	communic	0.926	gooddoer	0.947

Table 5: The top 10 tokens and their probability scores for the best topic in Quran, OT and NT, which is labelled as topic 16 with the highest average score of 0.207, topic 14 with the highest average score of 0.113, and topic 11 with the highest average score of 0.228 respectively.

LDA can divide the topic and assign a best topic for each document. For a topic, a set of words is given with its probability to be in that topic. Therefore, by observing the top set of words in the best topic for a corpus, we can deduce what's expected in the corpus. Let's consider the examples given in Table 5. The topic I labelled as Arabian & kindness is due to the Arabian words visible like 'thamood' and 'jinn' and reference of 'harm', 'secur' and 'orphan'. It tells about one feature of Quran that it's related to Arabian and it's call for kindness.

Quran_topic	Quran_score	OT_topic	OT_score	NT_topic	NT_score
16	0.207	14	0.113	11	0.228
18	0.204	9	0.081	4	0.136
13	0.110	12	0.080	2	0.105
0	0.073	6	0.078	9	0.055
7	0.054	1	0.066	14	0.052

Table 6: The top 5 topics and their scores for Quran, OT and NT.

Table 6 is a list of topic ranking (top 5) and scores for Quran, OT and NT. There exist topic 14 and 9 that appear to be common in OT and NT but not in Quran. To take a closer look, Table 7 shows the top ten words for each topic. I formerly labelled topic 14 as love & hope which is both acceptable for OT and NT. Topic 9 contains words like 'didst', 'makest', 'lovest', 'comest', etc. to represent an archaic way of verses, which is both acceptable in OT and NT. The main difference of the results given by LDA and MI or χ^2 is that the category is ranked in a topic level. It's not single words that either positively or negatively decide the category. All topics can be combined, introducing a fine-grained results. It seems that for each topic, the words are always positively supportive for deciding its category.

TOPIC14_WORD	TOPIC14_SCORE	TOPIC9_TOPIC	TOPIC9_SCOR
GOEST	0.979	VINEYARD	0.986
IMAGIN	0.969	DIDST	0.97
PRECEPT	0.963	STORE	0.96
PROLONG	0.96	MAKEST	0.949
LOVINGKIND	0.96	THRICE	0.937
MEDIT	0.955	COCK	0.921
SPEAKEST	0.941	LOVEST	0.921
TRUSTETH	0.941	COMEST	0.915
DEGRAD	0.927	MADEST	0.914
COMMUNIC	0.927	CURSETH	0.914

Table 7: The top 10 tokens and their probability scores for topic 14 and 9.

4 Classification

SYSTEM	SPLIT	P-MACRO	R-MACRO	F-MACRO
BASELINE	TRAIN	1.000	1.000	1.000
BASELINE	DEV	0.934	0.913	0.922
BASELINE	TEST	0.931	0.910	0.918
IMPROVED	TRAIN	0.998	0.997	0.997
IMPROVED	DEV	0.936	0.924	0.930
IMPROVED	TEST	0.933	0.920	0.927

Table 8: The macro precision, macro recall and macro f1 in training, development and test set under baseline and improved model.

Table 8 shows the performance of each set under baseline model and improved model. It is shown that the improved model has achieved certain improvement over the baseline model. In the development set, the macro-f1 improves 0.8%. In the test set, it's glad to see an improvement of 0.9%. I will illustrate the flow of classification experiment in the following paragraphs, which includes problem identification in baseline and how I improved the model.

I picked the first **three wrong labeling** in the baseline model. The document unique id (it is assigned as the order of verses in the input file) are 23066, 1419 and 23145. The tokens of document 23066 are full of simple and less identifiable words, which we can not help. There are vast appearances of numbers like '2' and stopwords like 'and' and 'but'. Both document 23066 and 23145 are NT wrongly labelled as OT, which is understandable due to the use of simple words. Document 1419 contains more meaningful tokens like 'heaven', 'endure', 'wills', etc. However, it lacks more determining words like Arabian expression. It can cause the mislabelling.

To **improve the model**, I tried three methods — 1) add stemming and word stopping in preprocessing. 2) use mi scored word list to reduce features. 3) adjust model parameters. The final best model is the one without using stemming and word stopping in preprocessing. It uses mi score to reduce features. The reduction scale is set as the 3000 top words for each corpus. To clarify, it means that if a word in the second corpus has appeared in the former corpus, I would like to skip it for the current corpus to ensure that all corpuses have at least 3000 words that are uniquely important for the corpus. The model parameters should be C=50 and gamma=0.007. It is shown that changing decision_function_shape to ovo and changing tol bears almost no effect in my case.

To make the experiments reproducible and comparable, I set a random seed for shuffling the training and development set and for the SVM model, which is set as my student number. Let's call the baseline model V_BASE. To make my model improve for all three corpuses, I observe the both the weak classification, which is distinguishing OT and NT in this case, and the overall score, which is a simple average of all scores over all corpuses. I will show the overall score for important comparisons. To begin with, the score is 0.923 for V_BASE.

V_PRE++ is a V_BASE adding stemming and word stopping. Understandably, all the scores dropped and the score is 0.862. It could be due to the large selection of stopping words. I stopped here without further adjusting what words should be stopped.

V_REDUC is a V_BASE adding mi score relevant feature reduction. The N top words are set as 1000, 2000 and 3000. The score are 0.906, 0.917 and 0.919, which is a little lower that V_BASE but looks more promising and reasonable. In order to use a model with reduction but without changing current preprocessing requires me to output another set of files for mi scores which is calculated with the corresponding simple preprocessing. The part is is in the **NEWANALYSIS**.

During changing the model parameters, I reduced C from 1000 to 500, 50 and 25. I set the gamma to 0.001, 0.0025, 0.004, 0.007 and 0.01. I tried to change decision_function_shape to ovo and change tol but it shows no difference. The best combination is C=50, gamma=0.007. V_BEST is a V_REDUC based model with the best model parameter, which gives a overall score of 0.930. Check the final results in Table 8.