SMAI

Sentiment Analysis in Twitter

Introduction

With the enormous increase in web technologies, number of people expressing their views and opinions via web are increasing. This information is very useful for businesses, governments and individuals. With over 500+ million Tweets (short text messages) per day, Twitter is becoming a major source of information. Twitter is a micro-blogging site, which is popular because of its short text messages popularly known as Tweets. Tweets have a limit of 140 characters. Twitter has a user base of 240+ million active users and thus is a useful source of information. Users often discuss on current affairs and share their personals views on various subjects via tweets. Out of all the popular social medias like Facebook, Google+, Myspace and Twitter, we chose Twitter because of the following reasons:

- Twitter contains an enormous number of text posts and it grows every day. The collected corpus can be arbitrarily large.
- Twitter's audience varies from regular users to celebrities, company representatives, politicians, and even country presidents. Therefore, it is possible to collect text posts of users from different social and interests groups.
- Tweets are small in length, thus less ambiguous.
- Tweets are unbiased in nature.

Using this social media we built two models for classifying ”tweets”. In the first one, we classify the tweets into positive, negative and neutral classes. We build models for two classification tasks: a 3-way classification of already demarcated phases in a tweet into positive, negative and neutral classes and another 3-way classification of entire message into positive, negative and neutral classes. We experiment with the baseline model and feature based model. We do an incremental analysis of the features. We also experiment with a combination of models: combining baseline and feature based model.

In the second model, we classify the tweets into subjective sentiments of sad, happy, angry and surprised. We experiment with various hyperparameters and features in a GMM. The GMM gives us a relative score for the tweet instead of an absolute classification which is in line with the subjectivity of any tweet.

DATASET

Twitter is a social networking and microblogging service that allows users to post real time messages, called tweets. Tweets are short messages, restricted to 140 characters in length. Due to the nature of this microblogging service, people use acronyms, make spelling mistakes, use emoticons and other characters that express special meanings. Following is a brief terminology associated with tweets.

- Emoticons: These are facial expressions: pictorially represented using punctuation and letters; they express the user's mood.
- Target: Users of Twitter use the @ symbol to refer to other users on the microblog. Referring to other users in this manner automatically alerts them.
- Hashtags: Users usually use hashtags to mark topics. This is primarily done to increase the visibility of their tweets.

In this project, for the multipolar classification we use the dataset collected and annotated for the SMILE project. This collection of tweets mentioning 13 Twitter handles associated with British museums was gathered between May 2013 and June 2015. It was created for the purpose of classifying emotions, expressed on Twitter towards arts and cultural experiences in museums.
It contains 3,085 tweets, with 4 emotions namely anger, happiness, surprise and sadness. Out of them, we used 2,500 tweets for training the GMM and the rest 1,085 were used for the testing of the model. This was done in multiple folds with a change in training, validation and testing dataset each time for better results.

For the bipolar classification of tweets using SVM, the dataset is based on data from the following two sources:

- University of Michigan Sentiment Analysis competition on Kaggle
- Twitter Sentiment Corpus by Niek Sanders

The Twitter Sentiment Analysis Dataset contains 1,578,627 classified tweets, each row is marked as 1 for positive sentiment and 0 for negative sentiment.

RESOURCES AND TOOLS

In this work we use various external resources in order to preprocess the data and provide prior score for some of the commonly used words:

- Emoticon Dictionary: We use the emoticons list extracted from a CLARIN.SI repository titled Emoji Sentiment Ranking 1.0 which contains a lexicon of 751 emoji characters (in Unicode format) with automatically assigned sentiment. The sentiments are computed from 70,000 tweets, labeled by 83 human annotators in 13 European languages. The table below shows a small snapshot of the dictionary:

Emoticon	Polarity
: −) :) : o) :] : 3	Positive
: −D : D 8D xD XD	Extremely Positive
: −/ : / = / = < /3	Negative
D: D8 D= DX v.v Dx	Extremely Negative
>:) B) B-) :) :−) >	Neutral

