Emotion Detection From Tweets

Utsav Baghela
utsav21101@iiitd.ac.in
MTech CSE, IIIT Delhi
New Delhi, India

Saurabh Pandey
saurabh21077@iiitd.ac.in
MTech CSE, IIIT Delhi
New Delhi, India

Kirpali
kirpali21040@iiitd.ac.in
MTech CSE, IIIT Delhi
New Delhi, India

Ekta Gambhir
ekta21025@iiitd.ac.in
MTech CSE, IIIT Delhi
New Delhi, India

Karan Singh
karan21038@iiitd.ac.in
MTech CSE, IIIT Delhi
New Delhi, India

1 ABSTRACT

Social Media Analysis provides an overview of people's opinions and sentiments towards certain entities. Users post their thoughts and insights on these networking sites.

The main aim of our project work is to comprehend and classify the tweets posted on Twitter, into five classes of emotions. Previously, a very broad classification scheme such as positive-negative, happy-sad, etc. has been developed by the NLP community. Since very little work is done on an in-depth classification of tweets into several classes, we decided it was a worthy problem to tackle.

With our work we seek to create a multi-class classification system that segregates tweets into one of several classes that are less general and offer a profound idea of the sentiment behind it.

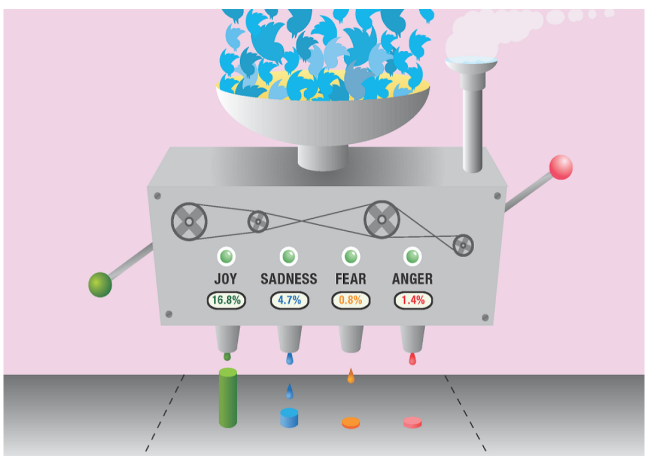

Figure 1: Tweets to Emotions

2 INTRODUCTION

More than a decade of being launched, Twitter now has an active user base of around 300 million people. The platform has a strong impact and influence over the world. Being used for awareness about social issues, to raise an opinion or to spread political messages, today's youth is very much involved in all that's going on.

There is a lot of interest in what people are tweeting about in a lot of domains like politics, business and climate change.

Every tweet made has an associated feeling and emotion which is termed as the Sentiment. Even with the character-limit of 280, a lot is expressed. Emoticons and hashtags are also the high impact forms of expression. Thus, a lot of work is being done to analyse the posts made on twitter and to determine the emotion behind them.

3 LITERATURE REVIEW

[1] The team built two SVM classifiers (obtaining the most optimal parameters using cross-validation), one to detect the sentiment of messages such as tweets and SMS (message-level task) and one to detect the sentiment of a term within a message (term-level task), obtaining an F-score of 69.02 in the message-level task and 88.93 in the term-level task. They also generated two large word–sentiment association lexicons, one from tweets with sentiment-word hashtags, and one from tweets with emoticons.

[2] The corpus contains 5,553 tweets and is developed using small-scale content analysis, They classified twitter tweets into 28 emotion categories. Out of 28 emotional categories 33% of the tweets containing emotion are positive, 13% are negative and only 3% are neutral.. They used Machine Learning models to predict the classifier and metric for result used was Micro-averaged F1 for multi-class-single (MCS) and multi-class-binary (MCB).Existing classifiers achieve only moderate performance in detecting emotions in tweets even those trained with a significant amount of data. BayesNet classifiers produce consistently good performance for fine-grained emotion classification.

[3], The paper mentions about the use of NLP in Social Media posts like twitter. Further it very randomly define as to how applying NLP models is very different for Traditional posts/articles/blog compared to Social Media posts/tweets etc and the challenges that it create. Moreover, it explains in about how to collect twitter dataset and modify it as per the challenges of the target model like handling emoji's and lexical lexical relationships.

[4] Research article focuses on classifying Amazon product reviews into three classes - positive, negative and neutral. It uses data collected from Amazon product reviews and adopts a machine tagging approach, implemented via bag of words model.Prior to machine tagging, several pre-processing steps are namely tokenisation, lemmatization and part of speech tagging (POS-tagging) were implemented. This bag of words model counts the occurrences of positive and negative tokens in a sentence and assigns it a ground truth label based on that. Feature vectors were developed based on total tokens and several models such as SVM, Naive Bayesian Model and Random Forests were used out of which the Random Forest classifier which was essentially an ensemble method using bagging, outperformed the rest.

