E1:246 - Natural Language Understanding - Sentiment Analysis of Twitter Data

Anirban Biswas [1] **Nihar Ranjan Sahoo** [1] **Monish Keswani**[1]

Abstract

With the rise of social networking age, there has been a rise in user generated content. One such source of data is the social networking sites. Twitter is one such online news and social networking service on which users express their opinions and display their sentiments with their posts that are commonly known as "tweets". Sentiment analysis is an approach to computationally identify and categorize opinions expressed in a piece of text. Twitter Sentiment Analysis is an application to extract the sentiments conveyed by the user on the social networking platform. The research in this field has grown consistently. One identified difficulty in the analysis is to process the tweets which have strict constraints on character length is the analysis of slang words and abbreviation. We have done a comparative study of different methods for sentiment analysis on twitter data.

Keywords: Twitter, Sentiment Analysis, Natural Language Processing

1 Motivation

Sentiment analysis allows individuals or business to be proactive as opposed to being reactive when any negative conversational thread is emerging. Alternatively, positive sentiment can be identified thereby allowing the identification of product advocates or to see which parts of a business strategy are working. In the past decade, new forms of communication, such as micro-blogging and text messaging have emerged and become ubiquitous. While there is no limit to the range of information conveyed by tweets and texts, often these short messages are used to share opinions and sentiments that people have about what is going on in the world around them. We have chosen to work with Twitter since it is a better approximation of public sentiments as opposed to conventional internet articles and web blogs. The amount of relevant information available is much larger for twitter as compared to traditional blogging sites.

2 Problem Statement

Given a message also known as "Tweet", classify whether the tweet is of positive or negative sentiment. For tweets conveying both a positive and negative sentiment, the sentiment that is stronger should be chosen

3 Introduction

Sentiment is defined as *"a personal belief or judgment, whose foundation does not rely on any proof or certainty"*. With the rise in social networking and micro blogging sites users from all over the world have been using these platforms actively to convey their feeling and views about the various happenings around the world. Automated identification of these sentiment can be useful for many systems like media analysis, dialogue systems etc. It can also be used by businesses for the market analysis of a product where sites like Twitter help in collecting the feedback and review of product and services. With the recent popularity of the Twitter micro-blogging service, a huge amount of frequently self-standing short textual sentences (tweets) became openly available for the research community. Sentiment analysis has been put in use by many people to extract out sentiments from these short messages. A lot of research is being carried out on Twitter data for classification of the tweets and analysis of the results thereafter. In this project, we have studied and reviewed some of the research work in this domain.

4 Previous Works

(Parikh and Movassate, 2009) have implemented Naive Bayes unigram model, Naive Bayes bigram model and Maximum Entropy model for the classification of tweets.In their findings they have claimed that Naive Bayes model worked better than the Maximum Entropy model. (Go et al., 2009) proposed a solution in which tweets consisted of emoticons as well by using distant supervision. They have used Naive Bayes, Maximum Entropy and Support Vector Machine (SVM) for classification . The emoticons were served as noisy labels. The SVM outperformed other models.

(Apoorv Agarwal, 2011) has approached this problem as a three way task of classifying sentiment into positive, negative and neutral. The three types of models that were used are unigram model, tree kernel based model and feature based model. Unigram model used around 10,000 features and feature based model used 100 features. They found that features combining prior polarity of words and POS tags are most important for classification task. The tree kernel based method performed significantly better than the other two.

The Sentiment analysis can be done at different levels of granularity - word level, phrase or sentence level and feature level. The format of tweet is constrained to few characters hence the word level granularity suits the setting. In this project, we have tried both sentence level and word level approach.

5 Characteristics of tweets

Twitter messages have unique attributes which help in differentiating sentiment analysis from other domains.

1. **Message Length:** The maximum length of the tweet is 140 characters which makes it different from other sentiment classification domains which focus on longer bodies such as movie reviews. From our training set, we have calculated the average length of a tweet which is 19 words .

