COMP90049 Project 2: tweets r mad, or r they!?

Anonymized

1. Introduction

Sentiment analysis, the process to identify and extract positive, negative or neutral emotions in source texts is often used to help businesses to understand customers' attitudes towards their products and services so they can make specific adjustments. [1] In this project, we try to identify and categorize people's emotion by analyzing

their tweets. Several machine learning methods are used to implement sentiment analysis models such as Support Vector Machines, Naïve Bayes, Decision Trees and so on.

Three kind of datasets [2] are given for different purposes. The 'train' related tweets data sets are leveraged to train classifier models with various machine learning algorithms while the 'eval' related data sets are used to evaluate the performance of developed models. Evaluation metrics utilized in this report are Accuracy, Precision, Recall and F1-score. F1 score is an evaluation metric which balances Precision and Recall.

- $\text{Accuracy} = \frac{\text{TP+TN}}{\text{TP+TN+FP+FN}}$
- $\text{Precision} = \frac{\text{TP}}{\text{TP+FP}}$
- $\text{Recall} = \frac{\text{TP}}{\text{TP+TN}}$
- $\text{F1\_Score} = \frac{2\text{Recall}*\text{Precision}}{\text{Recall+Precision}}$

This project aims to develop several sentiment analysis models based on various machine learning methods to gain some knowledge about given tweets' data sets and identify whether we could or not recognize people's emotion using tweet text. By comparing different models, we could also obtain a better understanding towards those machine learning methods.

2. Methodology

Here are two kinds of learning problems in machine learning area, one is supervised in which data comes with extra attributes that we want to predict, while another one is called unsupervised in which the training data consists of a set of input vectors x without any corresponding target values. [3] In this project, the problem belongs to supervised learning where we trying to classify the sentiment of each tweet.

2.1 Scikit-Learn

Scikit-Learn is leveraged in this project as a tool to build sentiment analysis models and predict for the unseen dataset. Scikit-learn [3] is a popular machine learning library designed for Python which provides a robust set of algorithms including Naïve Bayes, Decision Trees and so on.

2.2 Model Implementation

There are two major parts to during the process of building each model: training and evaluation. As all raw data has been preprocessed and feature extraction is performed, we could directly use given data to train and evaluate our model. The main steps are as follows:

1) At first, extract and select feature words from tweets after tokenizing, counting and normalizing.
2) Secondly, train the classifier model with given the train data set utilizing build-in method based on certain machine learning algorithm.
3) After building the model, evaluate it with given evaluation data sets and calculate evaluation metrics with output results.
4) Last step, the well trained classifier model allots labels for given unlabeled testing data.

3. Results

3.1 Naïve Bayes Classifier

Naïve Bayes classifier is based on Bayes' probability theorem and assumes that all non-class attributes are conditionally independent. In this report, multinomial naïve bayes is chosen as it is suitable for text classification. Training NB

classifier with given 45 feature words is considered as the baseline of this project. One hypothesis is that the performance of NB classifier might not be good since there might be a strong correlation between certain features.

Table 1. Results of NB model (baseline)

	Precision	Recall	F1-Score	Accuracy
Negative	0.5735	0.1541	0.2430	0.5449
Neutral	0.5284	0.8908	0.6634	
Positive	0.6423	0.2594	0.3696	
Weighted Avg	0.5723	0.5449	0.4860	

3.2 Feature Engineering

To improve the performance of analysis model, feature engineering has to be conducted. Instead of applying given 45 features, we extract features from raw tweet texts. The main steps are as follows:

1) Tokenizing. Convert each tweet text into a sequence of tokens.
2) Data cleaning. Remove all punctuations, and stop words. Stop words refers to commonly used words such as "a", "the", "in" which do no help to train a model.
3) Featuring selection. After feature extraction, we may obtain a huge number unigram feature words. Only top 5000 most frequently occurring words are concerned in this project.
4) Train the classifier with selected features.

Table 2. Comparison of NB model

	Ave Precision	Ave Recall	Ave F1-Score	Accuracy
Old model	0.5723	0.5449	0.4860	0.5449
New model	0.6418	0.6303	0.6182	0.6303

3.3 Decision Trees/Random Forest Classifier

Decision tree is a rule-based, non-parametric supervised learning method can be used for both classification and regression. In this report, entropy is selected as the function to measure node impurity.

Table 2. Results of DT model

	Precision	Recall	F1-Score	Accuracy
Negative	0.6406	0.2765	0.3863	0.60211
Neutral	0.5756	0.8488	0.6860	
Positive	0.6837	0.4315	0.5290	
Weighted Avg	0.6219	06021	0.5754	

Random Forests are variants of Decision Trees, instead of building one tree with training data, RF builds n trees where at each node, "m" features are selected out of all features. One hypothesis is

Table 3. Results of RF model, n=10

	Precision	Recall	F1-Score	Accuracy
Negative	0.6125	0.4249	0.5017	0.6251
Neutral	0.6097	0.7675	0.6796	
Positive	0.6717	0.5350	0.5956	
Weighted Avg	0.6290	0.6251	0.6167	

One hypothesis is that the more trees(N) we build, the better the performance.

