Sentiment Analysis of Twitter Data

Abstract:
The growth in the field of sentiment analysis has been rapid and tremendous and it aims to extract and present the sentiments associated with the mined data from various social media platforms using machine learning algorithms. Even though various machine leaning algorithm has been used for prediction of sentiment during an election season still there is huge scope of improvement in this area. So, in order to deal with the data, we have adapted a hybrid approach which compares the outcome of various machine learning algorithms such as Decision trees, Maximum entropy and Naive Bayes and provides the best outcome of the 3 methods.

Introduction:
Over the period websites have evolved to provide various kind of information. This is due to expression of real-time opinions through text, blogs, reviews, and tweets on a variety of topics to discuss on a range of content like current issues, complaints, and review of products, etc.
In the recent day's companies producing or launching a new product in the market are no longer exclusively making it available to a limited set of people instead they are launching it and scrapping the web for any information related to the product and getting the sentiment that is associated with the product. But the main challenge for this field is to put in a reliable system in place so that the sentiment of the product is effectively tracked. In this project, we are looking at once such aspect of classifying the sentiment by extracting the data from Twitter to gauge the sentiment of the tweets into positive, negative, and neutral. We experiment with 3 types of models. 1. Naïve Bayes 2. Maximum entropy 3. Decision tree. Our experiments show that twitter specific features like emoticons, hashtags add value to the classifier. We have done this experimentation to gauge the-sentiment of the people for Indian general elections.

Related Work:

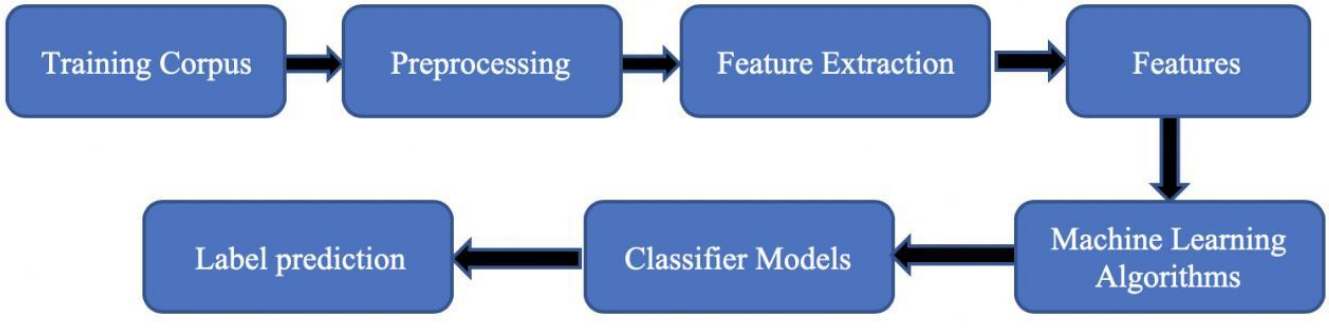
1. Go, Bhayani and Huang (2009): They had collected training dataset directly from Twitter to classify the tweets as a negative or positive sentiment. They had tried various features like unigram, bigram, and part of speech to train their model and classify using Naïve Bayes, Support vector machine and maximum entropy. They had concluded that Naïve Bayes is the best classifier and bigrams alone are not helpful for sentiment prediction.
2. Pak and Paroubek (2010): They identified that using creative and inform language make it difficult to analyze the tweets. They have leveraged the previous work done in this field using the #tag to build their classifier. They have used Edinburgh twitter dataset to manually annotate the hashtags and run the classifier on the corpus to classify the sentiment. They found that n-grams with lexicons provide the best result but while using P-O-S features causes a drop in the accuracy of the model.

Data Description:

Twitter is a microblogging and networking application that allows everyone to post messages called tweets. These are short messages that are restricted to 280 characters per tweet. Due to the availability of limited characters people generally use acronyms, emoticons and other characters that express a special meaning to the text that has been written. Users generally use the following terminology to express themselves 1. Hashtags: which is generally done to increase the visibility and express themselves for a particular topic to trend the topic. 2.Emoticons: These are the special characters that the users use to express the sentiment that is associated with the text. 3. Target user: '@' is used to target a particular user towards whom the message is directed. We have used 2 of the manually annotated data that are generally used for sentiment analysis i.e., Stanford twitted corpus Which have over a million annotated tweets that are used to build our classifier. The mentioned corpus is available publicly which they have collected over a period by real-time streaming of Twitter data. The above data is annotated manually by a human annotator as positive, negative, and neutral. For the test data set data was extracted from twitter using tweepy API for #mainbhichowkidaar to gauge the sentiment of the Indian general election.