- Acronym Dictionary: We created a custom acronym dictionary from scratch which sits in the *utils* module. We created this using information regarding the most used acronyms on the internet for texting, tweeting and other social network updates. The table below is a snapshot of the acronym dictionary:

Acronym	Expansion
admin	administrator
afaik	as far as I know
omg	oh my god
rofl	rolling on the floor laughing
rip	rest in peace

- Sklearn -Scikit-learn (formerly scikits.learn) is a free software machine learning library for the Python programming language. It features various classification, regression and clustering algorithms including support vector machines, random forests, gradient boosting, k-means and and DBSCAN, and is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy.
- Nltk - The Natural Language Toolkit, or more commonly NLTK, is a suite of libraries and programs for symbolic and statistical natural language processing(NLP) for English written in the Python programming language.
- Python spell checker - It tries to choose the most likely spelling correction for a word. Probabilities are used to select most likely spelling.
- SentiWordNet - SentiWordNet is a lexical resource for opinion mining. SentiWordNet assigns to each synset of WordNet three sentiment scores: positivity, negativity, objectivity.
- Tweepy - It is an easy to use Python library for accessing the twitter API.
- Hashtag segmenter

PREPROCESSING

A. Tokenisation -

After downloading the tweets using the tweet id's provided in the dataset, we first tokenize the tweets. This is done using the Tweet-Tokenizer provided by the

NLTK library. It is important to note that this is a twitter specific tokenizer in the sense that it tokenize the twitter specific entries like Emoticons, Hashtag and Mentions too. After obtaining the tokenized tweet we move to the next step of preprocessing.

B. Replacing Emoticons -

Emoticons play an important role in determining the sentiment of the tweet. Hence we replace the emoticons by their sentiment polarity by looking up in the Emoticon Dictionary that we have created using Emoji Sentiment Dataset used in the 'emoji' python library.

C. Remove Url-

The url's which are present in the tweet are shortened using TinyUrl due to the limitation on the tweet text. These shortened url's did not carry much information regarding the sentiment of the tweet. Thus these are removed.

D. Remove Target-

The target mentions in a tweet done using '@' are usually the twitter handle of people or organisation. This information is also not needed to determine the sentiment of the tweet. Hence they are removed.

E. Replace Negative Mentions-

Tweets consists of various notions of negation. In general, words ending with 'nt' are appended with a not. Before we remove the stopwords 'not' is replaced by the word 'negation'. Negation play a very important role in determining the sentiment of the tweet. This is discussed later in detail.

F. Hashtags-

Hashtags are basically summariser of the tweet and hence are very critical. In order to capture the relevant information from hashtags, all special characters and punctuations are removed before using it as a feature.

G. Sequence of Repeated Characters-

Twitter provides a platform for users to express their opinion in an informal way. Tweets are written in random form, without any focus given to correct structure and spelling. Spell correction is an important part in sentiment analysis of user-generated content. People use words like 'cooooool' and 'hunnnnngry' in order to emphasise the emotion. In order to capture such expressions, we replace the sequence of more than three similar characters by three characters. For example, wooooow is replaced by wooow. We replace by three characters so as to distinguish words like 'cool' and 'cooooool'.

H. Numbers-

Numbers are of no use when measuring sentiment. Thus, numbers which are obtained as tokenized unit from the tokenizer are removed in order to refine the tweet content.

I. Nouns and Prepositions-

Given a tweet token, we identify the word as a Noun word by looking at its part of speech tag given by the tokenizer. If the majority sense (most commonly used sense) of that word is Noun, we discard the word. Noun words don't carry sentiment and thus are of no use in our experiments. The same reasoning go for prepositions too.

J. Stop-word Removal -

Stop words play a negative role in the task of sentiment classification. Stop words occur in both positive and negative training set, thus adding more ambiguity in the model formation. And also, stop words don't carry any sentiment information and thus are of no use to us. We create a list of stop words like he, she, at, on, a, the, etc. and ignore them while scoring the sentiment.

