Twitter based model for emotional state classification

Ravinder Ahuja
Dept. of Computer Science
Jaypee Institute of Information Technology
Noida, India
ravinder.ahuja@jiit.ac.in

Rohan Gupta
Dept. of Computer Science
Jaypee Institute of Information Technology
Noida, India
rohan-gupta@outlook.com

Saurabh Sharma
Dept. of Computer Science
Jaypee Institute of Information Technology
Noida, India
saurabhsharma576@gmail.com

Ayush Govil
Dept. of Computer Science
Jaypee Institute of Information Technology
Noida, India
ayush.govil@gmail.com

Karthik Venkataraman
Dept. of Computer Science
Jaypee Institute of Information Technology
Noida, India
karthik.vraman96@gmail.com

***Abstract*—With the advent and the subsequent rise of social network, there has been a surge of users expressing their emotions and daily feelings leveraging the social media platform. Each unit time, such data that is generated in monumental sizes, can be utilized to accurately detect one's emotional state. Twitter tweets is seen as a great source of information that can be exploited to build highly accurate and relevant emotion classifiers [1].**

Through this paper, we aim to propose a model to classify an individual's recent emotional state into eight predefined states. We also subsequently compare the results and accuracy of SVM, KNN, Decision Tree & Naive Bayes algorithm to implement and justify our prescribed approach.

***Keywords*—Social Network, Twitter, Tweet, Support Vector Machines, K Nearest Neighbors, Decision Trees, Prediction Model, Emotional Analysis, Emotional State.**

I. INTRODUCTION

In present times where social media reigns as the supreme choice for an individual to let there emotions out; twitter is a rich ensemble of emotions, sentiments and moods[2]. Each short message or in terms of Twitter's definitions - tweets, holds certain individual words that are key to that message's emotional identity. For example - " I spent a fabulous evening with my family at the Taj hotel! ", has the word 'fabulous'. This is an indicative of the happy state the individual is in.

TABLE I. EXAMPLES OF TWEETS WITH THEIR EMOTIONS

Tweet	Corresponding Emotion
Current social conditions of our country, sadden me.	Sad
This is a serious misuse of power!	Anger
Waiting for my dreams to come true.	Anticipation
What a pleasant surprise it was to him, that day.	Joy

After a careful selection, we have put down 8 classes of emotion, that would be effective to classify a broader range of tweets. These are: 'Anger', 'Anticipation', 'Disgust', 'Fear', 'Joy', 'Sad', 'Surprise', 'Trust'.

II. RELATED WORK

There has been a large amount of research in sentiment

analysis as compared to emotion analysis. Below described are some research works related to ours.

Zhao and Gui [5] have discussed and compared various methods to preprocess texts. They have discussed how removing numbers and stop words has little to no influence on text classification results as opposed to expanding acronyms and replacing negations. It also has established through multiple experiments, how Naive Bayes and Random Forest Classifiers are more sensitive than Logical Regression and Support Vector Machine when sophisticated pre-processing techniques are used.

In their paper [6] Mondher and Tomoaoki, talk about quantising the emotion percentage in a given tweet. This implies, detecting each sentiment within a tweet and calculating the weight or the influence it particularly has over the overall sentiment. Their approach involves initial classification of positive, negative and neutral. Then the ones which aren't neutral are sent ahead into the model for actual emotion classification, which then are further measured for correctness of sentiment detection.

In another paper [7], the researchers predicted the results of general state elections in India using Twitter tweets. Much like us, they used both supervised and unsupervised approaches in order to compare the approaches. The results of the analysis for Naive Bayes was the BJP (Bhartiya Janta Party), for SVM it was the BJP (Bhartiya Janta Party) and for the Dictionary Approach it was the Indian National Congress. SVM predicted a 78.4% chance that the BJP would win more elections in the general election. The actual results of the elections were clear indicative of BJP's majority.

III. PROPOSED SOLUTION

The model follows a two tier process during the analysis of a tweet string. The first tier, is involved with preprocessing the tweet in order to remove the words that would otherwise be a

hinderance in detecting words key to knowing the emotion state. The algorithms have been compared using the accuracy metric. Each algorithm's accuracy is calculated by comparing the results of the model with the labels of the test data set.

This is achieved by running over the tweet to remove useless words/special-terms/symbols. Firstly the HTML tags and URLs are removed, then the apostrophes are converted to back to original form. For example didn't becomes did not, I've becomes I have and so on. This is followed by splitting attached words, which mainly happens due to typographical mistakes or also known as typos. Now, we search for slang words and normalise them back to their original form, so things like 'luv' change to 'love'. Finally, we standardise words like 'happyyyyy' back to 'happy'. After the first phase of the processing, we would then be left with a 'cleaned' version of the tweet that can now be very easily traversed over in order to obtain the emotion.