Methodology:

We have used different type of classifiers to determine the best outcome for sentiment analysis of twitter data. As there will be a visible comparison between the diff classifiers and how they are performing for same set of features so that the best classifier and feature set can be taken.



Pre-processing:

User-generated data that is available on the web is not present in the form so that it can be directly used for learning of the model. So, it is necessary to normalize the data by using pre-processing steps. In this project, we have applied an extensive set of preprocessing to decrease the feature size to make it suitable for machine learning algorithms. Below are the preprocessing methods that we have implemented in this project.

1. **Hashtags**: A phrase that is prefixed with # in order to target a specific topic and increase the visibility by generating more no of tweets for that particular topic.

   Ex: #Champions #football

   Regular expression: #(\w+)

   Replaced with: HASH_\1

2. **Handel's**: Every user has a unique name that starts with @. So, any content that starts with an @ is directed towards a particular user.
    Ex: @Microsoft, @umich
    Regular expression: @(\w+)
    Replaced with: HNDL_\1
3. **URLs**: Users often use hyperlinks to share content from a different website in their tweet. Which is generally used to lead the user out of twitters domain. From the classification point of view, URLs don't have any impact, but these URLs can contain valuable information pertaining to the tweets. As twitter automatically shortens the URLs the below regular expression was used to clean the data
    Ex: @Microsoft, @umich
    Regular expression: (http|https)://[a-zA-Z0-9\\./]+
    Replaced with URL
4. **Punctuations**: Certain punctuations play an important role in expressing the sentiment of the text like exclamation which generally means excited, question mark, etc. each word boundary was replaced with the relevant punctuation and every single quote present in the text was removed.
    Ex. '.' replaced with: PUNC_DOT
    '!' replaced with: PUNC_EXCL
    '?' replaced with: PUNC_QUES
5. **Emoticons**: Emoticons are the most relevant part while extracting the sentiment from microblogging sites. Each emoticon was replaced with a word in the corpus.
    Ex. ':-), :), (:, (-:' replaced with EMOT_SMILEY
6. **Repeating Characters:** When on a certain excited mood people often repeat or elongate the word. So, these characters are replaced as two words in the dataset.
    Ex: Goooooood, Hungryyyy
    Regular expression: (.)\1{1,}
    Replaced with: \1\1

Stemming and lemmatization:

1. **Stemming**: Porter stemmer was used to remove the prefix and suffixes from the words, but this can sometimes lead to inaccurate meaning. But it offers good speed, readability, and accuracy and it doesn't involve recursion. Ex: Gets rid of plurals and -ed or -ing suffixes.

2. **Lemmatization**: The process of normalizing a word rather than stemming makes more sense as it keeps the meaning of the phrase intact. Ex: verb 'saw' can be lemmatized to 'see'. But for this project, we haven't implemented lemmatization which can be implemented in the future to check the outcome of the model.

Features:

Multiple features can be used to frame a classifier in order to classify the tweets. The most frequently implemented and fundamental features are word n-grams. Although, there is a plethora of domain relevant information available in the tweets, they also could be used for classification. In this project, we have worked with four sets of features.

Unigrams:

Using unigrams is the most common features in the text classification segment. A tweet is designated by multiple clusters of words in it. The existence of a particular word is more consequential than its frequency of repeating multiple times. In our project, we gauge all the unigrams inside the tweets as a feature. For example, in the statement “Travelling to king’s palace!”, the existence of each unigram gives better information despite of considering the number of times a unigram is repeating.

Bi grams:

Bigrams are collection of two unigrams as pairs. By the usage of bigrams, our probability of detecting the succeeding word and its senses would be increased. Based on the frequency of the occurrence of bi-gram pairs, we can detect the probabilities of words co-occurring with them.

Trigrams:

Trigrams are the collection of three unigrams as one trigram. By using the trigrams, our probability of detection of words increases. But as we increase the number of words, the number of features also increase phenomenally. The more the number of features, it becomes complicated to manage.

Negation Handling:

This is an important segment in the feature recognition as they affect the contradiction of other words. Here we detect the negation words in the sentence level. Using negative word detection, we can differentiate between meanings of phrases.