Table 4. Comparison of RF models

	Ave Precision	Ave Recall	Ave F1-Score	Accuracy
N=10	0.6290	0.6251	0.6167	0.6250
N=100	0.6378	0.6303	0.6200	0.6303
N=200	0.6407	0.6324	0.6218	0.6324

To be noticed that the processing time of Random Forests models is considerably longer than others.

3.4 Support Vector Machines Classifier

Support Vector Machines are a set of geometric based statistical machine learning methods which are quite helpful in text and hypertext categorization. In this report, linear, polynomial, and radial-basis(rbf) kernel functions are used.

Table 6. Results of SVMs model (kernel=linear)

	Precision	Recall	F1-Score	Accuracy
Negative	0.6448	0.4827	0.5521	0.6553
Neutral	0.6269	0.8058	0.6659	
Positive	0.7453	0.5329	0.6215	
Weighted Avg	0.6664	0.6553	0.6479	

Table 7. Results of SVMs model(kernel=rbf)

	Precision	Recall	F1-Score	Accuracy
Negative	0.6578	0.4297	0.5198	0.6482
Neutral	0.6141	0.8342	0.7074	
Positive	0.7541	0.5007	0.6018	
Weighted Avg	0.6656	0.6482	0.6360	

For polynomial kernel function, the degree of function may affect the performance of the model.

Table 8. Results of SVMs model(kernel=polynomial)

	Ave Precision	Ave Recall	Ave F1-Score	Accuracy
Degree=1	0.6664	0.6553	0.6479	0.6553
Degree=2	0.6566	0.6271	0.6067	0.6271
Degree=3	0.6283	0.5558	0.4797	0.5558

4. Analysis

4.1 Model Evaluation

In general, the precision of "positive" is moderately higher than "negative" and "neutral" which indicates that sentiment analysis models are more likely to identify "positive" emotions, the reason might be it is unambiguous that people use positive words to reveal their positive attitudes, for **example**, *"happy 2nd birthday prince George such a cutie pie you are"*. In contrast, the recall of "neutral" is significantly higher than that of others, suggesting that it is more difficult for models to identify "positive / negative" emotions than "neutral" ones, possible explanation may be selected non-emotional words, **such as** *"people", "make",* and *"look"*, are more frequently appears a large number of tweets. This hitch might be improved by setting different weights to each feature.

We surprised find that the result of the NB-based model is even better compared with the DT-based model. Its Accuracy is slightly higher than DT-based model, the reason maybe DT-based model could automatically prune out "bad" features which occur little which may cause misclassify for some instances. For **example**, although feature *"criminal"* occurs little in training data, it might be important to identify negative sentiment. As variants of DT-based model, RF-based model out performs DT-based ones and as the number of trees increase, the performance becomes better. When the classifier develops 200 trees, the accuracy reaches 0.6323 which is a lot better than DT-based classifier.

The SVMs aim to find optimal decision boundary which is partitioned by kernels while the choice of kernel depends on the distribution of the data. As the result demonstrates, the kernel function "linear" works best among all SVMs models. This infers the distribution of tweets' vectors could be separated well by a linear function.

4.2 Analysis of Feature Engineering

There are 45 non-class features in given preprocessed data set other than id for each tweet, some of them may not be helpful for training the model. For example, attribute "the", "and" and "is", these stop words may appear in a large number of tweets and do no help to train the model to identify sentiments. After extracting features from raw tweets, a lot of useful new features are founded **such as** *"best"*, *"awful", "love"* and so on. The result shows that after selecting preferable features, the performance raises which indicates that feature selection is quite important for some machine learning algorithms such as Naïve Bayes since every feature weights the same in these methods.

5. Conclusion

This project focus on sentiment analysis for short social media messages with machine learning methods, several models are implemented based on Naïve Bayes, Decision Trees and other algorithms. The best result (Accuracy =65.53%) is provided by the SVMs-based model with the kernel is linear.

Comparing with the baseline, after feature extraction, the accuracy of the sentiment analysis model improved dramatically from 54.49% to 63.03%. The reason is there are too many English words that can be used to express emotions, 45 features used in baseline are far too few to train a good model to identify people's sentiments. One **example** is that the obvious negative tweet *"My teeth hurt"* could not correctly identified since the word "hurt" is not selected as a feature attribute.

I believe that after selecting a sufficient number of appropriate features and training with a larger data set, the model will be good enough to identify people's emotions on Twitter.

Reference

[1] Jayasanka, Sachira & Madhushani, Thilina & R. Marcus, E & A. A. U. Aberathne, I & Premaratne, Saminda. (2013). Sentiment Analysis for Social Media.

[2] Rosenthal, Sara, Noura Farra, and Preslav Nakov (2017). SemEval-2017 Task 4: Sentiment

Analysis in Twitter. In Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval ’17). Vancouver, Canada.

[3] Documentation of scikit-learn 0.21.1 https://scikit-learn.org/stable/documentation.html