2. **Real time:** Limiting the characters in a tweet makes it favourable for frequent updates unlike blogs which are longer in nature and writing them takes time.

Table 1: Emoticon Tagger

Emoticons	Examples					
SMILEY	:-)	:)	(:	(-:		
LAUGH	:-D	:D	X-D	XD	xD	
LOVE	<3	:*				
WINK	;-)	;)	;-D	;D	(;	(-;
FROWN	:-(	:(	(:	(-:		
CRY	:,(	:'(	:"(	:((		

3. **Topics:** Twitter messages cover range of topics unlike other sites which focus on a particular topic making it suitable for data analysis.

6 Preprocessing

6.1 Twitter Feature Removal

The first step in preprocessing is to remove the URLs (www.example.com) and targets (@username) as they do not convey any information about the polarity of the sentiment, hence their presence is fruitless.

Regular Expression for URL:
((www\.[\S]+)—(https?://[\S]+))
The URLs are replaced with space

Regular Expression for targets: @[\S]+
The targets are replaced with space

6.2 Emoticons

Emoticon is a representation of a facial expression such as a smile or frown, formed by various combinations of keyboard characters and used in electronic communications to convey the user's feelings or intended tone. Each emoticon is given a positive or a negative tag based on their meaning. Smiley, laugh, love and wink are given positive tag while frown and cry are given negative tag. We have maintained a dictionary which stores the tag of each emoticon.

6.3 Hashtags

The ideal way to separate out words is to separate them via capital letters and look them up in the dictionary but users do not always follow the convention of creating a correct "hashtag" thus it is difficult to extract out the words from the hashtags when there is no such separation. We used an alternative approach for extraction which is to read characters from a hashtag till a word

Table 2: Number of Features per tweet

	SemEval-2013		SemEval-2015	
Features	Avg	Max	Avg	Max
Handles	0.42	8	0.49	4
Hashtags	0.26	12	0.32	5
URLs	0.22	2	0.19	2
Emoticons	0.08	3	0.08	2
Words	19.53	35	19.65	31

Table 3: Number of Features per tweet

	SemEval-2016		Combined	
Features	Avg	Max	Avg	Max
Handles	0.37	6	0.4	8
Hashtags	0.25	7	0.26	12
URLs	0.36	3	0.27	3
Emoticons	0.0305	6	0.06	6
Words	19.43	34	19.50	35

highest possible length is matched in dictionary. It will so happen that the prefixes will be separated out as different words and can be handled with some modification.

Raw string: #iloveyou
Processed string: i love you

6.4 Multiple Frequency words

We handle multiple frequency words such as *Greatttttt* by converting them to double frequency words and searching for them in the word dictionary as well acronym dictionary. If the double frequency words are not present in either of the dictionaries then they are converted to single frequency words and same procedure is repeated till a match is found in either dictionary.

Regular Expression: (.) \1+
Replaced Expression: \1 \1

6.5 Acronyms

The limitation on the number of characters in the tweet has promoted the use of acronyms. The acronym dictionary helps in expanding the tweet text and thereby improves the overall sentiment score. The acronym dictionary has 5697 entries.

Raw string: asap
Processed string: as soon as possible

6.6 Miscellaneous tasks

We are using case insensitive analysis, we take two occurrence of same words as same due to their sentence case in-sensitiveness. Hence upper case words are converted to lower case words.The words like *can't* are converted to *can not*.The punctuation and numbers are removed as they do not contain any information about polarity of sentiment. After all the tasks the non-English words are removed from the data. The words which occur less than 3 times in the entire dataset are removed as they hardly convey any useful information for classification. We have used porter stemmer as stemming algorithm. Lemmantization has also been done to normalize a word.

