Report: Text Classification Comparison

Student ULCN: 4022297 & 4017633 (Nimish Pandey & Mehul Upase)

Introduction

So basically, in this report, we have presented the results of the text classification task using three different classifiers & three feature extraction methods. The overall goal was to evaluate the performance of classifiers on the 20 Newsgroups dataset across different feature types.

The classifiers that we considered were:

1. Multinomial Naive Bayes (MNB)
2. Random Forest
3. Support Vector Machine (SVM)

The feature extraction methods that we included are listed down here:

1. Count
2. Term Frequency (TF)
3. Term Frequency-Inverse Document Frequency (TF-IDF)

Dataset

The Dataset utilized is the 20 Newsgroups dataset, which basically contains a very diverse collection of newsgroup documents organized into 20 different categories & The dataset was actually preprocessed by removing headers, footers, and quotes from it to make it easier for processing.

Classifiers

1. **Multinomial Naive Bayes (MNB)**: Multinomial Naive Bayes algorithm is a probabilistic learning method which is mostly used in Natural Language Processing (NLP). The algorithm is based on the Bayes theorem and basically it predicts the tags of the texts such like for example maybe snippets from an email, magazine, blog, text, or any other article & news articles also. It will then calculate the probability of each tag for a given sample of text, after all this it gives the tag with the highest probability as result. So basically, that's how the Multinomial Naive Bayes works.
2. **Random Forest**: Random Forest is a supervised learning algorithm. The word "forest" in Random Forest is that it builds an ensemble of decision trees, mostly trained with the bagging method & bagging method also known as bootstrap aggregation, is the ensemble learning method that is commonly used to reduce variance within a noisy dataset & altogether in simple words random forest basically builds multiple decision trees & then merges them together to get more accurate and stable predictions.

3. **Support Vector Machine (SVM)**: So basically, SVM is a supervised machine learning algorithm which is like generally used for classification or regression problems, It uses a technique called the kernel trick to transform your data & then based on these transformations it then finds an optimal boundary between the possible output in other words it does a lot of extremely complex data transformations & then it figures out how to separate the data based on the labels or outputs are defined & the best part is that SVM is capable of doing both classification and regression

Features

1. **Count**: Basically it Converts text documents into a matrix of token counts
2. **Term Frequency (TF)**: It is the representation of the frequency of each term in the document
3. **Term Frequency-Inverse Document Frequency (TF-IDF)**: It Weights terms on the basis of their importance in the document & corpus

Results

So, in the table below we have listed the performance of each classifier & feature combination using the Precision, Recall, & F1-score & the results are as follows:

Classifier	**Feature Extraction**	**Precision**	**Recall**	**F1-Score**
Multinomial NB	Count	0.87	0.85	0.84
Multinomial NB	TF	0.83	0.77	0.75
Multinomial NB	TF-IDF	0.88	0.85	0.84
Random Forest	Count	0.85	0.85	0.85
Random Forest	TF	0.85	0.84	0.84
Random Forest	TF-IDF	0.95	0.45	0.61
Support Vector Machine	Count	0.06	0.05	0.01
Support Vector Machine	TF	0.86	0.85	0.85
Support Vector Machine	TF-IDF	0.91	0.91	0.91

Inferences made:

- **Random Forest**: Now as per the table above we can see that Random Forest showed consistent & competitive performance across all the feature extraction methods & yeah, It achieved high Precision, Recall, & F1-scores & with the best results using TF-IDF features & Yes Random Forest turns out to be the best one.

- **Multinomial Naive Bayes (MNB)**: Now coming to MNB it performed well It struggled to capture the nuances of the text data, resulting in inconsistent Precision, Recall, and F1-scores. It also struggled in TF features slightly.

- **Support Vector Machine (SVM)**: Now as per the above results SVM actually outperformed the other classifiers, only when using TF-IDF feature & It demonstrated superior Precision, Recall, and F1-scores in that particular features but unfortunately it didn't perform well in rest of the features, as we can see it messed up a lot in the count feature & this was quite significant difference as compared to other features.

- **Feature Extraction Methods**: Now if we notice TF-IDF consistently outperformed Count and TF across all classifiers this is due to the fact that TF-IDF actually considers the importance of terms in both individual documents and the entire corpus, making it effective in capturing the discriminatory power of words.

To conclude all of this, Random Forest with TF-IDF features emerged as the best performing combination for 20 Newsgroups text classification task & achieving the highest Precision, Recall, and F1-scores. While Multinomial Naive Bayes also performed decent enough but Random Forests' superior performance clearly makes it more suitable for more complex text classification problems.

This report outlines the methods employed, presents the results, and discusses the performance of the classifiers and feature extraction methods. Based on the evaluation, we recommend using Support Vector Machine with TF-IDF features for future text classification tasks on similar datasets.