[ICRTCET-2018]

ISSN 2348 – 8034
Impact Factor- 5.070

GLOBAL JOURNAL OF ENGINEERING SCIENCE AND RESEARCHES

EMOTION DETECTION AND ANALYSIS ON SOCIAL MEDIA

Bharat Gaind*1, Varun Syal2 & Sneha Padgalwar3
*1,2&3CSE Department, IIT Roorkee, Roorkee, India

ABSTRACT

In this paper, we address the problem of detection, classification and quantification of emotions of text in any form. We consider English text collected from social media like Twitter, which can provide information having utility in a variety of ways, especially opinion mining. Social media like Twitter and Facebook is full of emotions, feelings and opinions of people all over the world. However, analyzing and classifying text on the basis of emotions is a big challenge and can be considered as an advanced form of Sentiment Analysis. This paper proposes a method to classify text into six different Emotion-Categories: Happiness, Sadness, Fear, Anger, Surprise and Disgust. In our model, we use two different approaches and combine them to effectively extract these emotions from text. The first approach is based on Natural Language Processing, and uses several textual features like emoticons, degree words and negations, Parts Of Speech and other grammatical analysis. The second approach is based on Machine Learning classification algorithms. We have also successfully devised a method to automate the creation of the training-set itself, so as to eliminate the need of manual annotation of large datasets. Moreover, we have managed to create a large bag of emotional words, along with their emotion-intensities. On testing, it is shown that our model provides significant accuracy in classifying tweets taken from Twitter.

Keywords: *Sentiment Analysis, Machine learning, Data mining, Natural Language Processing.*

I. INTRODUCTION

Emotions are described as intense feelings that are directed at something or someone in response to internal or external events having a particular significance for the individual. And the internet, today, has become a key medium through which people express their emotions, feelings and opinions. Every event, news or activity around the world, is shared, discussed, posted and commented on social media, by millions of people. Eg. *"The Syria chemical attacks break my heart!! :'("* or *"Delicious dinner at Copper Chimney! :D"* or "OMG! That is so scary!". Capturing these emotions in text, especially those posted or circulated on social media, can be a source of precious information, which can be used to study how different people react to different situations and events.

Business analysts can use this information to track feelings and opinions of people with respect to their products. The problem with most of the Sentiment Analysis that is done today is that the analysis only informs whether the public reaction is positive or negative but fails to describe the exact feelings of the customers and the intensity of their reaction. With our emotional analysis, they can have a more profound analysis of their markets than the naive 2-way Sentiment Anal-ysis, which itself has turned their businesses more profitable. Business leaders can analyse the holistic view of people in response to their actions or events and work accordingly. Also, health-analysts can study the mood swings of individuals or masses at different times of the day or in response to certain events. It can also be used to formulate the mental or emotional state of an individual, studying his/her activity over a period of time, and possibly detect depression risks.

There are plenty of research works that have focussed on Sentiment Analysis and provide a 2-way classification of text. But few have actually focussed on mining emotions from text. However, machine analysis of text to classify and score it on the basis of emotions poses the following challenges :

- Instead of the usual two categories in Sentiment Analysis, there are six *Emotion-Categories* in which we need to classify the tweets.
- Lack of manually annotated data to train classifiers to label data into six categories.
- Unavailability of a comprehensive bag of *Emotion-words* labeled and scored according to *Emotion-Categories* (Happiness, Sadness, etc.) and their intensities, that can be used to detect *Emotion-words* in text.

In order to address the aforementioned challenges, it was important to devise a system that could generate a good and reliable training-set for the classifier, a labeled bag of words, and an algorithm that could not only detect emotions, but also score and label the tweets according to those emotions.

A lot of research has been done on classifying comments, opinions, movie/product reviews, ratings, recommendations and other forms of online expression into positive or negative sentiments. Emotions have also been studied, but in a limited extent, such as by asking specific questions and judging on the basis of replies, or an analysis done only on short one-lined headlines or a few others [1], [2], [3], all of which depended on the manual annotation of the training dataset of a small size and limited scope.