Table 4: My caption

	Hindi Corpus	
Features	Avg	Max
Handles	0.13	7
Hashtags	0.25	7
URLs	0.01	2
Emoticons	0.07	19
Words	14.32	39

The summary of the entire preprocessing is given in Table 5

7 Methodologies

We use different features and machine learning classifiers to determine the best combination for sentiment analysis of twitter. Figure 1 illustrates the steps taken in the entire process.

Table 5: List of Preprocessing steps

Remove URLs
Remove Target mentions
Replace Emoticons
Hashtags seperation
Handle sequence of repeated characters
Replacing Acronyms
Converting to Lower case
Remove numbers
Remove punctuations
Remove Non-English Tweets
Remove Stop-words

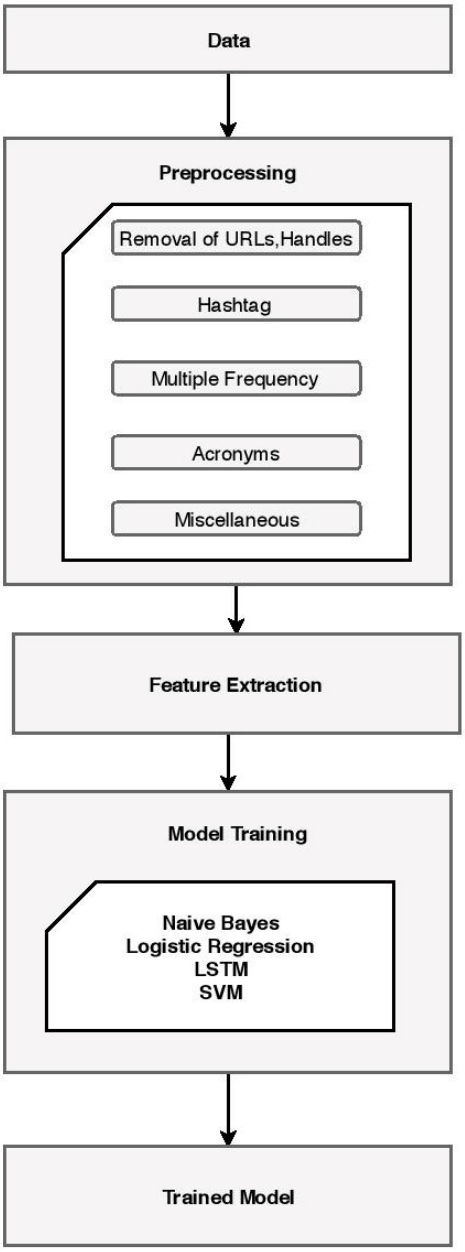

Figure 1: Architecture

7.1 Naive Bayes classifier

Naive Bayes classifier assumes that all the features are independent. Though the assumption may look too strong, it is particularly suited when the dimensionality is quite high. We are given a set of variables, $X = \{x_1, x_2, x_1, ..., x_N\}$, we want to construct the posterior probability for the event C_j among a set of possible outcomes $C = \{c_1, c_2, c_1, ..., c_d\}$. X is the feature set and C is the set of sentiment classes.

$$P(X/c_j) = \prod_{i=1}^{N} P(x_i/c_j)$$

$$P(c_j/X) \propto P(c_j) * \prod_{i=1}^{N} P(x_i/c_j) \quad (1)$$

$$\hat{y} = argmax_j P(c_j) * \prod_{i=1}^{N} P(x_i/c_j) \quad (2)$$

Naive Bayes classifier is not suitable when the dependency of the features is high. Though the accuracy of Naive Bayes is less, it can converge faster than other models, thus providing better understanding of features in the early stage.

7.2 Logistic Regression

Logistic Regression is linear model of classification. It is also known as Maximum Entropy classifier or log-linear classifier. In this model, probabilities of possible outcomes are modeled using logistic function. The objective function of a binary classification using logistic function is given below.

$$min_{w,b} \frac{1}{2} w^T w + C \sum_{i=1}^{n} log(exp(-y_i(X_i^T w + c)) + 1) \quad (3)$$

The above equation is the L2-regularized Logistic Regression equation where C is the regularization parameter. Naive Bayes is a generative classifier as it can model the joint probability $P(x, y)$ which tries to predict the likelihood under the assumption of conditionally independence between features, while logistic regression is a discriminative classifier which models the conditional probability $P(y/x)$. It is expected that Logistic Regression should outperform Naive Bayes classifier.

