Coursework2

S2066387

December 6, 2020

1 Introduction

The coursework is divided into three parts: IR evaluation, token and topic analysis, and text classification.

In IR evaluation, the results of 10 queries in 6 systems from **system_results.csv** are compared with the true relevant results in **qrels.csv** by 6 kinds of IR evaluation measurements: P@10, R@50, r-precision, AP, nDCG@10 and nDCG@20:

- P@10: precision at 10, with the formula P@k = $\frac{\text{\# of relevant documents retrieved @ k}}{\text{\# of retrieved documents @ k}}$
- R@50: recall at 50, with the formula R@k = $\frac{\text{\# of retrieved documents that are relevant @ k}}{\text{\# of relevant documents}}$
- r-precision: precision @ rank r for a query with known r relevant documents
- AP: average precision, with the formula AP = $\frac{1}{r}\sum_{k=1}^{n} P(k) * rel(k)$
  r = # of relevant documents for a given query;
  n = # of documents retrived for a query;
  $P(k)$ = precision @ k;
  $rel(k)$ = 1 if retrieved doc @ k is relevant, 0 otherwise
- nDCG@10, nDCG@20: normalised discount cumulative gain at 10 and 20, with the formula nDCG@k = $\frac{DCG@k}{iDCG@k}$
  $DCG@k = rel_1 + \sum_{i=2}^{k} \frac{rel_i}{log_2(i)}$;
  $iDCG@k$ = ideal discount cumulative gain @ k

By implementing these measurements, I learnt how to inspect an IR system from different angles. Also I learnt how to judge whether the best system is better than others using 2-tailed t-test with the formula $t = \frac{A-B}{\sigma_{(A-B)}} \cdot \sqrt{N}$, where $\sigma_{(A-B)}$ is the standard deviation of $A - B$ and N is the number of samples.

In token and topic analysis, two methods for evaluating token importance namely MI and χ^2 are introduced and implemented on **train_and_dev.tsv** which contains verses from Quran, New Testament (NT) and Old Testament (OT):

- MI: $I(U;C) = \sum_{e_t \in \{1,0\}} \sum_{e_c \in \{1,0\}} P(U = e_t, C = e_c) log_2 \frac{P(U=e_t, C=e_c)}{P(U=e_t)P(C=e_c)}$
  U = document contains term t
  C = class is the target class
- χ^2: $\chi^2(D,t,c) = \sum_{e_t \in \{1,0\}} \sum_{e_c \in \{1,0\}} \frac{(N_{e_t e_c})^2}{E_{e_t e_c}}$

From the table of top ranking words, we can get an intuitive idea of what the corpus is about. Here we see that some words never appear in a corpus but get a high score (such as "alla" in NT), which may be an interesting further research direction. For topic analysis, we focus on distinguishing the topic of each corpus by using filtered LDA model. Ideally we want to get a topic with high frequent and unique words for each corpus. We do get a distinctive topic for each corpus from LDA, but the scores of the words are quite low. Perhaps more specified filter condition could be implemented.

In text classification, the baseline classification model used in the coursework is multi-class Support Vector Machine (SVM). The collection of verses from the three corpora is divided into train an development set. 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 are calculated and the macro-f1 score is used as the measurement of the classifier. Since we get a pretty high macro-f1 score for the development set, the biggest challenge is how to improve the classifier. Several methods are tested such as changing the SVM parameters, changing the learning method, changing the preprocessing and changing feature selection, two of which show improvement for macro-f1 score of development and test set.

2 IR Evaluation

As described in introduction part, several kinds of IR evaluation measures are implemented based on the six different systems, including precision at cutoff 10, recall at cutoff 50, r-precision, average precision, nDCG@10 and nDCG@20. The table below shows the mean scores of each measurement for each system:

Table 1: mean scores for system 1-6

	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

Accordingly, the best systems and second best systems are listed as follows. Assuming that the difference between the scores for the best and second best systems obeys normal distribution, we calculated the p-value of the two-tailed t-test result:

Table 2: best and second best systems with p-value

	P@10	R@50	r-precision	AP	nDCG@10	nDCG@20
best system	S3,S5,S6	S2	S3,S6	S3	S3	S3
second best system	S1	S1	S1	S6	S6	S6
p-value	0.888	0.703	0.759	0.967	0.882	0.869

Notice here for some measures there are more than one best systems, but the results of p-values are the same. We examined the following hypothesis using 2-tailed t-test:

H_0: true difference in means is equal

H_1: true difference in means is not equal

As Table 2 indicates, the p-values are much larger than significance level $\alpha = 0.05$, which means we cannot reject H_0 in any of the six measures, i.e. the best system is not statistically significantly better than the second best system. Evidence can be seen in Table 1 that the differences between mean scores of best and second best system are less than 0.05.

3 Token Analysis

Table 3: Top 10 highest scoring words for MI and χ^2

corpus	MI	χ^2
Quran	allah, thou, thi, ye, thee, god, punish, king, believe, man	allah, punish, believ, thou, messeng, unbeliev, guid, beli, vers, disbeliev
OT	alla, jesus, israel, king, thi, lord, thou, christ, thee, believ	allah, jesus, lord, israel, thi, king, thou, thee, believ, christ
NT	jesus, christ, allah, discipl, lord, faith, ye, israel, paul, peter	jesus, christ, discipl, faith, paul, ye, peter, lord, thing, allah

The top 10 words with highest MI and χ^2 scores for each corpus are listed in Table 3. I'm using the **old version** of the **train_and_dev.tsv**. We can see that there are some stop words in Old English which are not in the current stopwords collection, such as "thou" and "thi". They get relatively higher score in MI than χ^2 for Quran Corpus. Beside these words, the other chosen words are similar but their rankings vary to some extend. For instance, for NT corpus, the word "allah" was ranked 3rd in MI but 10th in χ^2. As these words indicate, we can get an intuitive ideal about each corpus: Quran mainly talks about alla, punishment and believes, OT talks about Jesus Christ and Israel and NT talks about Jesus Christ and his disciples.