In this paper, we propose a method to classify and quantify tweets according to six standard emotions suggested by Paul Ekman [4]. Here, we base our analysis on tweets posted on Twitter, but it can be easily extended to any kind of text whether it is one lined headlines, messages and posts on social media or larger chunks of writings, because of automatic development of our training set. Our main contributions are listed below:

- We have developed a system that could score and label any piece of text, especially tweets and posts on social media according to six *Emotion-Categories*: Happiness, Sadness, Fear, Surprise, Anger and Disgust along with their intensity scores, making use of its textual features, a variety of NLP tools and standard Machine Learning classifiers.
- Another significant contribution is that we have success-fully devised a system that could automatically (without any manual effort) build an efficient training set for our ML Classifiers, consisting of a large enough set of labeled tweets from all *Emotion-Categories*.We have created a large bag of words in English, that consists of words expressing a particular emotion along with the intensity of that emotion.
- We were able to achieve an accuracy of about 91.7% and 85.4% using J48 and SMO classifiers respectively using the training set we built.

II. RELATED WORK

In the recent past, with the rise of social media such as blogs and social networks, a lot of interest has been fueled in Sentiment Analysis. Lately, a lot of research has been done on classifying comments, opinions, movie/product reviews, rat-ings, recommendations and other forms of online expressions into positive or negative sentiments. Earlier research involved manually annotated corpus of limited size to classify the emotions. Wiebe et al. [5] worked on the manual annotation of emotions, opinions and sentiments in a sentence corpus (of size 10,000) of news articles. Segundo et al. [6] also studies the presence of emotions in text, and is a functional theory of the language used for expressing attitudes, judgments and emotions [3]. This paper deals explicitly with emotions, which none of the aforementioned works do. There was, however, one research work [7], that classified text into six *Emotion-Categories*, but that was only limited to classification of news headlines, and the training set used was created manually. We, on the other hand, have developed a system, which classifies text in any form (eg. news, tweets, or narrative) and uses a training set, which is generated automatically. This saved a lot of effort. Another similar work was of Carlo and Rada [8] who did a similar classification on news headlines but even they used manually annotated corpus for their classification. There is one research work [9], that explores the possibility of creating automatic training datasets, like we do and also tries to find out if creating large emotion datasets can increase the emotion-detection accuracy in tweets. Some aspects of their work are similar to ours, but the emotion-detection accuracy we have achieved is much higher. Another similar work is [10], but again, our accuracy is much higher. A recent work [11] explores the possibility of predicting future stock returns based on tweets related to presidential elections and NASDAQ-100 companies. Another recent work [12] uses convolutional neu-ral network architecture for emotion identification in Twitter messages. Their approach uses unsupervised learning, whereas we use supervised learning and their accuracy of 55.77% is much lower than ours.

III. DATA SETS

This section describes the various data sets used, such as *Tweets Set*, *Emotion-Words Set (EWS)*, *Degree-Words Set* and *Location-Areas Set*.

A. Tweets Set

We use Tweepy [13] to collect tweets, which is a Python library for accessing the Twitter API. It takes as input various parameters, such as coordinates, radius, etc., and after removal of duplicates, links, hashtags, and words in other languages (besides English) from these tweets, stores the tweet-ids, text and location of the most recent ones in the database. This provided us with a list of tweets from various locations across the country. Eg. The *Tweets Set*, we created for Delhi has about 10,000 entries. Another way we used Tweepy is by feeding it a twitter-username (of a user) as an input to store all the tweets of that user (till date), in our database.

TABLE I: An example of Emotion-Words Set (EWS)

Word/ Emoticon	Emotion-Category	Intensity-Category
:O	SURPRISE	STRONG
Repugnance	DISGUST	STRONG
Delighted	HAPPINESS	MEDIUM
Afraid	FEAR	STRONG
:(	SADNESS	STRONG
Irritated	ANGER	STRONG
Lucky	HAPPINESS	LIGHT

B. Emotion-Words Set (EWS)

A proper selection of relevant and commonly-used *Emotion-words* is one of the most essential and indispensable aspects in the emotional quantification of a sentence. We have created a high-quality accurate bag of words, called *Emotion-Words Set (EWS)*, of around 1500 words. It has been developed by recursively searching (Depth First Search) the synonyms of the 6 basic *Emotion-Categories (HAPPINESS, SADNESS, ANGER, SURPRISE, FEAR, DISGUST)* in a thesaurus [14], up to two levels. Each of these words was then manually labeled to one of the three *Intensity-Categories (STRONG, MEDIUM, LIGHT)*, as shown in Table I.