APPROACH

In Twitter sentiment analysis we build baseline model and feature based model. Figure 1 and 2 represent the approach of our training and testing model.

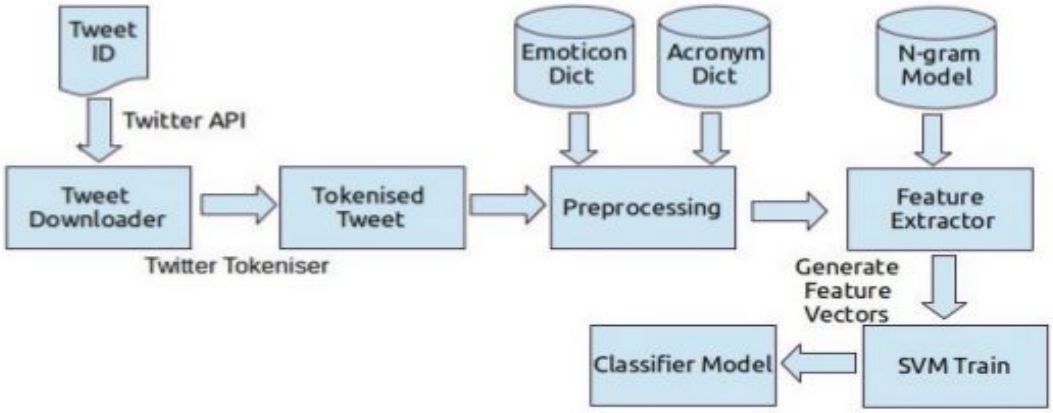

Figure 1. Flow Diagram of Training: Hybrid Model

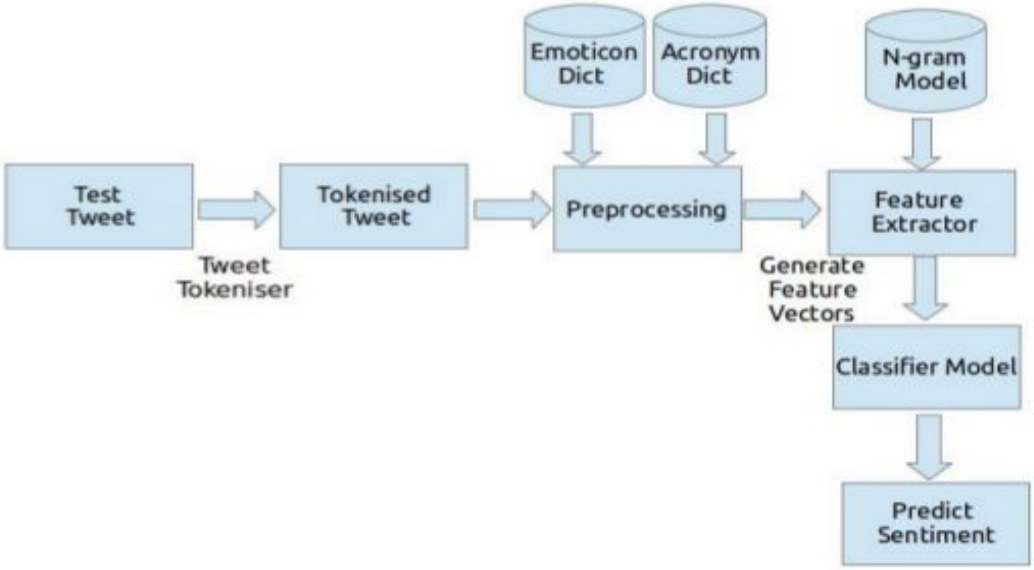

Figure 2. Flow Diagram of Testing: Hybrid Model

For multipolar data, instead of SVM Train, we train a Gaussian Mixture Model using the base and feature model and later instead of absolute classification and accuracy scores we use posterior probabilities obtained and log likelihood to measure relative accuracy of the model. Hence for the multipolar data, we are comparing which features are better for such an analysis instead of absolute accuracies.