The second phase involves the vectorising, training using train corpus and then finally testing using sample tweets in order to gauge the accuracy. The accuracy of each compared algorithm will further aid in justifying the best choice of algorithm for the model.

Vectorising the tweet is a means to quantise it, changing its qualitative value to quantitative. Since there are 8 emotions we maintain an 8 dimensional vector; each coordinate addresses the count of key words in tweets, that belong in any one of the emotion class. For every emotion class, a dictionary of words associated with that particular emotion is maintained using and online source. The 8 dictionary data set was obtained from National Research Council, Canada.

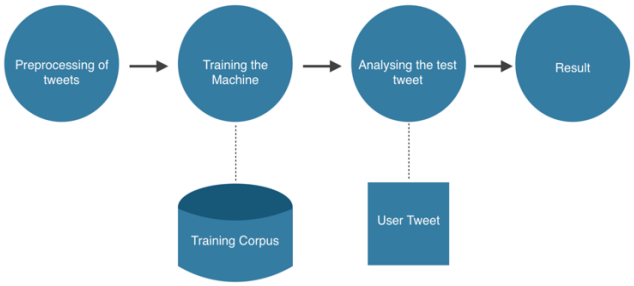


Fig. 1. Graphical representation of the analysis model

Now that each and every tweet in the train and test data corpus is vectorised, we can move ahead to perform the classification using the SVM, KNN, Decision Tree & Naive Bayes algorithms. First up, let's look at the SVM algorithm for classification and how our model utilises it.

A. Naive Bayes

We used Naive Bayes for its simplicity and great results despite a very obvious lack of sophistication as compared to the other ones in the model. In simple terms it is assuming that the presence of a particular feature in a class is unrelated to the presence of any other feature.

The Naive Bayes algorithm assumes that all the features are independent of each other (hence the name Naive Bayes). It is represented by a document as a bag of words. This is a disturbingly simple representation: it only knows which words are included in the document and how many times each word occurs, and throws away the word order. Naïve Bayes classification is nothing more than keeping track of which feature gives evidence to which class.

B. Support Vector Machines

SVM model [3] represents the vectors or points in space, mapped so that the vectors are grouped as a cluster and the division gap in between two clusters is widest possible. Then, classification is done by mapping the test vector; the vector is predicted to belong to that particular class based on which side of the gap it is. SVMs are also capable of performing non-linear classifications but for our model it suffices by using the linear classification.

Since implementing SVM from scratch can be a time taking business, one can leverage the open sourced libraries. Our model was built using the LibSVM implementation of the algorithm. Here, SVM is producing multi hyperplane space to support the eight dimensional vector. In order to train the machine we are using a fairly large data set of about 30000 tweets, each signifying one of the eight emotions. Following is a pictorial representation of what the space might look like.

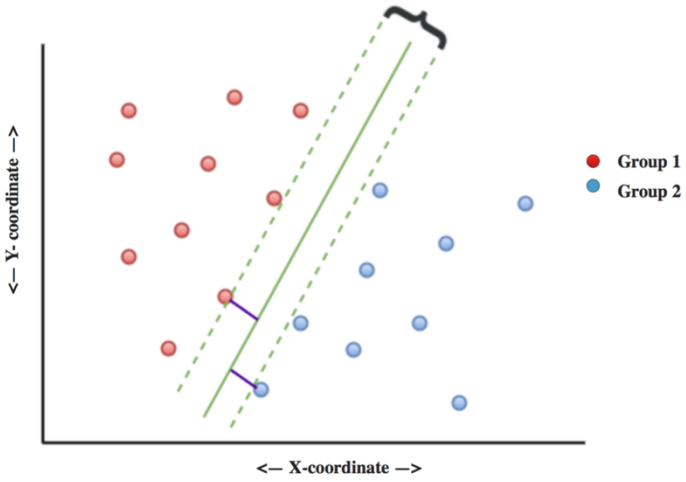


Fig. 2. SVM plane depicting the divison gap in between two groups

C. K-Nearest Neighbours

KNN[4] can be used for both classification and regression. KNN favours time and all parameters of consideration. For a new instance, predictions are made by searching through the training dataset for the k nearest instances. Nearness is calculated through Euclidean distance. Euclidean distance is calculated to

determine the K neighbours which are most similar to the new instance.