Scope of Negation

The scope of the negative word can be limited just to the word followed by negation, or it can be extended up to certain words preceding and succeeding the negation word. For example, “This burger does not taste good, but its quality is fine.” The scope of the negation word is applicable only to the word following negative word. In the sentence. “This machine is not functioning for 2 years”, the negation holds valid till the end of the sentence. The scope of negation is not limited, and it varies depending on multiple features like punctuations, part of speech (POS), conjunction etc. There are various forms of negation but in high level, we can consider two types of negations: morphological and syntactic negations. Negation handling and detection is easy in the simple sentences, but it is a bit challenging in the complex sentences.

Experimentation:

We have taken 95% of data on train of our information utilizing different coalescences of options and check them on the remaining 5%. We tend to take the options in the following cumulations.

Only unigrams, unigrams added with filtered bigrams and trigrams, unigrams including negation, unigrams with additional filtered bigrams and trigrams and negation. We tend to then train classifiers utilizing different relegation algorithms - Ingenuous Naive Bayes Classifier, Maximum Entropy Classifier and Decision Tree Classifier.

The task of relegation of a tweet is often worn out 2 steps – at first categorize “neutral” versus “objective” tweets and so secondly, objective tweets into two forms “positive” versus “negative” tweets. We tend to this trained two step classifiers.

1. Naive Bayes Classifier:

Naive Bayes classifier is that the simplest and therefore most time saving classifier. Several researchers claim to own gotten best results utilizing this classifier.

For a given tweet, if we tend to need to search out the label for it, we discover the chances of the labels, if feature then cull the label with most probable chance.

The accuracies of Unigrams are lowest at 76.40%. The accuracy will increase if we use Negation detection (80.66%) or higher order n-grams (77.66%). we tend to visually understand that if we tend to employ each Negation detection and better order n-grams, the accuracy is marginally increasing, but simply utilizing higher order n-grams (79.3%). We can yet note that accuracies for double step classifier are less than those for corresponding single step classifier.

2. Maximum Entropy Classifier:

This classifier works by finding a probable distribution that maximizes the likeliness of testable information. This probable function is parameterized by weight

vector. The favorable value of which can be observed utilizing the technique of Lagrange multipliers. Accuracies follow a same trend as compared to Naïve Bayes classifier. Unigram is the lowest feature at 52.5% and that we visually understand associate incrementation for negation detection at 51%. The maximum value is achieved with unigrams, bigrams, and trigrams at 51% proximately followed by n-grams and negation at 51%.

3. Decision Tree Classifier:

Decision tree algorithm is used to for classification problems. Decision trees are a classifier model in which each node of the tree is representing a test on the attribute of the data set, and its children represent the outcomes. The leaf nodes represent the final classes of the data points. It is a supervised classifier model that uses knowledge with notable labels to create the choice tree, so the model is applied on the look at knowledge.

Conclusion:

Data Set: Labelled(supervised)

Classifier: Naïve Bayes classifier is the only classifier we are implementing.

Naive Bayes classifier achieves highest average accuracy by implementing features

1. Unigrams+ negation detection gives accuracy 74.955%
2. Unigrams+ negation detection + higher order n-grams (bigrams and trigrams) 74.955%

Maximum Entropy classifier average accuracy is

1. Unigrams+ negation detection gives accuracy 73.024%
2. Unigrams+ negation detection + higher order n-grams (bigrams and trigrams) 73.024%

In conclusion, Naive Bayes achieves highest accuracy.

From the project implementation we have a strong feeling that feeding the predicted value of one classifier as training data to another classifier that is in series can increase the accuracy drastically.

Future Work:

1. Multilingual: Create a sentiment classifier applicable to many languages used in twitter. Ex. 22 official languages in India.
2. More no features: Adding some more features like strong words, POS tagger, intensity word to the model can increase the model
3. Semantic Analysis: Understanding nouns helps us classify a tweet better. Ex. If Apple and iphone used in the same line. Classifier should be able to identify “Apple” is a company and “iphone” is a product.

References:

1. “Twitter as a corpus for sentiment analysis and opinion mining” by Alexander Pak and Patrick Paroubek.
2. “Twitter sentiment analysis: The good the bad and the omg!” by Johanna Moore, Theresa Wilson and Efthymios Kouloumpis.