A. Baseline Model

In the baseline approach, we first clean the tweets. We perform the preprocessing steps listed in section 4. This gives us tokenized text, tokenized hashtags, emoticons. Now we create a feature vector of tokens which can distinguish the twitter to be positive, negative or neutral with high confidence. For example, presence of tokens like, “horrible day”, “feeling blessed”, etc help in determining that the tweet carries positive, negative or neutral sentiment with high confidence. We call such words as **Emotion Determiner**.

$\mathbb{N}$	Polar	POS	# of (+/-) POS (JJ, RB, VB, NN)	f_1
		Other	# of negation words, positive words, negative words	f_2
			# of extremely-pos., extremely-neg., positive, negative emoticons	f_3
			# of (+/-) hashtags, capitalized words, exclamation words	f_4
	Non-Polar	POS	# of JJ, RB, VB, NN	f_5
		Other	# of slangs, latin alphabets, dictionary words, words	f_6
			# of hashtags, URLs, targets, newlines	f_7
$\mathbb{R}$	Polar	POS	For POS JJ, RB, VB, NN, $\sum$ prior pol. scores of words of that POS	f_8
		Other	$\sum$ prior polarity scores of all words	f_9
	Non-Polar	Other	percentage of capitalized text	f_{10}
$\mathbb{B}$	Non-Polar	Other	exclamation, capitalized text	f_{11}

Table 4: $\mathbb{N}$ refers to set of features whose value is a positive integer. They are primarily count features; for example, count of number of positive adverbs, negative verbs etc. $\mathbb{R}$ refers to features whose value is a real number; for example, sum of the prior polarity scores of words with part-of-speech of adjective/adverb/verb/noun, and sum of prior polarity scores of all words. $\mathbb{B}$ refers to the set of features that have a boolean value; for example, presence of exclamation marks, presence of capitalized text.

B. Feature Model

Next, we determine the attributes which would play a role in determining the polarity of any tweet. Table 4 shows a reference of features we have used. We have in total **21 feature** as listed below:

#f0: Percentage of capitalized text

#f1: COMPLETELY CAPITAL text (Emphasis words)

#f2: Capital + exclamation presence in a tweet

#f3-f6: No. of nouns, adjectives, verbs and adverbs.

#f7-10: No. of positive [nouns, adj, verbs, adv]

#f11-14: No. of negative [nouns, adj, verbs, adv]

#f15: Summation of prior polarity scores of words of that POS (Baseline)

#f16: Sentence polarity

#f17-18: No. of [positive hashtags, negative hashtags]

#f19: No. of hashtags

#f20: Emoticon polarity score

ERROR HANDLING

A. **Bipolar Sentiment Model:**

We use SVM to classify our tweet. SVM classifier is not a generative model and hence the binary classification is leading to more errors. Sentiments can be better judged using a generative model with probabilistic output and probability loss functions.

Also, sentiment being a very sophisticated portion of Language related research, we could use many other NLP applied features/techniquest to further boost the accuracy of our model.

B. **Multipolar Sentiment Model:**

We are getting accuracies as high as 92% but this is only at a labelled data of size approximately 4000 tweets. Not much work has been done around analyzing tweet as sad, happy, angry, etc sentiments and hence it required a great effort to even get a labelled data of 4000 tweets. We are getting an accuracy of around 90% on 60-40 train-test distribution.

The accuracies also vary with the initialization of covariance matrix while using GMM model. The accuracies can further improve by adding bi-gram or trigram features and simultaneously using more NLP techniques to get more features.

