Twitter Sentiment Analysis

Rucha Dubbewar
A20373886

Mithil Amin
A20386345

1. Abstract

Our project is focused on sentimental analysis of Chicago cubs twitter data. We collect data from 27 September 2017 to 15 April 2018 from twitter using twitter API. We started with preprocessing data. And then we use simple positive and negative word counting method to assign sentiment to a given tweet. Then based on number of positive and negative words we mark the tweet as positive and negative. We use 80 percent of data as a training data and remaining 20 percent as test data. We used different machine learning algorithms to conduct sentimental analysis and estimate the model with it's best accuracy.

2. Introduction

Twitter is a social networking service and an online news where users interact with messages and post events and news, known as "tweets". Twitter can be used for business in expansion of potential as well as to express feeling of an individual or thought of particular event or a subject. In this report, we will attempt to conduct sentiment analysis on "tweets" using various different machine learning algorithms. We attempt to classify the polarity of the tweet where it is either positive or negative. If the tweet has both positive and negative elements, the more dominant sentiment should be picked as the final label. We use various machine learning algorithms to conduct sentiment analysis using the extracted features. However, just relying on individual models did not give a high accuracy so we pick the top few models to generate a model ensemble. Ensembling is a form of meta learning algorithm technique where we combine different classifiers in order to improve the prediction accuracy.

3. Background/Related Work

Sentiment analysis: capturing favorability using natural language processing: This paper illustrates a sentiment analysis approach to extract sentiments associated with polarities of positive or negative for specific subjects from a document, instead of classifying the whole document into positive or negative.

Sentiment analysis of Twitter data: Contributions of this paper are POS-specific prior polarity features, use of a tree kernel to obviate the need for tedious feature engineering.

Opinion mining and sentiment analysis on a Twitter data stream: This paper discusses an approach where a publicized stream of tweets from the Twitter microblogging site are preprocessed and classified based on their emotional content as positive, negative and irrelevant; and analyses the performance of various classifying algorithms based on their precision and recall in such cases.

A system for real-time Twitter sentiment analysis of 2012 U.S. presidential election cycle: This paper describes a system for real-time analysis of public sentiment toward presidential

candidates in the 2012 U.S. election as expressed on Twitter, a micro-blogging service. Sentiment analysis can help explore how these events affect public opinion. While traditional content analysis takes days or weeks to complete, the system demonstrated here analyzes sentiment in the entire Twitter traffic about the election, delivering results instantly and continuously. It offers the public, the media, politicians and scholars a new and timely perspective on the dynamics of the electoral process and public opinion.

Combining Lexicon-based and Learning-based Methods for Twitter Sentiment Analysis: This paper presents a new entity-level sentiment analysis method for Twitter using lexicon based approach to perform entity level analysis.

4. Approach

We perform experiments using various different classifiers. We use 20% of the training dataset for validation of our models to check against overfitting. For Naive Bayes, Maximum Entropy, Decision Tree, Random Forest and SVM, we use sparse vector representation of tweets. Existing libraries will be used for above stated machine learning algorithms.

5. Experiment

5.1 Baseline

For a baseline, we use a simple positive and negative word counting method to assign sentiment to a given tweet. We use dataset of positive and negative words to classify tweets. In cases when the number of positive and negative words are equal, we assign positive sentiment. Using this baseline model, we would try to achieve most accuracy. With baseline in use, accuracy of 43% is obtained.

5.2 Data Description

Live stream twitter data will be used. Data will be collected using Twitter API. Problems related with data collected will mostly be refining it and modifying it in such a way to make it useful for analysis. The train dataset is a csv file of type tweet_id, sentiment, tweet where tweet_id is unique integer identifying the tweet and sentiment is either positive(1) or negative (0). Similarly test dataset is a csv file of type tweet_id, tweet. This data collected contains words, user mentions, emoticons, URLs, unigrams and bigrams. User profile used for this project is Chicago cubs.

	Train Data	Test Data
Total	2590	647
Positive	1147	278
Negative	1433	369
@	1511	507
URL	1906	555
Emojis	4	12
Unigrams	20908	5048
Bigrams	18340	4406

5.2.1 Pre-processing

Raw tweets scraped from twitter generally result in a noisy dataset. Tweets has some special characters like re-tweets, emoticons, urls, user mentions, etc. which have to be suitably extracted. Therefore, we normalize raw twitter data and create dataset which can be easily learned by various classifiers. We applied some preprocessing steps to standardize the dataset to reduce its size. Preprocessing of raw tweet is as follows.