7.3 Support Vector Machines

Support Vector Machines are supervised learning models that analyze data for classification and regression. We are given a set of training vectors $x_i \in R^p, i = 1, ..., n$, in two classes, and a vector y $\in \{1, -1\}^n$, SVM solves the following primal problem:

$$\begin{aligned} min_{w,b,\xi} \ \ & \frac{1}{2} w^T w + C \sum_{i=1}^{n} \xi_i \qquad (4) \\ subject\ to \ \ & y_i(w^T x_i + b) \geq 1 - \xi_i \\ & \xi_i \geq 0, \ i = 1, 2, 3..., n, \end{aligned}$$

The decision function is given by

$$sgn(w^T x + b) \qquad (5)$$

As our problem is multiclass classification problem. We use *One-Vs-Rest* approach for classification.

7.4 LSTM

The Long Short-Term Memory (LSTM) architecture was designed to solve the vanishing gradients problem in RNNs, and is the first to introduce the gating mechanism. An LSTM has three of gates, to protect and control the cell state, an input gate, an output gate and a forget gate. The input gate controls the extent to which a new value flows into the cell, the forget gate controls the extent to which a value remains in the cell and the output gate controls the extent to which the value in the cell is used to compute the output activation of the LSTM unit.

Each LSTM cell can be computed as follows

$$X = \begin{bmatrix} h_{t-1} \\ x_t \end{bmatrix} \qquad (6)$$

$$f_t = \sigma(W_f \Delta X + b_f) \qquad (7)$$

$$i_t = \sigma(W_i \Delta X + b_i) \qquad (8)$$

$$o_t = \sigma(W_o \Delta X + b_o) \qquad (9)$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \tanh(W_c \Delta X + b_c) \qquad (10)$$

$$h_t = o_t \odot \tanh(c_t) \qquad (11)$$

Table 6: Distribution of number of positive, negative and neutral tweets for semeval-2016 data

	Positive	Negative	Neutral
Train set	3640	1458	4586
Test set	7059	3231	10342

Table 7: Distribution of number of positive, negative and neutral tweets for semeval-2016 data

	Positive	Negative	Neutral
Train set	3249	4644	2354
Test set	780	1179	603

where $W_i, W_f, W_o \in R^{d*2d}$ are the weighted matrices and $b_i, b_f, b_o \in R^d$ are biases that will be learned during training. is the sigmoid function and $\odot$ stands for element-wise multiplication. x_t includes the inputs of LSTM cell unit, representing the word embedding vectors w_t The vector of hidden layer is h_t.

7.5 Ensemble Classifier

The classifiers discussed above are used as the base classifiers to do the best classification. The data is classified based on the output of majority of classifiers.

8 Datasets

The following datasets have been used for model evaluations:

1. **SemEval-2016 Task-4:** It consists around 20000 labelled tweets for the "Message Polarity Classification "problem. This dataset contains only english tweets. Summary of this dataset are shown in Table **??**
2. **SAIL Codemixed - 2017:** A collection of 12000 tweets that contain both hindi and english data. The tweets are labelled positive or negative according to the emoticon. Summary of this dataset are shown in Table **??**

9 Experimental Evaluation

In this section, we experimentally evaluate the performance of all the proposed methods described earlier and compare the results with each of them. We used 2 real life datasets which contain three types of tweets - depicting positive, negative and neutral sentiments.

9.1 Evaluation Metric

We take two popularly used evaluation criteria - *Weighted F1-score* and *Accuracy*. Normally, the higher the values are, the better the classification performance is.