EXPERIMENTS AND RESULTS

We perform the following experiments (for both bipolar and multipolar sentiment analysis)

- Baseline model
- Feature based model
- Baseline + feature model

The results are summarized in the following tables:

A. Bipolar classification using SVM classifier (Accuracy score is the criterion for performance)

Model Used	**Accuracy**
Baseline	58.24
Baseline Model + f_{16}	71.65
Baseline Model + f_{16} + f_0 f_1 + f_2	75.24
Baseline Model + f_{16} + f_3 f_4 +f_5 +f_6	79.20
Baseline Model +f_{16}+ f_{17} f_{18} +f_{20} +f_7 + f_8 + f_9 + f_{10} + f_{11} + f_{12} + f_{13} + f_{14}	80.10
Baseline Model +f_{16}+f_7 + f_8 + f_9 + f_{10} + f_{11} + f_{12} + f_{13} + f_{14} + f_{19} + f_0 +f_1 + f_2 + f_3	80.01
Baseline Model +f_{16}+f_7 + f_8 + f_9 + f_{10} + f_{11} + f_{12} + f_{13} + f_{14} + f_{19} + f_0 +f_1 + f_2 + f_3+f_{17} f_{18} +f_{20}	82.34
Baseline Model +f_{16}+f_7 + f_8 + f_9 + f_{10} + f_{11} + f_{12} + f_{13} + f_{14} + f_{19} + f_0 +f_1 + f_2 + f_3+ f_{17} f_{18} +f_{20} + f_4 +f_5 + f_6 +f_7	**82.52**

Kernel Used	**Peak Accuracy(with best set of hyperparameters and features)**
linear	78.56
rbf	**82.52**

Poly (degree 3)	81.54
Poly (degree 2)	76.49

B. Multipolar classification using GMM (Log likelihood of the model is the criteria for performance)

Model Used	**Log likelihood obtained**
Baseline	65.24
Baseline Model + f_{16}	78.10
Baseline Model + f_{16} + f_0 f_1 + f_2	78.10
Baseline Model + f_{16} + f_3 f_4 +f_5 +f_6	79.00
Baseline Model +f_{16}+ f_{17} f_{18} +f_{20} +f_7 + f_8 + f_9 + f_{10} + f_{11} + f_{12} + f_{13} + f_{14}	**88.46**
Baseline Model +f_{16}+f_7 + f_8 + f_9 + f_{10} + f_{11} + f_{12} + f_{13} + f_{14} + f_{19} + f_0 +f_1 + f_2 + f_3	87.16
Baseline Model +f_{16}+f_7 + f_8 + f_9 + f_{10} + f_{11} + f_{12} + f_{13} + f_{14} + f_{19} + f_0 +f_1 + f_2 + f_3+f_{17} f_{18} +f_{20}	84.20
Baseline Model +f_{16}+f_7 + f_8 + f_9 + f_{10} + f_{11} + f_{12} + f_{13} + f_{14} + f_{19} + f_0 +f_1 + f_2 + f_3+ f_{17} f_{18} +f_{20} + f_4 +f_5 + f_6 +f_7	84.10

As we can clearly see, maximum likelihood is better when we consider the polarity scores of tokens and not just the count of capitalized words or count of nouns, adjectives, etc. This is the reason why we use Emotion Determining features.

References:

- Sentiment Analysis of Twitter Data: A Survey of Techniques
- Twitter as a Corpus for Sentiment Analysis and Opinion Mining
- Natural Language Processing with Python – Analyzing Text with the Natural Language Toolkit
- Brendan O'Connor, Michel Krieger, and David Ahn. TweetMotif: Exploratory Search and Topic Summarization for Twitter. ICWSM-2010.
- Python spell checker
- Twitter Sentiment Analysis- The good, the bad, the neutral!
- SVM: Scikit-learn
- Sentiment Analysis of Twitter Data - Columbia University
- Sentiment Analysis on Multi-View Social Data - University of Ottawa
- Twitter Sentiment Analysis: The Good the Bad and the OMG!
- Github-code