Sentiment Analysis of Hindi-English Code-Mixed Text

Kshitij Kumar Singh 2018124

Suyash Khare 2018257

Karan Tiwari 2018114

Abstract-- In this report, we address the problem of sentiment classification on twitter dataset. We use a number of machine learning techniques to perform sentiment analysis. At the end, then by using different classifiers, namely, Naïve Bayes',, Bernouille's Naïve Bayes', Logistic Regression, Stochastic Gradient Descent, Support Vector Machines and Maximum Entropy classifiers are trained on 85% of the dataset and tested over 15% of the remaining dataset..

INTRODUCTION

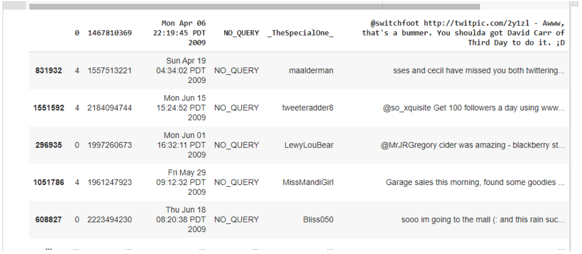
Code-mixing refers to the use of linguistic units words, phrases, clauses from different
languages at a sentence or utterance level. Sentiment analysis is related to the classification of emotions in text data using various text analysis and Machine learning techniques.Codemix can be defined as mixing two or more languages.
Sentiment Analysis on CodeMixed Text has various real life applications such as customer satisfaction.

II. DATA

A. Dataset

The dataset which is used was obtained from ***'Kaggle'*** called the Sentiment140 dataset. It contains 1.6 Million tweets extracted using the twitter API. The tweets have been annotated (0 = negative, 2 = neutral, 4 = positive) and they can be used to detect sentiment. But only 100000 rows were randomly selected from this dataset, with almost equal distribution of positive and negative tweets and neutral tweets were also considered as positive as objective was binary classification i.e positive or negative sentiments.

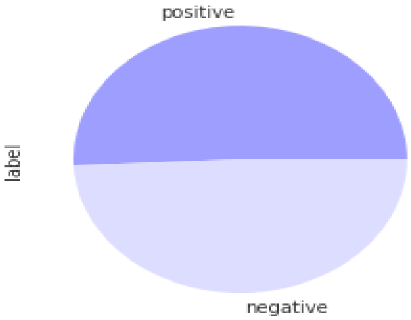
Snippet of dataset shown below:



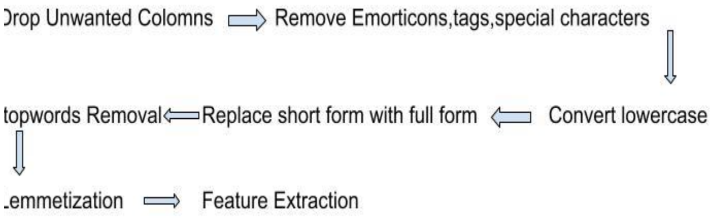
	0	1467810369	Mon Apr 06 22:19:45 PDT 2009	NO_QUERY	_TheSpecialOne_	@switchfoot http://twitpic.com/2y1zl - Awww, that's a bummer. You shoulda got David Carr of Third Day to do it. ;D
831932	4	1557513221	Sun Apr 19 04:34:02 PDT 2009	NO_QUERY	maalderman	sses and cecil have missed you both twittering...
1551592	4	2184094744	Mon Jun 15 15:24:52 PDT 2009	NO_QUERY	tweeteradder8	@so_xquisite Get 100 followers a day using www...
296935	0	1997260673	Mon Jun 01 16:32:11 PDT 2009	NO_QUERY	LewyLouBear	@MrJRGregory cider was amazing - blackberry st...
1051786	4	1961247923	Fri May 29 09:12:32 PDT 2009	NO_QUERY	MissMandiGirl	Garage sales this morning, found some goodies ...
608827	0	2223494230	Thu Jun 18 08:20:38 PDT 2009	NO_QUERY	Bliss050	sooo im going to the mall (: and this rain suc...
...	...	...	...	...	...	...