Mathematically,

$$Acc(\hat{\mathcal{C}}, \mathcal{C}) = \frac{\sum_{i=1}^{n} I(\hat{\mathcal{C}}_i, \mathcal{C}_i)}{n} \quad (12)$$

Here $\mathcal{C}$ is the ground truth labeling of the dataset such that $\mathcal{C}_i$ gives the ground truth label of ith data point. Similarly $\hat{\mathcal{C}}$ is the label assignments predicted by some algorithm. We assume I to be the identity function on R^2, defined as $I(a, b) = 1$ if $a = b$ and $I(a, b) = 0$ if $a \neq b$.

Similarly, F1-score is defined as :

$$F_1 = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

9.2 Experimental Setup

In our experiments, we have tried to clean data by processing it using various techniques e.g. hashtag-process, appropriate handling of urls and user-mentions and stopword removal. After processing the data, we have move on to the task of message polarity classification.

For this task, we have encoded the sequence of processed tweets using one-hot encoding technique, tf-idf based methods. We have also tried PCA based approaches. For classification tasks, we have used Naive Bayes model, Ada-Boost, Multilayer Perceptron and LSTM-based models. Finally we have used an ensemble of classifiers results.

9.3 Nature of Data

We have had a closer look at the data distribution for both the datasets given. We have generated the frequency count for words before and after processing the data. The following graph shows the nature of distributions:

We can easily figure out the similarities in nature of data. All of these follow the power-law distribution and nature of data distribution does not change even after processing. In general, in case of power law distributed data, both low frequency and high frequency words don't posses much information. Most of the information are captured by mid-frequency range words. Hence we remove those words as a step in our pre-processing stage.

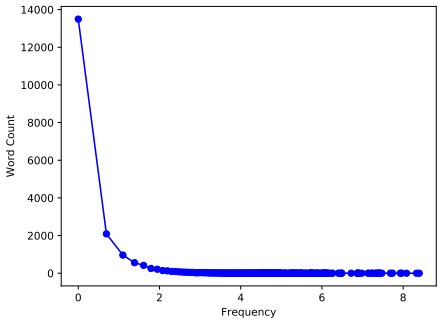

Figure 2: Frequency distribution of unprocessed semeval data

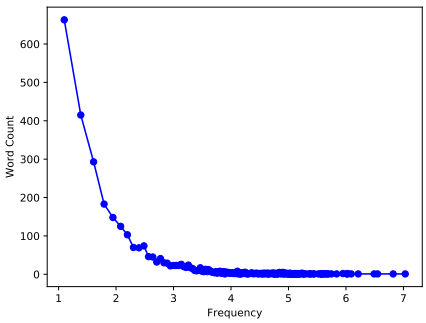

Figure 3: Frequency distribution of processed semeval data

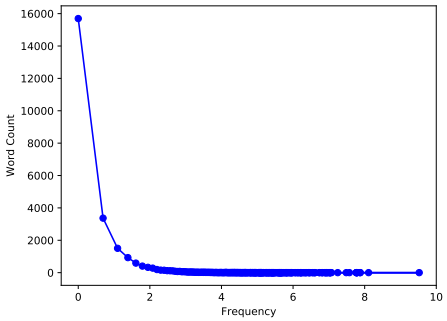

Figure 4: Frequency distribution of unprocessed SAIL data

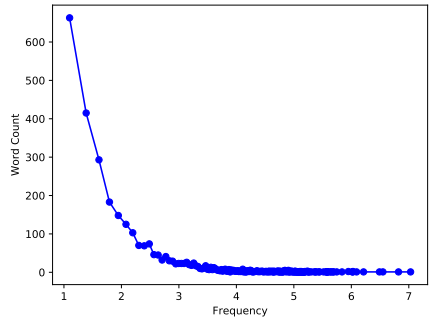

Figure 5: Frequency distribution of processed SAIL data

9.4 Data Augmentation

In our semeval task, from Table **??**, we can see that the test data size is much larger than train data. We have run experiments by training our classifiers using this set of data and we did not get satisfactory results. Hence, we tried to augment data. We found our semeval-2013 and semeval-2015 dataset and augmented those with out current dataset and we have got some improvements to the results as we can see it in results section.

