COMPARISON OF RESULTS – SENTIMENT ANALYSIS ON MOVIE REVIEWS FROM TWITTER USING DIFFERENT CLASSIFIERS

Soumil Mandal

Department of Computer Science Engineering, SRM University, Chennai, 603203, India

ABSTRACT:

Several classification and machine learning algorithms are available today. Sentiment analysis which deals with identifying and classifying of opinions expressed in text source uses such algorithms to do so. These classifiers show a wide range of accuracies depending on the quality of pre-processing of dataset. Micro blogging today has become a very popular communication platform to express one’s views and thoughts on a particular subject. In this paper we take our dataset from Twitter, one of the most popular micro blogging platform. The problem with datasets from such sources is that it contains a lot of misspelled words, slang words and twitter specific tags which decreases the accuracy of classifiers to a great extent. For this reason we introduce a new feature vector and use it to train and test some of the most commonly used text classifiers and analyze the results. It is observed that the overall accuracy increases and also the consistency.

Keywords: Sentiment Analysis, Emotion Analysis, Opinion Mining, Text Classification, Machine Learning, Movie Reviews

[1] INTRODUCTION

Internet is the biggest platform available to express one’s views on a particular topic or agenda. There are general social networking sites as well as sites specifically for writing reviews and opinions. These are huge collections of data, and when mined and analyzed, can be used for machine learning purposes. Using sentiment analysis on data related to reviews, the product owner as well as the general public can know how well it is doing in the market. Since the amount of such content is huge, it’s almost impossible for a customer to go through all of them and this is when sentiment analysis comes into place. Symbolic techniques or Knowledge based approach and Machine Learning techniques are some of the most commonly used methods to do so. While Knowledge based approach requires resources like a large database with predefined emotions, Machine Learning uses classifiers which are trained on a training data set and is then applied on a testing data set for classification.

Sentiment analysis is usually conducted at different levels varying from coarse level to fine level. Coarse level sentiment analysis deals with determining the sentiment of an entire document and fine level deals with attribute level sentiment analysis. Sentence level sentiment analysis comes in between these two [1]. There are many researches on the area of sentiment analysis of user reviews. Previous researches show that the performances of...
sentiment classifiers are dependent on topics. Because of that we cannot say that one classifier is the best for all topics since one classifier doesn’t consistently outperform the other. Sentiment Analysis in twitter is quite difficult due to its short length. Presence of emoticons, slang words and misspellings in tweets are forced to have a preprocessing step before feature extraction. There are different feature extraction methods for collecting relevant features from text which can be applied to tweets also. But the feature extraction is to be done in two phases to extract relevant features. In the first phase, twitter specific features are extracted. Then these features are filtered out from the tweets to create normal text. After that, again feature extraction is done to get more features. This is the idea used in this paper to generate an efficient feature vector for analyzing twitter sentiment. Since no standard dataset is available for twitter posts of movie reviews, we created a dataset by collecting tweets for a certain period of time. By doing sentiment analysis on a specific domain, it is possible to identify the influence of domain information in choosing a feature vector. Different classifiers are used to do the classification to find out their influence in this particular domain with this particular feature vector.

[2] PREVIOUS WORK

Previous work can be divided into two parts. Work related to Machine Learning Techniques [2] and that of Sentiment Analysis.

A. Machine Learning

Methods for Machine Learning uses a training data set for training the classifiers, and a testing data set to implement the trained classifiers for relevant results. The training data set is usually larger than the testing data set for better accuracy. Training sets contain essential feature sets and are class labelled data. Using these kinds of sets a classifier is trained and developed which in the future tries to classify the input data sets into probable class labels. The next step is to validate the model by predicting the probable class labels of unseen feature vectors. Some of the most popular learning techniques are Naïve Bayes (NB), Support Vector Machines (SVM) and Maximum Entropy method. It was found by Dominos et al. that NB works well for certain data sets with high dependent features. Zhen Niu et al. [3] introduced a new model where efficient approaches are used to select feature sets. It also contained efficient weight computation methods and classification. Wu et al. [4] proposed a model which utilized influence probability for twitter based sentiment analysis. Xia et al. [5] used an ensemble framework for sentiment classification. Ensemble framework is obtained by combining various feature sets and classification techniques.