C. Degree-Words Set

Degree-Words Set is a set of about 50 degree words, that are used to strengthen or weaken the intensity of emotions in a sentence. Eg. *"too happy"* and *"hardly happy"* have two different meanings, almost opposite to each other. In this set, each word has an associated *Degree-Intensity* stored with it; *H*

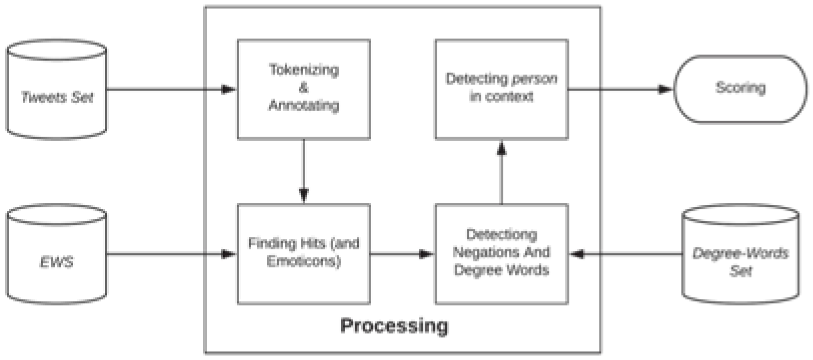

Tweets Set
EWS
Tokenizing & Annotating
Finding Hits (and Emoticons)
Detecting person in context
Detectiong Negations And Degree Words
Degree-Words Set
Scoring
Processing

Fig. 1: An overview of the first approach

meaning High (having a high multiplying effect), *N* meaning Negation (having an opposing effect), and *L* meaning Low (having a Low multiplying effect). Examples include words like, *"too"*, *"more"* (*H*); *"hardly"*(*N*); and *"nearly"* (*L*), etc.

D. Location-Areas Set
We also store the areas of about 20 major cities in India in *Location-Areas Set*, to calculate the radii that are to be used during tweet extraction. Examples are New Delhi (1484 *sq. km.*), Mumbai (603 *sq. km.*), etc.

IV. EMOTION-DETECTION ALGORITHM

Our model consists of two completely different, yet interde-pendent approaches. The first approach uses Natural Language Processing, *Emotion-Words Set* and several textual features. It attempts to classify and score text according to the emotions present in it. The second approach uses standard classifiers like SMO and J48 to classify tweets. Finally, we combine both these approaches to propose a Hybrid approach to detect emotions in text more effectively. Note that, even though we are extensively using the tweets example throughout this paper, this algorithm is very generic and can be used to detect and quantify emotions in any piece of text.

A. Using NLP and EWS: First Approach
The first approach uses the tweets in *Tweets Set*, which are already free from any unwanted characters, hyperlinks or hashtags. Various *Stanford CoreNLP* [15] tools are used for a comprehensive linguistic analysis. The following steps comprise the first approach.

Tokenizing and Annotating: First, we use *PTBTokenizer*
[16] to tokenize the tweets into sentences, which are further tokenized into tokens. Then, we remove all the stop words from these tokens. The filtered tokens are then annotated using the following *CoreNLP annotators*:

- *pos*: Parts of Speech (noun, verb, adjective, for example)
- *lemma*: Lemmatized version of that word.
- *ner*: For Named Entitiy Recognition.

We also fetch the Grammatical dependencies between the tokens in a sentence which include *nsubj* (the subject of a sentence), *dObj* (the object of a sentence), *advMod* (the relation between a word and an adverb), and *neg* (the relation between a word and a negative word).

Finding Hits: In this step, the annotated (and filtered) tokens are matched against the words present in the *EWS*. But first, all the words in the *EWS* are lemmatized to their base form. Eg. *"happiness"* and *"happily"* are changed to *"happy"*. While matching the tokens against the *EWS*, only the tokens that are annotated as *"O"* (the *other* entity in Named Entity Recognition) are considered, because a named entity (location, time or a person word, for example) can never be an *Emotion-word*. A matched token along with all its characteristics/annotations is stored as a *hit*.