EuclideanDistance(x, xi):

$$y = \sqrt{\sum_{1}^{n}(x_j - x_{ij})^2}$$

Selecting the value of k is the most critical section as it diversify the output emotion. The best approach to select k is using the formula k = n where number is the total points.

There are many distance measures other than Euclidean distance which can be used with KNN such as Hamming Distance (distance between binary vectors), Manhattan Distance (distance between vectors using the sum of their absolute difference), Minkowski Distance (generalised form of Euclidean and Manhattan distance). The best distance metric can be chosen by the data properties.

- For regression, prediction is based on mean or median of k-similar instances.
- For classification, output is based on the mode of k- most similar instances.

Below is the pictorial representation of what the space would look like.

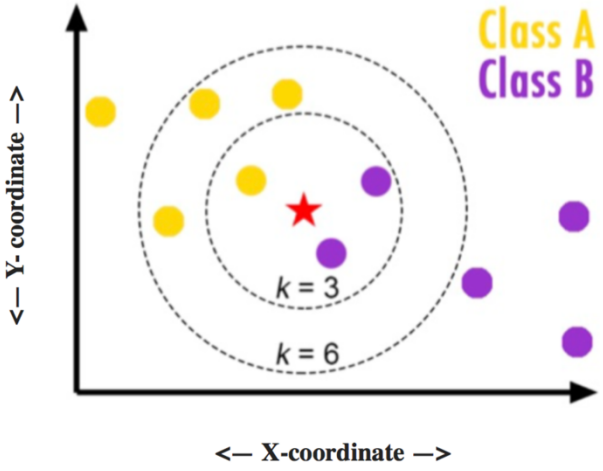


Fig. 3. KNN plane depicting points to be classified

In our model, we made sure to prepare data before using KNN. The algorithm succeeds more if all the data is scaled and normalised. Any missing data should be excluded from the dataset so that all the distances can be calculated. KNN works well with lower dimensional data (small number of input variables). Following is a table showing results by the algorithm using the test tweets.

TABLE II. EXAMPLES OF TWEETS WITH CALCULATED EMOTION WEIGHT

Emotions	"I feel Happy"	"I've been in foul mood these days"	"I hate the intermission in between any movie!"
Joy	64.2	0	0
Anticipation	25.5	0	8.5
Anger	0.2	46.5	56
Disgust	1	41	21
Sad	0.3	10.5	24
Surprise	1	0	0
Fear	0.8	2	10.5
Trust	2	0	0

D. Decision Tree

A decision tree [5] helps to predict what will be the output, given certain input variables. It represents all the variables involved in the decision and what are the consequences of each case. The decision tree is a greedy algorithm and it follows a top down approach. It partitions the data recursively at each step on the basis of some input conditions [6].

ID3:

ID3 is one of the Decision tree algorithms which generates a decision tree using a dataset. Initially there is a dataset and for each iteration and for every attribute, the entropy and information gain is calculated.

Entropy H(S):

It is defined as the randomness in a subset of the dataset and is used to create homogenous subsets.

$$H(S) = \sum p(x) log_2(1/p(x))$$

Where,

- S – The current (data) set for which entropy is being calculated (changes every iteration of the ID3 algorithm)
- X – Set of classes in S
- p(x) – The proportion of the number of elements in class to the number of elements in set

When H(S) = 0, the set is perfectly classified (i.e. all elements are of the same class).

Information gain:

Information gain/mutual information is a process of choosing an attribute having the least entropy and splitting the dataset on that attribute. The idea of the information gain is to reduce entropy.

$$IG(A,S) = H(S) - \sum p(t)H(t)$$

- H(S) – Entropy of set S.
- T – The subsets created from splitting set S by attribute A such that

$$S = \bigcup_{t \epsilon T} t$$

- p(t) – The proportion of the number of elements in t to the number of elements in set S.
- H(t) – Entropy of subset t.

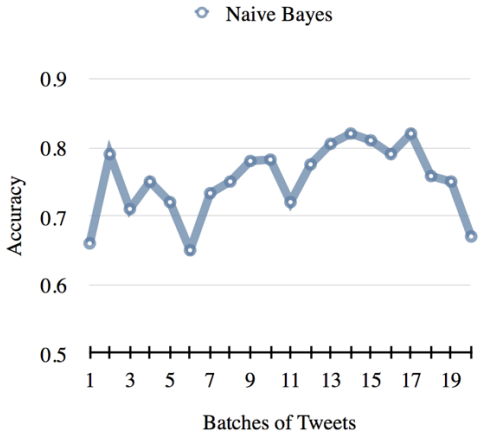
It then selects the attribute which has the smallest entropy (or largest information gain) value. The set S is then split by the selected attribute to produce subsets of the data. The algorithm continues to recurse on each subset, considering only attributes never selected before.