9.5 Solution Approaches

We have experimented with different features like frequency of words, emoticons etc. To generate embeddings of tweets, we have done one-hot encoding, if-idf based encoding. We have also tried PCA based approach. Finally, we tried to combine results for our best models for and take an ensemble of all those.

- **One-hot encoding** - We have used one-hot encoding technique to generate the embedding for tweets. If a certain word is present, the corresponding feature will be assigned 1 else it is assigned zero. The feature dimension equals the vocabulary size
- **tf-idf encoding** - It is term frequency-inverse document frequency based technique.
- **Count based encoding** - This is similar to one-hot, only difference is here the frequency of word is used instead of binary values.
- **PCA based technique** - We have used PCA which is a dimensionality reduction technique. We have reduced the number of dimension of feature vector in such a way that maximum information is retained.
- **Ensemble of classifiers** - In this approach, we tried to combine the classifier results and take the prediction which occurs with highest frequency in all classifier results.

9.6 Results

For semeval data, from Table 8, we can see that in terms of accuracy, LSTM performing best whereas in terms of average F1-score, ensemble approach is giving best resutls.

For SAIL codemixed data, from Table 9, we can see similar trends.

10 Future Work

In the preprocessing step we have used maximal matching method to split the Hashtags. The results which we have obtained are not so accurate according to the context. We will try new methods to improve the accuracy. Currently our approach doesn't detect sarcasm which express negative opinion about a target using positive words. Advanced models have to be used to deal with Sarcasm. Sentiment for a particular entity in the tweet is currently not handled. There is a need to analyze the sentiment towards it. We have tested our approach on LSTMs. The next step would be to test our approach on Attention models. The datasets available for Hindi are few in number and are very unstructured . Synthetic data has to be generated to deal with the scarcity of data.

References

Ilia Vovsha Owen Rambow Rebecca Passonneau Apoorv Agarwal, Boyi Xie. 2011. Sentiment analysis of twitter data. In *Proceedings of the Workshop on Languages in Social Media*, pages 30–38.

Alec Go, Richa Bhayani, and Lei Huang. 2009. Twitter sentiment classification using distant supervision. 150.

Ravi Parikh and Matin Movassate. 2009. Sentiment analysis of user-generated twitter updates using various classication techniques.

Table 8: Performance of the algorithms for Classification on the semeval data. We have tried the methods using tf-idf, one-hot and pca.

Metric	Classifier	Methodology			
		one-hot	tf-idf	pca	other
Accuracy	Naive Bayes	0.51	.58	.38	-
	Max Entropy	0.50	.52	.37	-
	MLP	0.52	.57	.40	-
	SVM	0.48	.48	.38	-
	Ada Boost	0.36	.59	.35	-
	Random Forest	0.34	.59	.39	-
	LSTM	-	-	-	**0.67**
	Ensemble	-	-	-	0.60
Average-F1	Naive Bayes	0.49	.57	.34	-
	Max Entropy	0.47	.52	.35	-
	MLP	0.49	.56	.38	-
	SVM	0.47	.48	.36	-
	Ada Boost	0.23	.53	.35	-
	Random Forest	0.17	.52	.39	-
	LSTM	-	-	-	0.49
	Ensemble	-	-	-	**0.59**

Table 9: Performance of the algorithms for Classification on the SAIL(hindi-english code-mixed) data. We have tried the methods using tf-idf, one-hot and pca.

	NB	Max.Ent.	SVM	MLP	AdaBoost	RF	LSTM	Ensemble
Accuracy	0.55	0.51	0.48	0.54	0.55	0.45	**0.67**	0.56
Average-F1	0.54	0.50	0.48	0.53	0.29	0.51	0.46	**0.56**