4 Topic Analysis

From token analysis, we know that stopwords in Old English may have an impact on our analysis. Also, some common high frequency words in all of the three corpora such as "lord" and "god" will not help us to distinguish the corpora. So to get rid of them as well as uncommon words, I filtered out tokens which are contained in less than 50 verses or more than 10% of the verses to train the LDA model. The top 10 tokens and their probability scores for each of the topic that has the highest average score for the three corpora are as follows:

Table 4: top 10 tokens with probability scores

corpus	id	score	top 10 tokens
Quran	9	0.074	0.127*earth + 0.093*eye + 0.071*heaven + 0.060*6 + 0.052*beast + 0.044*prais + 0.031*mine + 0.028*saul + 0.027*rich + 0.025*hand
OT	13	0.064	0.078*david + 0.059*hous + 0.059*holi + 0.048*dwell + 0.045*place + 0.032*jacob + 0.031*land + 0.029*coven + 0.027*spoken + 0.026*stood
NT	16	0.063	0.096*year + 0.067*voic + 0.063*cri + 0.061*thousand + 0.054*hundr + 0.049*month + 0.048*walk + 0.038*stone + 0.037*young + 0.034*men

From the top 10 ranking tokens we can conclude that the top topic for Quran talks about heaven and earth, the top topic for OT talks about people and places, and the top topic for NT talks about time. Topic 16 seems to be more common in NT and OT (it is ranked 2nd in OT). And the high probability words are "year" (appears 522 times in OT, 65 times in NT and 29 times in Quran), "voic" (appears 264 times in OT, 100 times in NT and 8 times in Quran) and "cri" (appears 236 times in OT, 77 times in NT and 12 times in Quran). We do see that these words are more common in OT and NT than in Quran.

LDA model gives us a general conclusion of the possible topics of each corpus. The overall scores for each token are quite low because we removed some high frequency words in order to make distinguishment, which differs from the purpose of MI and χ^2 method. One word may get high score in MI and χ^2 method, but it may make no contribution to topic analysis because it appears equally likely in all of the three corpora.

5 Classification

To train the baseline model on SVM, I divide the **train_and_dev.tsv** dataset into training set (90%) and development set (10%) randomly. Then punctuation removal and casefold are implemented as preprocessing. And then I extract BOW features and train the SVM classifier on training set with $C = 1000$. The outcome of the model is quite satisfying: the macro-f1 score reaches 0.909 for the development set. But on the other side, it makes the model pretty hard to get improved.

Three instances from the development set that the baseline system labels incorrectly are shown below:

Table 5: 3 instances of incorrect label

true	predicted	content
Quran	OT	in a secure abode in the presence of the powerful king
NT	OT	and they stirred up the people and the elders and the scribesand came upon him and caught him and brought him to the council613 and set up false witnesses which said this man ceaseth not tospeak blasphemous words against this holy place and the law 614 forwe have heard him say that this jesus of nazareth shall destroy thisplace and shall change the customs which moses delivered us
Quran	OT	and among his signs are the ships that run on the sea like mountains and

We see that the model tend to incorrectly classify the given BOW feature as OT category. The reason may be that there are much more BOW features in the OT corpus than Quran and NT corpus (16720, 5612 and 5242 respectively). The classifier gets more knowledge about OT corpus and tends to classify the BOW feature as this category.

Since we can't get more verses of Quran and NT corpus from the given dataset, I've tried several methods for improvement such as changing the SVM parameters, changing the learning method, changing the preprocessing and changing feature selection. They are listed as below:

A : set $C = 10$

B : set $C = 100$

C : set $C = 5000$

D : use LogisticRegression on OneVsRestClassifier

E : use RandomForest

F : remove stopwords

G : use top 5000 features with the highest MI scores

H : use top 5000 features with the highest χ^2 scores

Macro-f1 score is usually used as the measurement of the classifier. So here I include the macro-f1 scores for each of the methods evaluated on the development set and the test set:

Table 6: macro-f1 score for each method

	baseline	A	B	C	D	E	F	G	H
dev set	0.909	**0.914**	0.909	0.909	**0.912**	0.752	0.850	0.751	0.758
test set	0.923	**0.924**	0.923	0.923	**0.926**	0.771	0.861	0.790	0.787

Table 6 shows that there are two kinds of methods which improve the performance of the classifier: set $C = 100$ or use logistic regression on OneVsRestClassifier. By contrast, the former one gets slightly better improvement (0.6% for development set and 0.1% for test set) than the latter one (0.3% for development set and 0.3% for test set), so I reported it in **classification.csv**. To understand this, as C being the regularization parameter, the strength of the regularization is inversely proportional to C. The larger C is, the greater the punishment for wrong classifications is. So the accuracy in the training set is higher, but the generalization ability is reduced. That is, the classification accuracy of the test set might get lower. On the contrary, if C is reduced, some misclassification error samples are allowed in the training samples, and the generalization ability is strong, i.e. the macro-f1 score of the development and test set is higher.