Report: TTDS Coursework 2

Xingtian Zhang, s1946669

November 2019

1 General report

The two parts of this Coursework are separated. The evaluation metrics of the first part are different from that of the second part. So this coursework is not so coupled together as the first coursework, thus easier to start.

1.1 IR Evaluation

Each evaluation metric is calculated for each system-query pair. For a query q, we only need to focus on the relevant documents which are retrived by a system S. Let $D = \{d_1, d_2, ..., d_n\}$ is the relevant document set for query q. For each system-query pair (S, q), the following information is collected for each relevant document $d_i, i = 1, 2, ..., n$:

1. Its ground truth relevance: rel_i. Every relevant document has this information.

2. Its rank in the retrival result of S: $rank_i$. If the document not found by S, this will be infinity.

We can calculate all the metrics from these 2 stats.

T-test is done by hand, and I have found out why we can use t-test to determine whether two systems are statistically different. The formula is $t = \frac{B-A}{\sigma_{B-A}}\sqrt{N}$. The bigger t-value is, the more $B - A$ deviates from 0, which means B is better than A. p-value decreases as t-value increases, so if p-value is smaller than 0.05, B is better than A significantly.

From this perspective, although it is required that we use two-tailed t-test, I'd suggest using one-tailed t-test.

1.2 Text Classification

Advanced preprocessing method is used. I also tried Naive bayes method apart from SVM. Details are in section 3.

2 IR Evaluation

The best system according to each score is shown in Table 1. The highest score is shown in red text, and the second score is shown in blue text. If there are more than one best systems, all of them are marked as red text. According to the table, the best system for each metric is listed below. The p-value threshold is 0.05.

1. P@10: S3, S5 and S6. No need to do t-test.
2. R@50: S2. The p-value of S2-S1 paired t-test is $0.32 > 0.05$. S2 is not significantly better than S1.
3. R-precision: S3 and S6. No need to do t-test.
4. AP: S3. The p-value of S3-S6 is $0.66 > 0.05$. S3 is not significantly better than S6.
5. nDCG@10: S3. The p-value of S3-S6 is $0.24 > 0.05$. S3 is not significantly better than S6.
6. nDCG@20: S3. The p-value of S3-S6 is $0.22 > 0.05$. S3 is not significantly better than S6.

Metric System	P@10	R@50	r-Prec	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 1: The best system according to each score

3 Text Classification

The text classification system is decoupled into following modules

1. Preprocess
2. Feature extraction / Selection
3. Classification Model

The focus is on different ways of preprocessing and two different classification models, SVM and Naive Bayes classifier. The main classification model is SVM provided in the lab. Under SVM, multiple preprocessing methods are applied, such as casefolding and stemming. The best preprocessing settings is then used in Naive Bayes classifier.

3.1 Preprocessing settings

Preprocessing settings A, B, C, D, E are

- remove punctuation
- remove punctuation, casefold
- remove punctuation, casefold, stem
- remove punctuation, casefold, stem, remove stopwords
- remove punctuation, casefold, stem, remove stopwords, duplicate hashtags and ats

When dealing with test set, all unseen words are replaced with OOV symbol.

3.2 SVM parameter

In the lab we run the command `svm_multiclass_learn -c 1000 feats.train model`, where `-c` means trade-off between training error and margin. When comparing results with Naive Bayes, 1000 is kept. But other experiments are conducted to examine the effect of this parameter.

3.3 Naive Bayes method

Smoothing and OOV techniques are used.

When estimating $P(word|class)$, an addtional symbol OOV is added to the dictionary. The initial frequency of OOV is set to be 0.3.

Then for each word in the vocabulary, 0.2 is added to the frequency. For example, the word w_1 already appears under class c and its frequency is f_1. After smoothing, its frequency is $f_1' = f_1 + 0.2$. The frequency of the word OOV is now $0.3 + 0.2 = 0.5$.

After all these steps, we normalize $P(word|class)$ to be a probability distribution.

3.4 Experiment

Effect of preprocessing and Classifer.

Table 2 shows the experiment results. The score is Macro-F1. The **baseline** only removes punctuation and uses SVM. Its Macro-F1 score is 0.614. It shows that use the more preprocessing applied, the higher Macro-F1 score will be. Applying all preprocessing methods increases the results by 8%.

Preprocess Classifier	A	B	C	D	E
SVM	0.614	0.644	0.658	0.678	0.694
Naive Bayes	-	-	-	-	**0.698**

Table 2: Macro-F1 under different experiment settings

Also using Naive Bayes classifier achieves slightly better performance, with an enhancement of 0.4%.

Effect of the tradeoff parameter. The tradeoff parameter around 10^4 gives the best performance, with an increase of 9.7%.

-c Classifier	1	100	1000	3000	10^4	3×10^4
SVM	0.521	0.661	0.694	0.705	**0.711**	0.704

Table 3: Macro-F1 under different experiment settings

I provide 3 Eval.txt files. `Eval.txt` is baseline. `Eval2.txt` is the best SVM result, that with 10^4 tradeoff. `Eval3.txt` is the Naive Bayes result.