- Convert tweets to lower case.
- Dots (.) and spaces are being replaced.
- Urls are replaced with “URL”.
- User Mention replaced with “USER_MENTION”.
- Hash (#) is replaced from hashtags.
- Strip spaces and quotes (“ and ‘) from the ends of tweets.
- Emoticons are replaced with EMO_POS and EMO_NEG
- Replace multiple spaces with single space

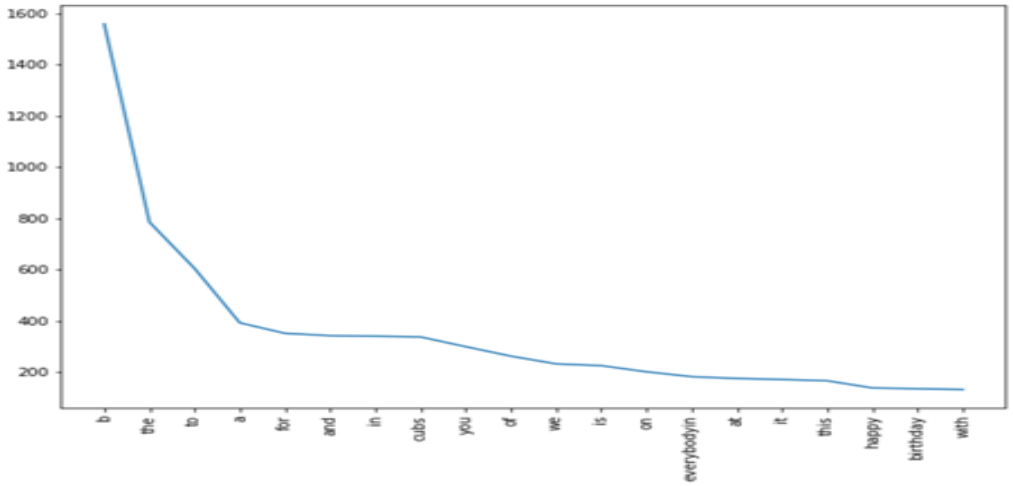
5.2.2 Feature Extraction

For a baseline, we use a simple positive and negative word counting method to assign sentiment to a given tweet. We use dataset of positive and negative words to classify tweets. In cases when the number of positive and negative words are equal, we assign positive sentiment. Using this baseline model, we would try to achieve most accuracy.

We extract two types of features from our dataset, namely unigrams and bigrams. We create a frequency distribution of the unigrams and bigrams present in the dataset and choose top N unigrams and bigrams for our analysis.

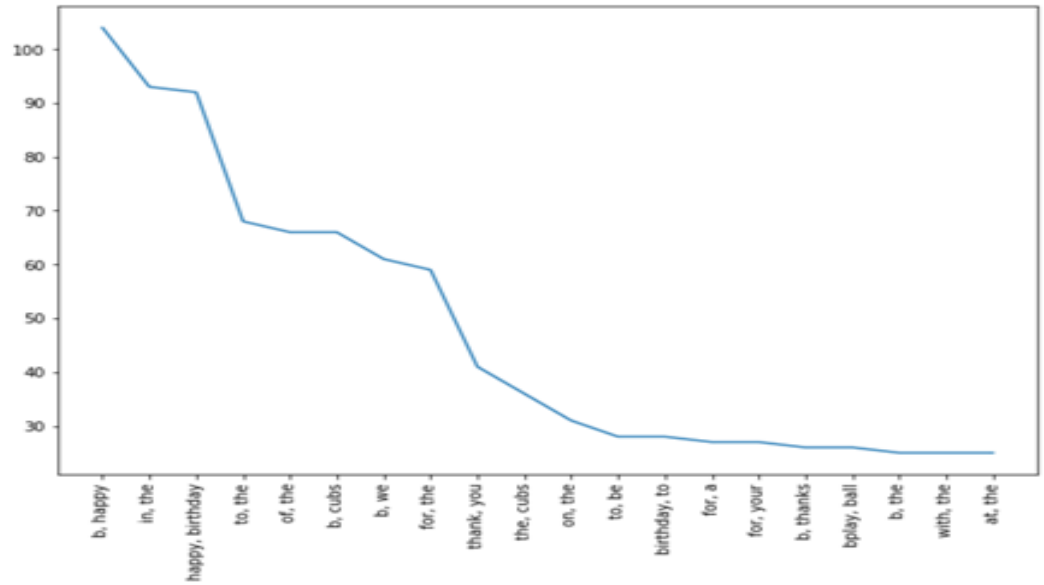
5.2.2.1 Unigrams

In unigram we extract single words from the training dataset and create a frequency distribution of these words. A total number of 20908 unique words extracted from datasets. In unigram (b, the, to, a , for) are top 5 frequently used words.



5.2.2.2 Bigrams

Bigrams are word pairs in the dataset which occur in succession in the corpus. These features are a good way to model negation in natural language. A total of 18340 unique bigram were extracted from the dataset. [(b, happy), (in, the), (happy, birthday), (to, the), (of, the)] are the top 5 frequently used bigrams. It shows that they wish lot of their followers or users on their birthdays and also they comment “b happy” too many times.



5.3 Different Machine Learning Algorithms and Accuracies

5.3.1 Naive Bayes

We used MultinomialNB from sklearn.naive_bayes package of scikit-learn for Naive Bayes classification. We used Laplace smoothed version of Naive Bayes with the smoothing parameter α set to its default value of 1. Using Naïve bayes, 73.74% of accuracy is obtained with presence of unigram while an accuracy of 74.90% is obtained with presence of combination of unigrams and bigrams.

