Coursework Data Wrangling
Part B – Deliverable 2

Module: SET 11121 - Data Wrangling

Name: Marek Meyer

Matriculation number: 40329541

E-Mail: 40329541@live.napier.ac.uk

Date: 20.04.2018

GitHub: https://github.com/mrstobbart/data_wrangling_assessment

Word count: 550

Introduction

This report will discuss the sentiment analysis approach regarding a specific microblog dataset. The code and the dataset are published on GitHub and can be found with the URL available on the title. The following sections examine the different parts of the completed sentiment analysis.

Feature selection

The first step for sentiment analysis of data is a manual analysis of the data. It became apparent that it contained several duplicate lines. To avoid overfitting of the classifier to the dataset, these duplicates were removed. Next, case-conversion was applied to the dataset and the text was then tokenized.

Four different sets of features were used to be trained with the classifier to be able to evaluate the effect of these techniques on the accuracy of the classifier:

1. Unigrams
2. Bigrams
3. Unigrams and bigrams combined
4. Part-of-Speech (POS) tags

Unigram features are simply represented by the tokens in a given text. The incentive to also use bigrams was to allow the classifier to account for negations of sentiment words. The third feature set combined unigrams and bigrams to test if it would further increase accuracy. The last approach creates features by adding the part-of-speech tags to each token and uses these instead of unigrams.

The data pre-processing of removing stopwords was done for unigrams and POS tags, but was not done for bigram features as tests revealed that they lowered the accuracy for them. The likely reason for this is that stopword removal changes the bigrams of a given text as some words are not available anymore.

Classification

The NaïveBayes classifier from the python 'nltk' package was used for classification and a test was run for each of the previously discussed feature sets. To split the available data into training and test sets, 10-fold cross validation was used. This is a technique were the dataset is split into 10 parts, with one used for testing and the rest used for training, repeated ten times and is shown to produce better results than a normal split.

Model evaluation

The different classifiers created were evaluated and compared using the accuracy as the only metric. The reason for this was, that the data was relatively evenly split with 43,2% negative entries and 56,7% positive entries. To also take the metric recall into account by using the F-score was not necessary for a representative evaluation.

Feature selection technique	Accuracy
1. Unigrams	94,65%
2. Bigrams	92,41%
3. Unigrams and bigrams	95,12%
4. Part-of-Speech tags	93,80%

Table 1: Results

It can be seen in Table 1 that the combination of unigrams and bigrams for features performed the best while bigrams had the worst performance. As POS-tagging only enhanced unigrams it can also be seen that this enhancement actually decreased the accuracy.

Discussion

An accuracy of slightly over 95% is a good result for sentiment analysis, however, as the dataset contained mostly simple data that had only three different topics, the evaluated classifiers are likely overfitting on the dataset.

Using lemmatization to improve accuracy was also attempted but disregarded as it had no effect.

Future improvements of the method include the use of different classifiers like SVM or using spell correction. The python library sklearn also supports term frequency-inverse document frequency, a technique often used in search engines that could also create good features for classification.