Universiteit Leiden

Text Ming
Assignment 1

Peiwen Xing (s3838501)
Zhipei Qin (s3977226)

Leiden Institute of Advanced Computer Science (LIACS)
Faculty of Science
Universiteit Leiden

October, 2023

1 Multiple Classifiers and Features Comparison

In this section, we compared three classifiers using *ScikitLearn*, including Naive Bayes, Support Vector Machine (SVM) and Random Forest. For each classifier, we compared three types of features: counts, tf, and tf-idf. The results tables (Precision, Recall, and F1 score) for different classifiers and features are as follows:

Table 1: Precision

Classifier	Feature		
	Counts	TF	TF-IDF
Naive Bayes	0.76	0.79	0.82
SVM	0.74	0.76	0.84
Random Forest	0.78	0.76	0.77

Table 2: Recall

Classifier	Feature		
	Counts	TF	TF-IDF
Naive Bayes	0.77	0.71	0.77
SVM	0.73	0.76	0.83
Random Forest	0.77	0.75	0.76

Table 3: F1 Score

Classifier	Feature		
	Counts	TF	TF-IDF
Naive Bayes	0.75	0.69	0.77
SVM	0.73	0.76	0.83
Random Forest	0.76	0.75	0.75

The experiment shows that for these three evaluation metrics, the combination of Support Vector Machine (SVM) classifier and tf-idf feature performs the best. We consider that SVM is effective in high-dimensional spaces like text data, and tf-idf highlights keywords while reducing the weight of common words, thereby improving the classifier's performance.

2 Parameter Analysis

For the best classifier-feature combination obtained before, which is the support vector machine mixed with tf-idf (name it "SVM-tf-idf"), attempts to tune four parameters and compares the results respectively. When one parameter is adjusted, the other parameters are set to their default values.

2.1 Lowercase

The Lowercase parameter controls whether alphabetic characters in text are converted to lowercase. By default, lowercase=true, which means that all alphabetic characters in the text will be converted to lowercase. The results table is as follows:

Table 4: Performance Metrics with Different Lowercase Values

Lowcase	Precision	Recall	F1-score	Runtime(s)
True(Default)	0.84	0.83	0.83	322.0
False	0.84	0.83	0.83	357.6

The experiment shows that whether to keep the original case of the letters in the text does not affect the performance of the SVM-tf-idf, as the evaluation metrics are the same for both.

2.2 Stopwords

The Stopwords parameter controls whether common stop words in the text should be filtered out. In addition to not setting stop words by default, this experiment also tried two stop words lists.

Stopwords="english" is a list of English stop words built into *ScikitLearn*; Another list of English stop words is obtained using the *stopwordsiso* library. The results table is as follows:

Table 5: Performance Metrics with Different Stopword Lists

Stopwords	Precision	Recall	F1-score	Runtime(s)
None(Default)	0.84	0.83	0.83	317.8
"english"	0.84	0.83	0.83	250.9
stopwordsiso	0.84	0.83	0.83	215.2

The experiment shows that whether to set stop words has no effect on the classification effect of SVM-tf-idf. However, setting stop words can save a little computational cost.

2.3 Analyzer (in combination with ngram_range)

The ngram_range parameter defines the range of word combinations (n-grams) that will be used to create feature vectors. This experiment tries six ngram_range values, and the results table is as follows:

Table 6: Performance Metrics with Different ngram_range Values

ngram_range	Precision	Recall	F1-score	Runtime(s)
(1, 1)(Default)	0.84	0.83	0.83	312.6
(1, 2)	0.85	0.84	0.84	891.7
(1, 3)	0.84	0.83	0.83	1280.5
(1, 4)	0.84	0.83	0.83	1554.7
(2, 2)	0.79	0.77	0.78	681.1
(2, 3)	0.78	0.76	0.76	1109.8

According to the evaluation results, when ngram_range = (1, 2), the SVM-tf-idf as the feature has the best classification effect, because it has the highest precision, recall and f1-score.

When ngram_range=(1, 2) is set, the text classifier considers both unigrams and bigrams. In this way, we can capture more text features than the default value ngram_range=(1, 1) and other combinations. In addition, The larger the range of word combinations considered, the greater the computational cost.

2.4 Max_features

The Max_features parameter is used to control the number of words contained in the generated vocabulary. Sort the term frequency of all words in descending order and select the max_features most frequent word as the feature. The default value max_features is None, which means that the size of the vocabulary is not limited. This experiment tries four values of max_features, and the results table is as follows:

Table 7: Performance Metrics with Different max_features Values

Max_features	Precision	Recall	F1-score	Runtime(s)
None(Default)	0.84	0.83	0.83	324.2
500	0.57	0.56	0.56	126.4
2000	0.74	0.73	0.73	154.2
5000	0.79	0.79	0.79	186.88

Experiments on the values of these four parameters show that the more the total vocabulary retained, the better the classification effect of the SVM-tf-idf, and the greater the computational cost. SVM-tf-idf performs best when max_features is None.