5.3.2 Maximum Entropy

The nltk library provides several text analysis tools. We use the MaxentClassifier to perform sentiment analysis on the given tweets. Using Maximum Entropy, 84.25% of accuracy is obtained with presence of unigram while an accuracy of 86.27% is obtained with presence of combination of unigrams and bigrams.

5.3.3 Decision Tree

We use the DecisionTreeClassifier from sklearn.tree package provided by scikit-learn to build our model. GINI is used to evaluate the split at every node and the best split is chosen always. Using Decision Tree, 75.67% of accuracy is obtained with presence of unigram while an accuracy of 78.33% is obtained with presence of combination of unigrams and bigrams.

5.3.4 Support Vector Machine

We utilise the SVM classifier available in sklearn. We set the C term to be 0.1. C term is the penalty parameter of the error term. In other words, this influences the misclassification on the objective function. Using SVM, 71.33% of accuracy is obtained with presence of unigram while an accuracy of 76.33% is obtained with presence of combination of unigrams and bigrams

5.3.5 Random Forest

We implemented random forest algorithm by using RandomForestClassifier from sklearn.ensemble provided by scikit-learn. Using Random Forest, 80.67% of accuracy is obtained with presence of unigram while an accuracy of 75.33% is obtained with presence of combination of unigrams and bigrams

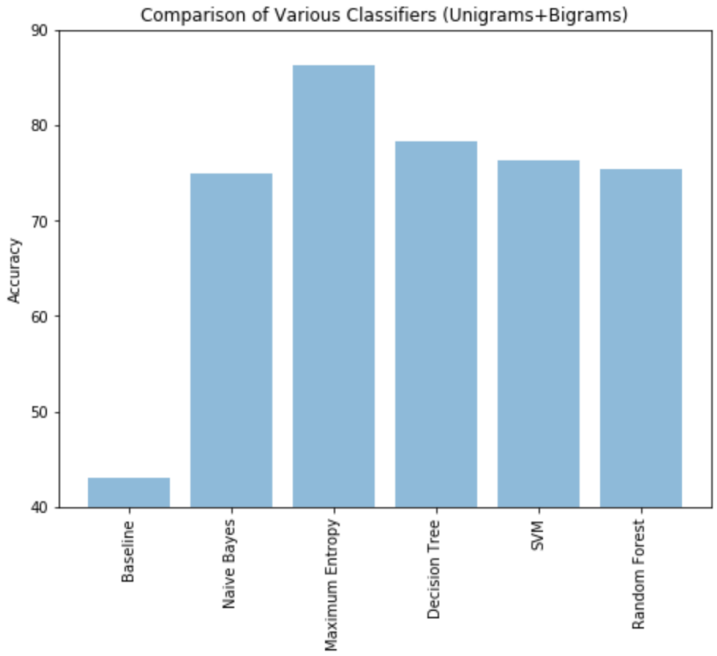
6. Conclusion

The provided tweets were a mixture of words, emoticons, URLs, hashtags, user mentions and symbols. Before training we preprocess the tweets to make it suitable for feeding into models. We implemented several machine learning algorithms like Naive Bayes, Maximum Entropy, Random Forest, SVM and Decision Tree to classify the polarity of the tweet. We used two types of features namely unigrams and bigrams for classification and observes that augmenting the feature vector with bigrams improved the accuracy. Once the feature has been extracted it was represented as either a sparse vector or a dense vector.

9.81E+17	-1	USER_MENTION still delayed
9.86E+17	-1	game between the cubs and braves has been postponed due to the forecast for inclement weather URL

Cubs twitter account posted lots of delayed match updates that's why we got more negative tweets then the positive.

Here is the graph of different algorithms with it's accuracies with unigrams and unigrams+bigrams



Machine Learning Algorithms	Unigram	Unigram+Bigram
Naive Bayes	73.74	74.90
Maximum Entropy	84.25	86.27
Decision Tree	75.67	78.33
SVM	71.33	76.33
Random Forest	80.67	75.33

Here as you can see Maximum Entropy has the highest accuracy with 86.27 percent.

7. References

- Sentiment analysis: capturing favorability using natural language processing - https://dl.acm.org/citation.cfm?id=945658
- Sentiment analysis of Twitter data - https://dl.acm.org/citation.cfm?id=2021114
- Opinion mining and sentiment analysis on a Twitter data stream- http://ieeexplore.ieee.org/abstract/document/6423033/
- A system for real-time Twitter sentiment analysis of 2012 U.S. presidential election cycle - https://dl.acm.org/citation.cfm?id=2390490
- Combining Lexicon-based and Learning-based Methods for Twitter Sentiment Analysis - https://search.proquest.com/openview/3bb4518ced02bbd25b28594d28b35094/1?pq-origsite=gscholar&cbl=2029261