B. Sentiment Analysis

The field of sentiment mining started with product and movie reviews. Pointwise Mutual Information (PMI) method was used by Turney [6] in 2002 to estimate the sentiment of phrases. In the same year, Pang et al. utilized supervised learning with several sets of n-gram features, with unigram presence features on the task of document level binary classification. These followed by research on other types of corpuses which included blogs, by Chesley [7] in 2006 and news, by Godbole et al. [8] in 2007. In the early days of sentiment analysis, researchers mainly concentrated on big data sets, which slowly changed to micro-text, where
the writer expresses his or her emotion in a more concise fashion. Bermingham and Smeaton [9] in the year 2010, confirmed that it is easier to classify sentiment in micro-text as compared to longer documents. The use of unigram and bigram features boost the accuracy of classifiers. Liu et al [10] have utilized real world knowledge about affect drawn from common sense knowledge base. They aim to understand the semantics of text to identify emotions at the sentence level. They begin with extracting from the knowledge base those sentences that contain some affective information. This information is utilized in building affective models of text, which are used to label each sentence with a six-tuple that corresponds to Ekman's six basic emotions. Neviarouskaya et al. [11] have also used a rule-based method for determining Ekman’s basic emotions in the sentences in blog posts.

[3] SOLUTION TO PROBLEM

The dataset is created using twitter posts of movie reviews. Tweets are short messages full of slang words and misspellings. Therefore we perform sentence level analysis. This is done in three phases. In the first phase preprocessing is done. Then a feature vector is created using relevant features. For the final phase, tweets are classified into either positive or negative classes. Based on the number of tweets in each class, the final sentiment is derived.

A. Collection of Dataset

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Positive</th>
<th>Negative</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Training</td>
<td>800</td>
<td>800</td>
<td>1600</td>
</tr>
<tr>
<td>Test</td>
<td>200</td>
<td>200</td>
<td>400</td>
</tr>
</tbody>
</table>

Table: 1. Distribution of Collected Dataset

Since balanced dataset of movie reviews from twitter was not available, we made our dataset by collecting tweets over a period of two months from January 2016 to March 2016. Tweets can be easily collected automatically using the Twitter API. These collected tweets were then manually tagged as either positive or negative. Sarcastic tweets and code mix tweets were ignored. The final dataset was made my taking 1000 positive tweets and 1000 negative tweets. [Table-1] shows the distribution of collected dataset for training and testing purposes.

B. Preparation of Dataset

Preprocessing of dataset is a tedious job when obtained from Twitter. This is mainly due to misspelled words, slang words and informal acronyms. To avoid this, some preprocessing routines are applied on the dataset. The function of the routines are to remove URL, ignore misspelled words and make a dictionary of slang words. Slang words can’t be simply removed because they contain an important ratio of weightage of the sentiment that a tweet holds. This dictionary is therefore used to replace slangs words occurring in tweets with...
their corresponding meanings. A specific domain, in this case movie reviews, contributes much to the information of slang word dictionary.

C. Creating the Feature Vector

Extraction of features is done in two basic steps. For the first step, we first extract twitter specific features. These include hashtags and emoticons. There are emoticons for both positive and negative emotions, therefore they are given different weights. Only commonly used emoticons are considered and complex and rarely used ones are ignored. Positive emotions are given a weight ranging from +1 to +1.5 while negative emotions are given a range of -1 to -1.5. A range of values are used because there are subsets of binary emoticons which shows different degrees of positivity or negativity. There may also be positive and negative hashtags. Therefore the count of positive and negative hashtags are added as two separate features in the final feature vector.

All tweets does not contain twitter specific features. Therefore a further feature extraction is done to collect other features as well. Collected twitter specific features are then removed from all the tweets. The newly formed dataset can then be considered as a collection of words. Then using unigram approach, tweets are represented as a collection of words. In unigrams, a tweet is represented by its keywords. We maintain a negative keyword list, positive keyword list and a list of different words that represent negation. Counts of positive and negative keywords in tweets are used as two different features in the feature vector. Presence of negation contribute much to the sentiment. So their presence is also added as a relevant feature. All keywords cannot be treated equally in the presence of multiple positive and negative keywords. Therefore a special keyword is selected from all the tweets. In the case of tweets having only positive keywords or only negative keywords, a search is done to identify a keyword having relevant part of speech. A relevant part of speech is adjective, adverb or verb. Such a relevant part of speech is defined based on their relevance in determining sentiment. Keywords that are adjective, adverb or verb shows more emotion than others. If a relevant part of speech can be determined for a keyword, then that is taken as special keyword. Otherwise a keyword is selected randomly from the available keywords as special keyword. If both positive and negative keywords are present in a tweet, we select any keyword having relevant part of speech. If relevant part of speech is present for both positive and negative keywords, none of them is chosen. Special keyword feature is given a weight of ’1’ if it is positive and ’-1’ if it is negative and ’0’ in its absence. Part of speech feature is given a value of ’1’ if it is relevant and ’0’ otherwise. Thus feature vector is composed of 8 relevant features. The 8 features used are part of speech (POS) tag, special keyword, presence of negation, emoticon, number of positive keywords, number of negative keywords, number of positive hash tags and number of negative hash tags.