Detecting person in context: In our analysis, we aim at detecting the emotions expressed by the tweet's poster, stress-ing on the poster's feelings. Therefore, we try to differentiate between cases in which the emotions involved are in relation to the poster himself or someone else. For instance, the degree of sadness of the poster in the tweet *"I am sad"* is much higher compared to the tweet, *"He is sad"*, posted by the same person, as the former clearly and directly suggests the sadness of the poster himself. This is done by using the *nsubj* annotation of the tokens, which tells us who the subject of the sentence is. After finding the subject of the sentence, we determine which *person* (*first*, *second* or *third*), that subject belongs to. For this, we maintain a list of *first* (*"I"*, *"me"*, etc.), *second* (*"You"*, *"your"*, etc.) and *third person* (*"He"*, *"She"*, *"They"*, etc.) pronouns. Also, if the *nsubj* is a proper noun, then the *person* of that *nsubj* is considered to be *third*. Next, each *hit* detected in the previous step is associated with a *person*, by finding the nearest *nsubj* to the *hit* on its left side. If there is no *nsubj* in a sentence, we associate the *hit* with the *first person*. For instance, the *hit "Nice"* in the tweet, *"Nice to see you!!"* is associated to the *first person*, as there is no *nsubj* in the sentence.

Effect of Negation and Degree-words: There are many instances, when some words detected using the *EWS* are ac-tually used in an entirely opposite sense, because of negations like *"not"*, *"never"*, *"don't"*, etc. For instance, the tweet *"Not at all feeling excited for school!"* may result in getting *Happiness* as its *Emotion-Category* (because of the *Emotion-word "excited"*), if negations are not accounted for. Also, there are many words (mostly adverbs), that enhance the intensity of an emotion. Eg. *"I feel good"* and *"I feel too good"* contain the same *Emotion-word "good"* but the score given to the second sentence should be more. For detecting negations and degree-words, we search for tokens that have been annotated as *neg* and *advmod*, while determining the grammatical de-pendencies in step 1, and tag them as degree/negation words, if they lie in the *Degree-Words Set*. The *Degree-Intensity* of the degree-word associated (if any) with every hit is stored as an annotation to the hit.

Emoticon Detection: Emoticons are a very useful and informative source for detecting emotions in text. Since emoti-cons are direct ways of knowing the emotions of the user, they are given a heavy weightage, if found. This feature is very effi-cient and accurate, especially with the ever-rising popularity of emoticons. We have added a list of 100+ most commonly used emoticons in the *EWS* to enhance our model, along with their *Intensity-Categories* (*STRONG* and *MEDIUM*). The emoticon-detection process is done before tokenization and uses regex. The emoticons detected in a tweet are also treated as *hits*.

TABLE II: Emotion-Scores of various feature combinations

Intensity-Category	*Degree-Intensity*	*emotScore*
STRONG	*None*	6
STRONG	*H*	8
STRONG	*L*	6
STRONG	*N*	2*
MEDIUM	*None*	4
MEDIUM	*H*	6

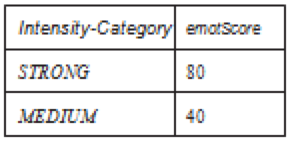
ISSN 2348 – 8034
Impact Factor- 5.070

MEDIUM	*L*	6
MEDIUM	*N*	4*
LIGHT	*None*	2
LIGHT	*H*	6
LIGHT	L	4
LIGHT	*N*	4*

6) *Scoring:* Every *hit* is finally, a tuple of its features that contribute in the scoring process:

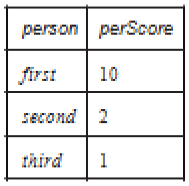
- *lemma*: the lemmatized version of the hit word
- *Emotion-Category*: *HAPPINESS, SADNESS, ANGER, SURPRISE, FEAR, DISGUST*

TABLE III: Emoticon Scores



Intensity-Category	emotScore
STRONG	80
MEDIUM	40

TABLE IV: Person Scores



person	perScore
first	10
second	2
third	1

- *Intensity-Category*: *STRONG, MEDIUM or LIGHT*
- *Degree-Intensity*: *H, L, N* or *None*
- *person*: *first, second or third*

