Text Mining - Assignment 1

Nick Radunovic (s2072742)
Cheyenne Heath (s1647865)

October 10, 2020

Introduction

The goal of this assignment was to classify text data and to evaluate what the effects of using different types of features and feature weights are. For this, three different classifiers have been used to perform a classification task on the "Twenty Newsgroups" (20newsgroup) data set.

Methods

The performance of the classifiers Naive Bayes, Support Vector Machine (SVM) and Random Forest were compared using several types of features and feature weights. The different features that were used with each classifier were Counts, TF and TF-IDF, all of which are features that compute the term weights. The train and test data of the 20newsgroup data set were fetched, followed up by initializing pipelines that train and evaluate each classifier with each of the different features. The default parameters were used for training each classifier to determine which classifier would be used for optimization. The performance of each of these classifiers were expressed in several quality metrics: accuracy, precision, recall and F1-score.

Subsequently, the performance of the SVM classifier when using several non-default values for the parameters of the CountVectorizer function were computed and compared. Only the parameter values of the SVM classifier were tweaked, as SVM appeared to obtain the best results compared to Naive Bayes and Random Forest. Various values of the following parameters were tested: lowercase, stop-words, analyzer and max_features. Not all combinations were analysed, as the results table would get too big. We chose to use lowercase = 'True', stop-words = 'none', ngram_range=(1,1) , analyzer='word' as our baseline and compared the difference when changing only one of the parameters at a time. We also chose to explore the ngram_range and analyzer parameters a bit more by adding three extra experiments change only those parameters.

Lastly, the best parameter values for the function CountVectorizer where computed using a grid search, in order to see whether or not the Count feature would obtain better results than the TF-IDF feature using tweaked parameters. the following grid was used: 'vect_ngram_range': [(1, 1), (1, 2), (1, 3), (2, 2), (2, 3), (3, 3)], 'vect_lowercase': (True, False), 'vect_stop_words': (None, 'english'), 'vect_analyzer': ('word', 'char', 'char_wb'), 'vect_max_features': (None, 10, 50, 100, 500, 1000, 5000, 10000). The exhaustive grid search algorithm ran on the whole data set and computed the best parameter combination out of the parameter values depicted in Table 1.

Table 1: The values for each parameter of the function CountVectorizer on which the grid search was used to calculate the best parameter combination for the classifier SVM.

Parameter	**Values**
lowercase	True, False
stop-words	None, 'english'
analyzer	'word', 'char', 'char_wb'
ngram_range	(1, 1), (1, 2), (1, 3), (2, 2), (2, 3), (3, 3)
max_features	None, 10, 50, 100, 500, 1000, 5000, 10000

Results

The performance obtained of each classifiers after training and evaluating on the 20newsgroup data set are depicted in Table 2. Note, that the SVM classifier obtained the best accuracy on the 20newsgroup data set when paired with the TF-IDF feature. The performance of the classifier SVM when different parameter values for the CountVectorizer where used are depicted in Table 3. The best parameter combination for the function CountVectorizer that were identified by the grid search are depicted in Table 4.

Table 2: Results table showing the accuracy, precision, recall and F1-score metrics for the classifiers and features. The results were obtained using default parameters. The results shown in bold are the best results over all tested classifiers and features.

	Naive Bayes			**SVM**			**Random Forest**		
	Counts	TF	TF-IDF	Counts	TF	TF-IDF	Counts	TF	TF-IDF
Accuracy	0.77	0.71	0.77	0.79	0.83	**0.85**	0.77	0.76	0.76
Precision	0.76	0.79	0.82	0.79	0.83	**0.85**	0.78	0.77	0.76
Recall	0.77	0.71	0.77	0.79	0.83	**0.85**	0.77	0.76	0.76
F1-score	0.75	0.69	0.77	0.78	0.82	**0.85**	0.76	0.75	0.75

Table 3: Results table showing the accuracy, precision, recall and F1-score metrics for the classifier SVM when paired with several parameters for the function CountVectorizer. The results shown in bold are the best resultsover all tested classifiers and features.

Parameter (CountVectorizer)				**SVM**			
lowercase	stop-words	ngram_range	analyzer	Accuracy	Precision	Recall	F1-score
True	None	(1, 1)	'word'	0.79	0.79	0.79	0.78
False	None	(1, 1)	'word'	0.79	0.79	0.79	0.79
True	'english'	(1, 1)	'word'	0.8	0.8	0.8	0.8
True	None	(1, 2)	'word'	**0.81**	**0.81**	**0.81**	**0.81**
True	None	(2, 2)	'word'	0.75	0.75	0.75	0.75
True	None	(1, 1)	'char'	0.09	0.27	0.09	0.06
True	None	(1, 2)	'char'	0.6	0.62	0.6	0.6
True	None	(2, 2)	'char'	0.61	0.61	0.61	0.61

Table 4: The best parameter combination for the function CountVectorizer according the a grid search that was performed on the parameter values of Table 1. This parameter value combination proved to obtain the best result in the classifier SVM.

Parameter	**Values**
lowercase	True
stop_words	'english'
analyzer	'word'
ngram_range	(1, 3)
max_features	None
Accuracy	0.83

Discussion & Conclusion

The classifier SVM peformed best on the 20newsgroup data set compared to Naive Bayes and Random Forest. Of all features that were used, TF-IDF obtained the best results on classifiers Naive Bayes and

SVM. However, for the classifier Random Forest, the feature Counts obtained the best results. The SVM classifier paired with the TF-IDF obtained the best results overall.

The use of custom parameter values resulted in better performances compared to using default parameter values when using SVM and the Count feature. Especially removing stop words and using a bigger ngram_range resulted in a better accuracy when using CountVectorizer. However, the Count feature with the best combination of feature values didn't result in a better accuracy than ID-IDF obtained when performing the same classification task with SVM. This suggests that the ID-IDF feature paired with SVM gets the best classification of the 20newsgroup data set.