[4] METHODS OF CLASSIFICATION

There are different types of classifiers that are generally used for text classification which can be also used for twitter sentiment classification.
A. Naïve Bayes Classifier

Naïve Bayes Classifier makes use of all the features in the feature vector and analyzes them individually as they are equally independent of each other. The conditional probability for Naive Bayes can be defined as

\[ P(X|y_j) = \prod_{i=1}^{m} P(x_i|y_j) \]

'X' is the feature vector defined as \( X = \{ x_1, x_2, x_3, ..., x_m \} \) and \( y_j \) is the class label. Here, in our work there are different independent features like emoticons, emotional keyword, count of positive and negative keywords, and count of positive and negative hash tags which are effectively utilized by Naive Bayes classifier for classification. Naive Bayes does not consider the relationships between features. So it cannot utilize the relationships between part of speech tag, emotional keyword and negation.

B. Support Vector Machine Classifier

SVM Classifier uses large margin for classification. It separates the tweets using a hyper plane. SVM uses the a discriminative function defined as

\[ g(X) = w^T \phi(X) + b \]

'X' is the feature vector, 'w' is the weights vector and 'b' is the bias vector. \( \phi(X) \) is the non-linear mapping from input space to high dimensional feature space. 'w' and 'b' are learned automatically on the training set. Here we used a linear kernel for classification. It maintains a wide gap between two classes.

C. Maximum Entropy Classifier

In Maximum Entropy Classifier, no assumptions are taken regarding the relationship between features. This classifier always tries to maximize the entropy of the system by estimating the conditional distribution of the class label. The conditional distribution is defined as

\[ P_\lambda(y|X) = \frac{1}{Z(X)} \exp \left\{ \sum_i \lambda_i f_i(X,y) \right\} \]

'X' is the feature vector and 'y' is the class label. Z(X) is the normalization factor and \( \lambda_i \) is the weight coefficient. \( f_i(X,y) \) is the feature function defined as
In our feature vector, the relationships between part of speech tag, emotional keyword and negation are utilized effectively for classification.

**D. Ensemble Classifier**

Ensemble classifiers can be of different types. They try to make use of the features of all the base classifiers to do the best classification. The base classifiers used here are Naïve Bayes, Maximum entropy and SVM. Here an ensemble classifier is generated by voting rule. The classifier will classify based on the output of majority of classifiers.

![Graph showing performance of different classifiers](image)

**Figure: 1. Performance of different classifiers**

Since we have selected a particular domain, there is no need of analyzing subjective and objective tweets separately. To understand whether the movie is good or bad, both of these qualities contribute similarly. This shows how context or domain information affects sentiment analysis. Part of speech tagging is done using the Python NLTK module which has built in speech tagger. The classifiers are tested using the scikit-learn package in Python. We have used three types of basic classifiers (Naïve Bayes, SVM, Maximum Entropy) and Ensemble classifier for sentiment analysis. All of these classifiers are readily available in the mentioned Python module. Performance of these classifiers is shown in [Figure-1]. All these classifiers have almost a similar average performance thus a very close G score. Naïve Bayes has better precision compared to the other three classifiers, but slightly lower accuracy and recall. SVM, Maximum Entropy and Ensemble classifiers have similar accuracy, precision and recall. They obtained an accuracy of 90% whereas Naïve Bayes obtained an
accuracy of 89.5%. This portrays the quality of the feature vector selected for the product domain. This feature vector aids in better sentiment analysis despite of the classifier selected.

[5] CONCLUSION

Lots of different methods are there for sentiment classification using Machine Learning techniques. The modern ML techniques are much faster and more efficient than Symbolic techniques. We see moderate issues popping up when the elements of the datasets, in this case movie reviews have multiple keywords in them. It also becomes a problem when it contains short forms, misspelled words and slang words. The newly introduced feature vector counters such problems by better pre-processing. Twitter specific features related to movie reviews are identified and incorporated in the feature vector. The incorporated features are then removed from the movie reviews and then feature identification and retrieval is done like normal text. The same features are also put into the feature vector. Naïve Bayes, SVM and Maximum Entropy classifiers accuracies are tested using this feature set. A voted classifier using the basic three classifiers is also tested. The average accuracy is found to be quite satisfactory. This method can be applied to other types of datasets as well like product reviews, place reviews, food reviews and such.

REFERENCES


Mr. Soumil Mandal is pursuing B.Tech in Computer Science and Engineering from SRM University, Kattankulathur (Chennai) India.