Based on the various possible combinations of these fea-tures, *emotScore* and *perScore* is calculated for each of the *hits*. If the *hit* is an English word (and not an emoticon), the values in Table II are used for calculating *emotScore*, whereas for emoticons, Table III is used. It should be noted that if the *Degree-Intensity* of a hit is *N* (negation ef-fect), its *Emotion-Category* changes. This means that *HAPPI-NESS* turns into *SADNESS*/*ANGER* and vice-versa (with their *emotScores* given in Table II, marked by asterisks), whereas the *emotScores* of the remaining three *Emotion-Categories* (*DISGUST*, *SURPRISE*, *FEAR*) becomes zero, when the *Degree-Intensity* is *N*. Table IV is used for calculating the *perScore* of a *hit*. The final emotional score (*Score*) of a tweet is a six-tuple, which is calculated by summing over all the *hits* (of a particular *Emotion-Category*) in all the sentences in the tweet for each of the six *Emotion-Categories* (*EC*$_j$, *j=1,2,...6).*

$$Score[EC_j] = \sum_{All\ Hits\ of\ EC_j} (emotScore * perScore) \quad (1)$$

$$RelScore[EC_j] = \frac{Score[EC_j]}{\quad} * 100 \quad (2)$$

6
Score[*EC*$_j$]
j=1

Also, the relative score (*RelScore*) is calculated to under-stand the percentage of each emotion in a tweet. If there are no *Emotion-words* in a tweet, the value of scores of all *Emotion-Categories* in *Score* will be zero. For example, if the tweet has no words of the *Emotion-Category FEAR*, then *Score[FEAR]* is 0. An overview of the first approach is given in Fig 1.

B. Using Machine Learning: Second Approach
We use another method for emotion-detection where we use a Machine Learning classifier. For this, the most important and critical step is to prepare a good training-set. We make use of the first approach to generate the training-set, which is free from any manual annotation and thus can be developed or updated quickly.

1) Generation of the training-set: We have chosen a set of seed words, comprising of commonly used *Emotion-words* and emoticons from the *EWS*, evenly distributed over all the Emotion-Categories. We then query Tweepy using these seed words to develop a huge database of around 13,000 tweets. The seed words are used to ensure that we get tweets that express at least one of the six emotions. The retweets are removed to avoid any repetition of tweets. The even distribution of seed words ensures that we get an even distribution of tweets over all the Emotion-Categories, so that our classifier isn't biased. However, we have analysed that there is a large proportion of tweets on Twitter expressing happiness, so we have kept the number of seed words possessing the *Emotion-Category HAPPINESS*, on the higher side.

2) Filtering and Labelling the training-set: For training an accurate classifier, we need all the tweets in the training-set to be strongly expressing only one of the six *Emotion-Categories*. However, there are many tweets comprising of more than one emotion. Such type of tweets reduces the accuracy of the classifier, as we can give only one label to a tweet in the training set. If a tweet contains more than one emotion in the training-set, it will result in faulty learning as words related to the other (unlabeled) *Emotion-Category* will also be treated as those related to the category labeled. Thus, we use the first approach to label all the 13,000 tweets and filter out the tweets having none or mixed emotions. Only those tweets which have a percentage of more than 70% for a particular emotion are labeled and fed to the classifier for training. The final distribution of tweets in our labeled training set is as shown in Table V.

TABLE V: Distribution of tweets in the training-set

Emoticon-Category	No. of tweets
HAPPINESS	2617
SADNESS	1416
FEAR	1459
DISGUST	776
ANGER	1316
SURPRISE	944
Total	8528

3) Training the classifiers: We use the open source library Weka [17] for the implementation of the classifiers. We use two very popular Weka classifiers, SMO [18] and J48 [19]. Before classification, we applied the following pre-processing steps on the data:

1. Stop-Word Filtering
2. Lower-casing all words
3. Stemming each word using Weka's Snowball Stemmer

The classifiers output the relative probabilities of each of the six *Emotion-Categories*, for a tweet (or a piece of text). The *Emotion-Category* with the highest probability is called the *Labeled-Category (L_c).*

C. Combining First and Second Approach Results

We finally combine the first and the second approach. The scores which are generated by the first approach are modified, according to the *Labeled-Category* of the classifier. The score of only the *Labeled-Category* is modified as follows:

$$FinalScore[L_c] = Score[L_c] + (0.2 * Score[M_c]) \qquad (3)$$

where M_c refers to the label with the maximum score, after the first approach results. This is done to give weightage to the results of the classifier by increasing the score of the classified category in proportion to the maximum so as to avoid too little or too much relative change in the scores. This will definitely make a difference in deciding the final *Emotion-Category*, when two different *Emotion-Categories* are too close by or equal and are in competition. After this, the *Emotion-Category* with the maximum final score is decided as the final *Emotion-Category* of the tweet (or the piece of text). An overview of the combined approach is shown in Fig 2.