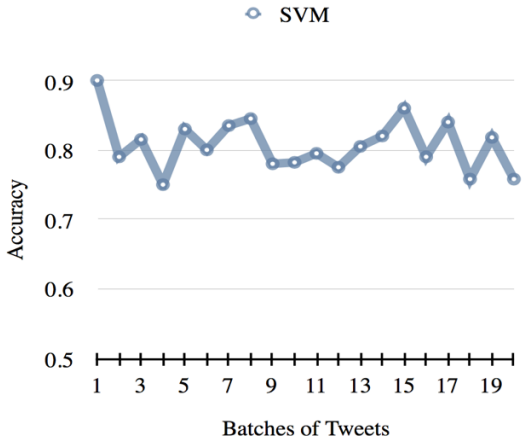
Recursion on a subset may stop in one of these cases:

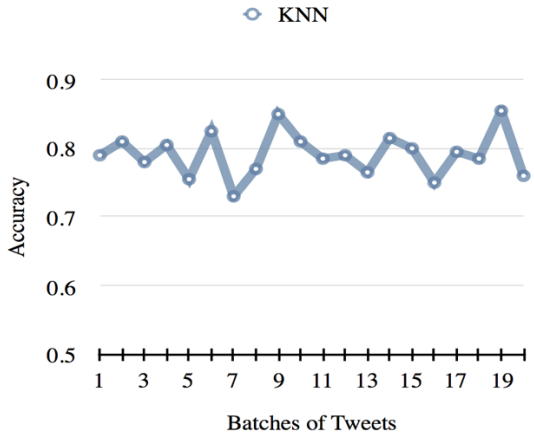
- Every element in the subset belongs to the same class (+ or -), then the node is turned into a leaf and labelled with the class of the examples
- There are no more attributes to be selected, but the examples still do not belong to the same class (some are positive and some happen to be negative), then the node is turned into a leaf and labelled with the most common class of the examples in the subset
- There are no examples in the subset, this happens when no example in the parent set was found to be matching a specific value of the selected attribute. Then a leaf is created, and labelled with the most common class of the examples in the parent set.

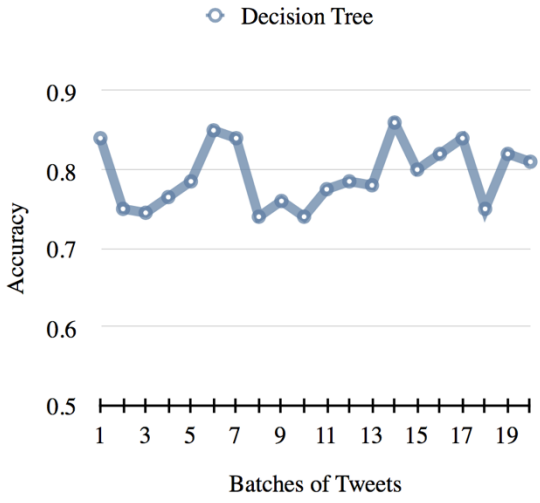
IV. RESULTS

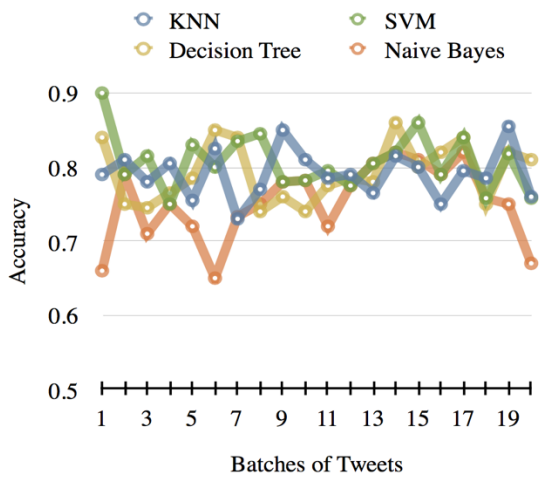
Following are the accuracy based comparison of the algorithms. We took 20 batches of 20 tweets of varied emotions in each case and calculated accuracy for each.











Peak value of each algorithm, is the highest value depicted in the graph.

Accuracy for each batch is calculated using pre-labeled tweets. Simply, by matching the model generated emotion with the label.

Average value of the each algorithm is simply the average of all the accuracy values obtain from analysing 20 batches of tweets (each containing 20 tweets).