[5] The team extracted tweets with the help of the Twitter Streaming API around 520,000 tweets as raw data performing text preprocessing and feature augmentation to add additional attributes that are effective for emotion identification. The decision tree, decision forest and decision majority rule are used to classify the tweets into the eight classes. The proposed classifiers are implemented on both WEKA and Apache Spark system over Hadoop cluster for scalability purpose.

4 METHODOLOGY

4.1 Dataset

The data is basically a collection of tweets annotated with the emotions behind them. We have three columns: id, emotions, and text. In "text" we have the raw tweet. In "emotions" we have the emotion behind the tweet. To suit our 5 class classification problem we manually re-annotated the tweets, primarily the sentiment column to fit our classes.

4.2 Data Preprocessing

As per the above foreseen challenges of a social media text which are comparatively a lot more informal and thus complex to process, we need to perform suitable and specific preprocessing that can allow our model to get more accurate results. For this purpose we need to utilize the common informal features of tweets.

• **URLs:** URLs present in the tweets mostly provide no significant value to identify the emotions of the author. So we remove them to get a cleaner data.

• **Mentions:** The mentions are used to get the attention of some other user which proves as not valuable for detecting the emotions, and thus can be removed.

• **Hashtags:** Hashtags are considered very important in a tweet. But they can be hard to process as they can very commonly happen to be a combination of words or abbreviations. So we need to carefully process the hashtag texts to get the meaningful data. For this we can divide the hashtags as follows:

• **Emoji's:** The emoji's are of key importance when it comes to emotion detection. For this purpose we classify the existing common emoji's that can help in emotion detection as per the categories that we need. And then based on that category we can give a probability score to the hashtag of the expected category.

• Next we further clean the data by checking for spelling errors and repetitions in words in the remaining text of the tweets and fetch the valid words from it.

4.3 Count-Vectorizer/BOW

In this embedding approach we build a vocabulary from the document/tweets which simply means we find unique words. We then create document (in our case tweets) vectors where each vector comprises binary 0/1's indicating presence or absence of each word in the document. Each vector is of the size of the vocabulary and this approach does not retain any semantic information about the document. It just indicates presence or absence of the word and not its context.

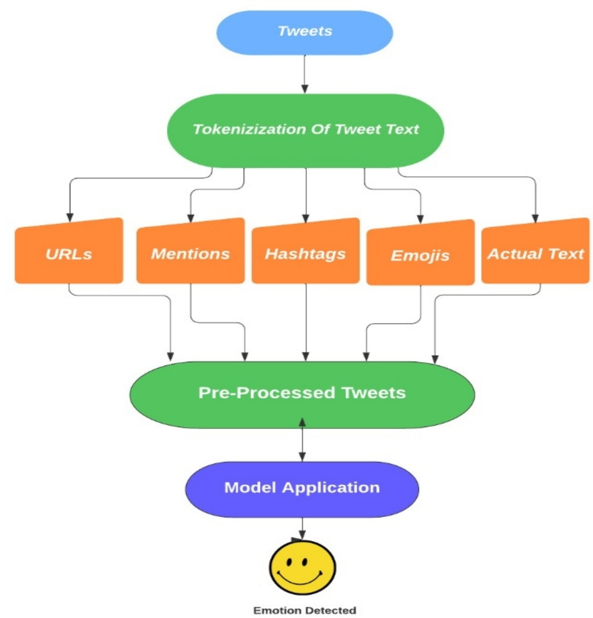

Figure 2: Workflow

4.4 TF-IDF

F stands for term frequency and IDF stands for inverse document frequency. TF is the normalized count of the number of times a word/term appears in a document. IDF involves measurement of how many documents a term appears in. The TF and IDF values are multiplied by and we obtain a score for each term per document. The TF-IDF approach, much like the Count-Vectorizer/BOW approach, does not retain any semantics of the words. The most notable feature about this approach is that it favors rare terms. The rarer the term, the higher its IDF value.

4.5 Word Vectorization

In this specific model, each word is basically represented as a vector of 32 or more dimensions instead of a single number. Here the semantic info and relation between similar words is also retained. The vectors are not hand crafted but are learnt. We use a fake problem such as fill in the blanks, and learn the vectors for each of the words. It is obviously more sophisticated than just that, but the basic intuition remains the same. We take a fake problem, use a machine learning model and help solve it and in the process, obtain vectors for the words. So for each sentence, we will have several M dimensional vectors, each corresponding to a word in the sentence. To have an overall vector for the sentence, we can take the average of all the M dimensional word vectors in the sentence.

5 EVALUATION

We implemented the following Machine Learning algorithms for the purpose of classification of the previously obtained twitter tweet vectors. We ended up classifying each tweet into one of the 6 categories namely - Sadness, Joy, Anger, Fear, Love and Surprise. The models have been trained on three different classifiers - Decision Trees, Random forest and Logistic Regression, and CNN.