As visible above, it has 6 columns containing: target class(i.e. Sentiment polarity), tweet _id , date&time , NO_Query, name and Tweet.

Our dataset has following positive and negative classification:



B. DATA PREPROCESSING



Dataset from twitter generally is a noisy dataset. This is due to the casual nature of people's usage of social media. Tweets have certain special characteristics such as retweets, emoticons, user mentions, etc. which have to be suitably extracted. Therefore, raw twitter data has to be normalized to create a dataset which can be easily learned by various classifiers. We have applied an extensive number of pre-processing steps to standardize the dataset and reduce its size. We first do some general pre-processing on tweets which is as follows.

URL:

Users often share hyperlinks to other webpages in their tweets. Any particular URL is not important for text classification as it would lead to very sparse features.

Therefore, we replace all the URLs in tweets with the word URL. The regular expression used to match URLs is **((www\.[\S]+)|(https?://[\S]+))**.

User Mention:

Every twitter user has a handle associated with them. Users often mention other users in their tweets by @handle. We replace all user mentions with the word USER_MENTION. The regular expression used to match user mention is @[\S]+.

Emoticons:

Users often use a number of different emoticons in their tweet to convey different emotions. It is impossible to exhaustively match all the different emoticons used on social media as the number is ever increasing. However, we match some common emoticons which are used very frequently and replaced them with corresponding words expressing the meaning of it.

Hashtag:

Hashtags are unspaced phrases prefixed by the hash symbol (#) which is frequently used by users to mention a trending topic on twitter. We replace all the hashtags with the words with the hash symbol. For example, #hello is replaced by hello. The regular expression used to match hashtags is #(\S+).

Feature Extraction

Bag of words

A popular technique for developing sentiment analysis models is to use a bag-of-words model that transforms documents into vectors where each word in the document is assigned a score.

Certain features, like adjectives, abstract nouns, and adverbs were focused on and the rest of the words were removed as they did not add any value to the sentiment. This was done as part for the feature identification and extraction. This was implemented by checking each word in the cleaned tweet with the words in a file, which was prefilled with most common adjectives, adverbs and abstract nouns. If the word was not present in this file, it was not chosen as a feature and if it matched a word in this file, it was selected as one of the features. All these features of each text was stored in another column. This method was used because the existing method is not completely accurate and is outdated.

III. Research And Related Work

The process of sentiment analysis can be said to be a form of application that combines and applies the concept of text analytics, computational linguistics as well as natural language processing.

In the present communication-based society, no natural language seems to have been left untouched by the trends of code-mixing. For different communicative purposes, a language uses linguistic codes from other languages

Paper[1] introduces the Sub-Word Long Short Term Memory model to learn sentiments in a noisy Hindi-English Code Mixed dataset.

In Paper [2] , a mechanism was proposed for machine translation of Hinglish to pure (standard) Hindi and pure English form is presented.It makes use of a system designed specifically to separate out the Hindi and English parts of a word that has a combination of the two.

The strategy described here is equally applicable to all Indian languages as these are verb ending languages and have similar mixture of lexicons as in case of Hindi.

Paper[3] proposes to perform sentiment classification on Hinglish text written in Roman script using a triumvirate of TF-IDF, GR, and RBFNN, which is found as the best combination for classifying sentiment expressed in the Hinglish text.

Classifiers

Naive Bayes
Naive Bayes methods are a set of supervised learning algorithms based on applying Bayes' theorem with the "naive" assumption of conditional independence between every pair of features given the value of the class variable.

Bernoulli's Naïve Bayes' Classifier
implements the naive Bayes training and classification algorithms for data that is distributed according to multivariate Bernoulli distributions; i.e., there may be multiple features but each one is assumed to be a binary-valued (Bernoulli, boolean) variable. Therefore, this class requires samples to be represented as binary-valued feature vectors

Logistic Regression : Logistic Regression is a statistical method for analysing a dataset in which there are one or more independent variables that determine an outcome . The outcome is measured with a dichotomous variable (in which there are only two outcomes).