V. RESULTS AND ANALYSIS

A. Testing Results

Similar to the training-set, we have created a testing-set of tweets by extracting tweets using some seed words and then using the first approach to filter and label these emotional tweets. This set consists of a total of 900 tweets, where each *Emotion-Category* has around 150 tweets, so as to maintain uniformity. Also, it is ensured that all tweets in the testing set are different from those in the training set. The two chosen classifiers, on testing with the testing-set, gave the results as shown in Table VI and VII. The correctly classified instances are the tweets for which the expected *Emotion-Category* matches the actual *Emotion-Category*. As can clearly be seen from the tables, we have achieved a remarkable accuracy of 91.7% for SMO and 85.4% for J48, which proves the merits of our Emotion-Detection Algorithm.

TABLE VI: Accuracy of the SMO Classifier

SMO		
Correctly Classified Instances	826	91.7 %
Incorrectly Classified Instances	74	8.22 %
Total No. of Instances	900	

B. Surety Factor (S_f)

The surety factor indicates how confident and assertive our analysis and results are, on a given text/tweet. Its value is low, when there is a mismatch between the results of the two approaches or when the text seems to lack any emotions. On

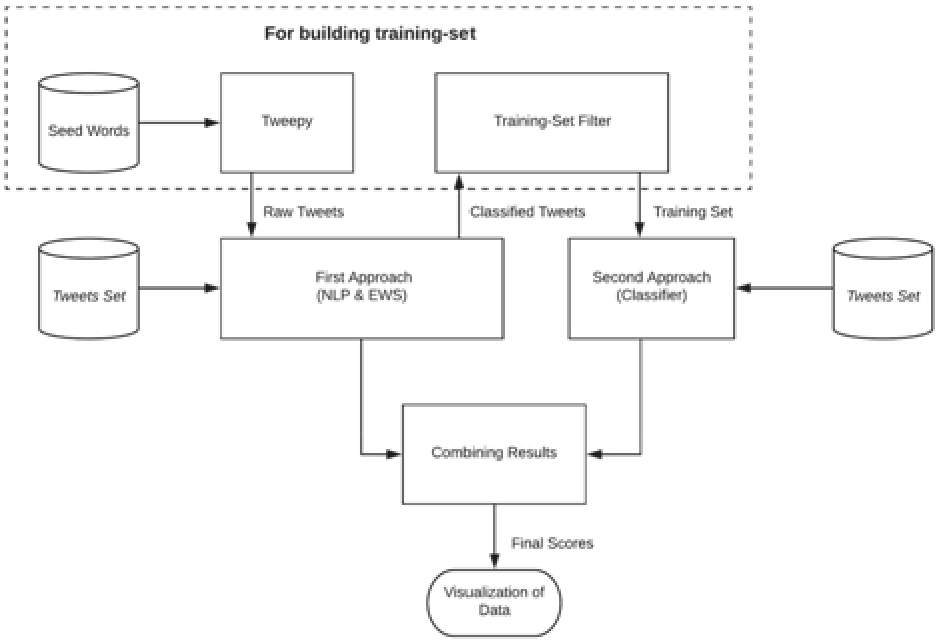


Fig. 2: Combining the first and the second approach

TABLE VII: Accuracy of the J48 Classifier

J48		
Correctly Classified Instances	769	85.4 %
Incorrectly Classified Instances	131	14.5 %
Total No. of Instances	900	

the other hand, its value is high when the two approaches concur with each other in results, there are too many *hits* or when one of the emotion-scores is very high. The surety factor is calculated on a scale of 6 and is dependent on several factors:

- Classifier label match: Whether the classifier label matches the category of the maximum score of the first approach. (Value: True/False).
- Max score: The value of the maximum of all the six scores.
- Max percent: The relative percentage of the Max score over all six scores.
- Second diff: The difference between the maximum and second maximum score value as a percent of the maxi-mum.
- Hits: The number of *Emotion-words* matched in the text.