5.1 Logistic Regression

The simple yet effective Logistic regression model ended up with an accuracy of 65.3 percent on the test set of tweets, when each tweet was converted into vectors using the Word2Vec model of Gensim library. The same model furnished an accuracy of 89.56 on the test set when the tweets were converted into tf-idf vectors. This stark change in accuracy when the method of converting tweets into vectors changed, could perhaps be attributed to the following reason. When we average the word vectors obtained from word2vec, to represent the tweet sentences, the individual essence of the word vectors get muddled up.

5.2 Decision Tree

The Decision Tree Classifier from Sklearn too had a monumental increase in terms of accuracy when the mode of embedding of tweets was changed from word2vec vectors to tf-idf vectors. We suspect it may be the same reason as mentioned above, at play here. The original accuracy of Decision Trees was 46.43 on the test set and it got increased to 83.981.

5.3 CNN

The Convolutional Neural Network approach was also adopted to solve the problem at hand. We used an embedding layer to convert the words of individual sentences into vectors. The process involved mapping every word of a sentence to an integer and then padding these vectors. This encoding was passed to the embedding layer that generated the vectors out of each integer. Then we added the following layers - Convolution, Max Pooling, Flatten, and two dense layers with the final layer containing the softmax activation. The model was trained for 4 epochs and it reached a training accuracy of 93 percent. Unfortunately it turned out that the testing accuracy was barely 26 percent. It could probably be due to the overall layers in the model. We lacked computational powers to further test other layers and models.

Model	Accuracy
Logistic Regression	65.37178976707148
Random Forest Classifier	61.602876766872384
Decision Tree Classifier	46.430171212422856

Figure 3: Classification Models in Baseline Models

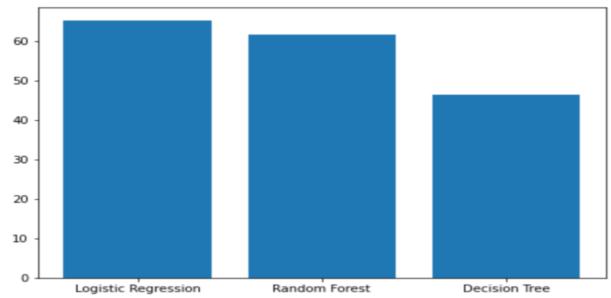

Figure 4: Accuracy Score comparison in Baseline Models

6 CONCLUSION

6.1 Result Analysis and comparison with baseline models

Generation of Tf-IDF vectors using L2 normalization and sublinear TF-scaling using TF-IDF transformer class gave the best accuracies. We have used Accuracy as the evalution metric for comparison with Baseline Models
Accuracies Were as Follows:

Model	Accuracy
Decision Tree	83.9831
Logistic Regression	89.5636

Figure 5: Final Models Comparison

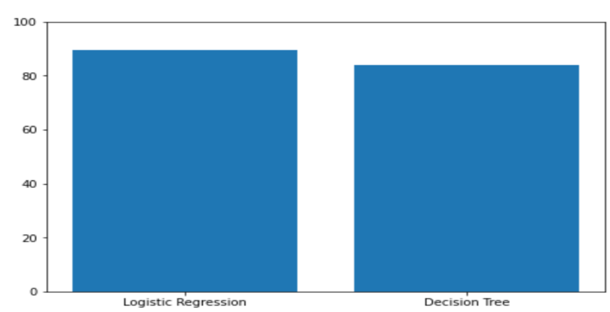

Figure 6: Final Models

6.2 System performance on new data and Handling Different cases

For the final implementation, we have proposed the following two methods wherein we can either choose to manually enter a particular tweet and predict its sentiment, or enter a twitter handle and perform web scraping and thereby predict the sentiment of the latest top ten tweets of that person. The final model i.e. Logistic Regression Model which was integrated into the Web UI using Flask although performs fairly well on the fetched tweets, falls prey to ambiguity of sentiment in certain cases. For example, Love and Joy and Surprise are often very closely present in certain tweets and prediction of sentiment of such tweets may sometimes lead to ambiguous results. Here Love and Joy is only predicted as Love in quite a few tweets. This is not inherently a problem with the model but with the nature of the tweet by virtue of which it contains equal amounts of more than one sentiment. Another problem we came across was ambiguous tweets. Several tweets are political, informational and do not possess much sentiment behind them. In such tweets it is difficult to classify them into our classes.

7 REFERENCES

[1] Saif M. Mohammad, Svetlana Kiritchenko, Xiaodan Zhu 2013. NRC-Canada: Building the State-of-the-Art in Sentiment Analysis of Tweets .

[2] Jasy Liew Suet Yan, Howard R. Turtle 2016. Exploring Fine-Grained Emotion Detection in Tweets.

[3] Maria Krommyda, Anastatios Rigos, Kostas Bouklas, Angelos Amditis 2020. Emotion detection in Twitter posts: a rule-based algorithm for annotated data acquisition.

[4] Xing Fang, Justin Zhan 2015. Sentiment analysis using product review data.

[5] Jaishree Ranganathan, Nikhil Hedge, Allen S. Irudayaraj, Angelina A. Tzacheva 2018. Automatic Detection of Emotions in Twitter Data - A Scalable Decision Tree Classification Method