So,

$$Average\ accuracy = \frac{\sum_{1}^{20}(Batch\ accuracy)}{20}$$

TABLE III. ACCURACY COMPARISONS OF THE FOUR ALGORITHMS

Accuracy	KNN	SVM	Decision Tree	Naïve Bayes
Peak Value	85.5	90	86	82
Average Value	79.125	80.73	79.275	75.125

It can be clearly seen that SVM performs the best regular accuracy hits in the late 80s region and peaking at at 90. This is followed by Decision Tree and KNN closely, with Naive Bayes coming at last.

V. CONCLUSION

We have proposed the discussed model as a method to classify Twitter tweets into 8 different broad emotional categories. The proposal has been designed as a two tier based model. The first tier is meant for preprocessing the tweet; which involves sanitising it from various words and terms, that include URLs, apostrophes, slang worlds and 'repetitive letter' words such as 'happppyyyyy'. After the tweet has been processed and cleaned appropriately, it is then vectorised by the identifying words that belong to one of the eight emotional categories. Very obviously, the vector has to be of eight dimension in order to carry the count of such words.

We create such vector of the test and training data, both. We have also, laid down comparison of the four algorithms that were used to achieve text classification of a given tweet. As discussed, previously our result indicates SVM as the best means to classify and achieve maximum accuracy. This model can very well fit in areas where predictions of ones emotion can prove to be vital, such as suggestions systems for individuals, or healthcare professionals.

In the future, we aim to improve the pre-processing phase of the model further. We also, intend to create a very practical application that would greatly underline the use of this model. Aside from justifying the model, the application would actually put the results of the model to a good use for the betterment of the society.

VI. REFERENCES

[1] Mark E. Larsen, Tjeerd W. Boonstra, Philip J. Batterham, Bridianne O'Dea, Cecile Paris, Helen Christensen, "We Feel: Mapping Emotion on Twitter" in IEEE Journal of Biomedical and Health Informatics, 2015

[2] Shulong Tan, Yang Li, Huan Sun, Ziyu Guan, Xifeng Yan, Jiajun Bu, Chun Chen, Xiaofei He, "Interpreting the Public Sentiment Variations on Twitter" in IEEE Transactions on Knowledge and Data Engineering, 2014

[3] Nipuna Upeka Pannala, Chamira Priyamanthi Nawarathna, J. T. K. Jayakody, Lakmal Rupasinghe, Kesavan Krishnadeva, "Supervised Learning Based Approach to Aspect Based Sentiment Analysis" in Computer and Information Technology (CIT), IEEE International Conference, 2016.

[4] Jun-li Lu et al., "Research and application on KNN method based on cluster before classification" in International Conference on Machine Learning and Cybernetics, 2008.

[5] Liu Jian, Wang Yan-Qing, "Research on Application of Decision Tree in Classifying Data" in International Conference on Intelligent Computation Technology and Automation (ICICTA)", 2011

[6] C. Hartmann et al., "Application of information theory to the construction of efficient decision trees" in IEEE Transactions on Information Theory, 2003

[7] Zhao Jianqiang, Gui Xiaolin, "Comparison Research on Text Pre-processing Methods on Twitter Sentiment Analysis" in IEEE Access, 2017

[8] Mondher Bouazizi, Tomoaki Ohtsuki, "Sentiment Analysis in Twitter: From Classification to Quantification of Sentiments within Tweets" in IEEE Global Communications Conference (GLOBECOM), 2016

[9] Parul Sharma, Teng-Sheng Moh, "Prediction of Indian election using sentiment analysis on Hindi Twitter" in IEEE International Conference on Big Data (Big Data), 2016

[10] Anil Bandhakavi et al., "Lexicon Generation for Emotion Detection from Text" in IEEE Intelligent Systems, 2017

[11] Li Zhiping, Sun Yu, Xie Jili, "The research of user models with emotional analysis" in 10th International Conference on Computer Science & Education (ICCSE), 2015

[12] Sadia Zaman Mishu, S. M. Rafiuddin, "Performance analysis of supervised machine learning algorithms for text classification" in 19th International Conference on Computer and Information Technology (ICCIT), 2016

[13] I. Shahin, "Analysis and investigation of emotion identification in biased emotional talking environments" in IET Signal Processing, 2011

[14] Kyunglag Kwon et al., "Sentiment trend analysis in social web environments" in IEEE International Conference on Big Data and Smart Computing, 2017

[15] Kim Schouten et al., "Supervised and Unsupervised Aspect Category Detection for Sentiment Analysis With Co- Occurrence Data" in IEEE Transactions on Cybernetics, 2017