Stochastic Gradient Descent Classifier :
Stochastic Gradient Descent (SGD) is a simple yet very efficient approach to discriminative learning of linear classifiers under convex loss functions such as (linear) Support Vector Machines and Logistic Regression.
Stochastic gradient descent refers to calculating the derivative from each training data instance and calculating the update immediately . It is particularly useful when the
sample data is in a larger number. It supports different loss functions and penalties for classification.

Maximum Entropy
Maximum Entropy Classifier model is based on the Principle of Maximum Entropy. The main idea behind it is to choose the most uniform probabilistic model that maximizes the entropy, with given constraints. Unlike Naive Bayes, it does not assume that features are conditionally independent of each other. So, we can add features like bigrams without worrying about feature overlap. In a binary classification problem like the one we are addressing, it is the same as using Logistic Regression to find a distribution over the classes. The model is represented by

$$P_{ME}(c|d, \lambda) = (\exp[\sum_i \lambda_i f_i(c, d)]) \,/\, (\sum_{c'} \exp[\sum_i \lambda_i f_i(c, d)])$$

Here, c is the class, d is the tweet and λ is the weight vector. The weight vector is found by numerical optimization of the lambdas so as to maximize the conditional probability

Support Vector Classifier

SVM, also known as support vector machines, is a non-probabilistic binary linear classifier. For a training set of points (xi , yi) where x is the feature vector and y is the class, we want to find the maximum-margin hyperplane that divides the points with yi = 1 and yi = −1. The equation of the hyperplane is as follow

$$w \cdot x - b = 0$$

We want to maximize the margin, denoted by γ, as follows

$$\max_{w,\gamma} \gamma, \text{ s.t.} \forall i, \ \gamma \leq y_i(w \cdot x_i + b)$$

in order to separate the points well.

BILINGUAL SENTIMENT ANALYSIS

After Training and testing different classifiers, a translation mechanism is used to handle the challenge of the Hinglish text using the Google Translator Machine . A translation mechanism is used to handle the challenge of the Hinglish text using the Google Translator Machine and a function is created that takes text as input and translates it if required and tests it with each of the seven trained base models and the hybrid model and the sentiment predicted by each of these is printed as the output along with the features. The final master function also uses the package TextBlob to check whether the language of the word is Hindi or English and accordingly passes the word to the Translator function. The final string is passed onto a function which returns the desired output.

Model Results:

Models	Accuracy
Naïve Bayes' Classifier	71.58
Bernoulli Naïve Bayes' Classifier	72.61
Logistic Regression	75.82
Stochastic Gradient Descent Classifier	76.64
Support Vector Classifier	76.56
Maximum Entropy Classifier	77.48

CONCLUSION

The study conducted is unique in multiple ways which act as a differentiation factor when compared to similar work. The methodology followed, helped to create an aggregated model consisting of all the classifiers used during the process. The ensemble model created worked to our advantage as it provided the highest accuracy of **around 76%** compared to the classifiers individually. TextBlob was used to detect the language of the words in a sentence. If the word was in Hindi, Google Translate was used to directly convert it into English. Using this approach of both the platforms, increased the accuracy significantly when compared to using them individually.. If we were to look towards the future to enhance the study, we could find a **larger and better dataset** to work with. We can try to use better translation techniques and give a try at more complex machine learning models for the classification of text. The complexities of each of these models can also be found and compared to give the best efficiency. Furthermore, the study can be extended to **multiple regional**

languages and can be translated into a multilingual sentiment analysis problem.

References

[1] Aditya Joshi, Ameya Prabhu Pandurang, Manish Shrivatsava and Vasudeva Varma, “Towards Sub-Word Level Compositions for Sentiment Analysis of Hindi-English Code Mixed Text,” 26th International Conference on Computational Linguistics, December 2017.

[2] R. Mahesh, K. Sinha and Anil Thakur, “Machine Translation of Bilingual Hindi-English (Hinglish) Text,” January 2005.

[3] Kumar Ravi and Vadlamani Ravi, “Sentiment classification of Hinglish text,” March 2016.

[4]
https://en.wikipedia.org/wiki/Support-vector_machine

[5]
https://en.wikipedia.org/wiki/Stochastic_gradient_descent

[6]
https://scikit-learn.org/stable/modules/naive_bayes.html