- Surety factor is calculated differently for the following two cases:

- If the *hits* in the tweet/text belong to the same *Emotion- Category*, then only the factors Classifier label match and Max score are used.
- If the *hits* in the tweet/text belong to different *Emotion-Categories*, all five factors are considered.

C. Visualization of Results
In order to demonstrate the usefulness of the proposed Emotion-Detection Algorithm, we have implemented the fol-lowing applications:

- One-user Analysis (twitter account): We have developed an interface, which takes as input, the username of any twitter-user, processes all his/her tweets till date and displays a time-varying mood-swings plot as well as the relative and absolute emotional distribution in the form of pie charts. Fig. 3 is an example of the varying happiness of a twitter-user over time.

- Location Analysis: Using Google Maps API, a world map is displayed and each circle on it represents the emotional analysis of that particular region. The radius of each circle is proportional to the area of the region. This analysis is done for around 20 major cities in India, using the *Location-Areas Set* (See Fig. 4)
- Document Analysis: Another interface takes as input, entire documents or blocks of text, processes it, and displays the relative and absolute emotional distribution of the document in the form of pie charts. (See Fig. 5 for an example sentence.)

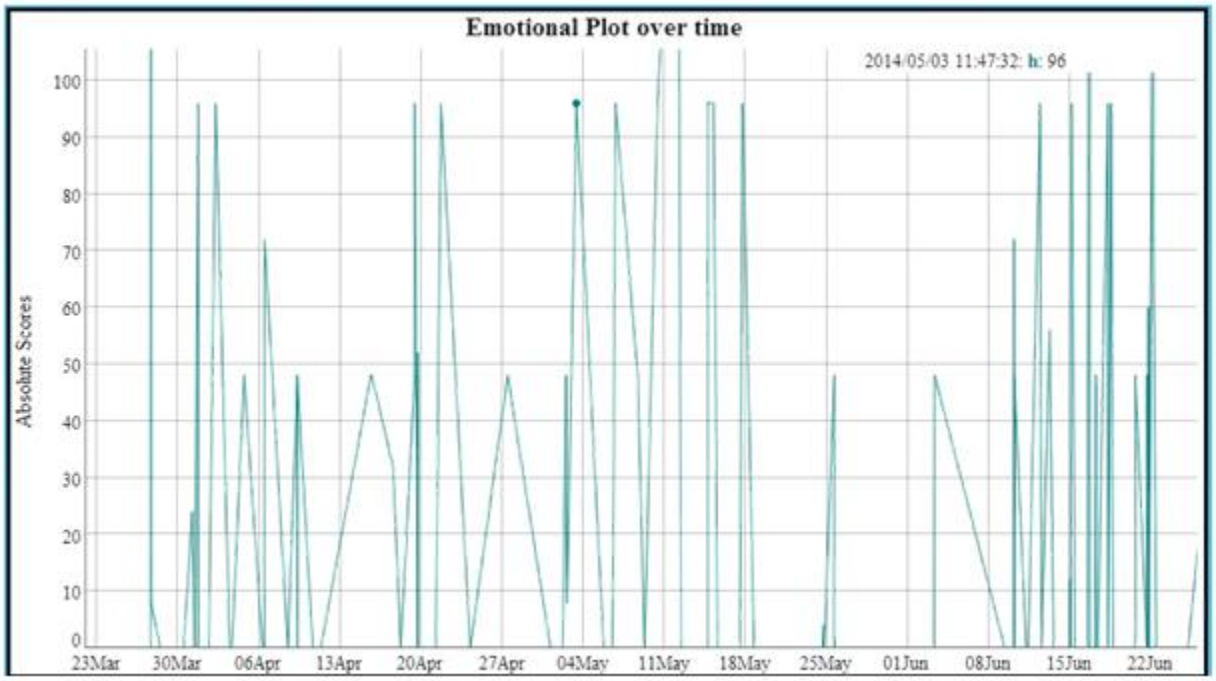


Fig. 3: Time varying happiness plot of a twitter-user

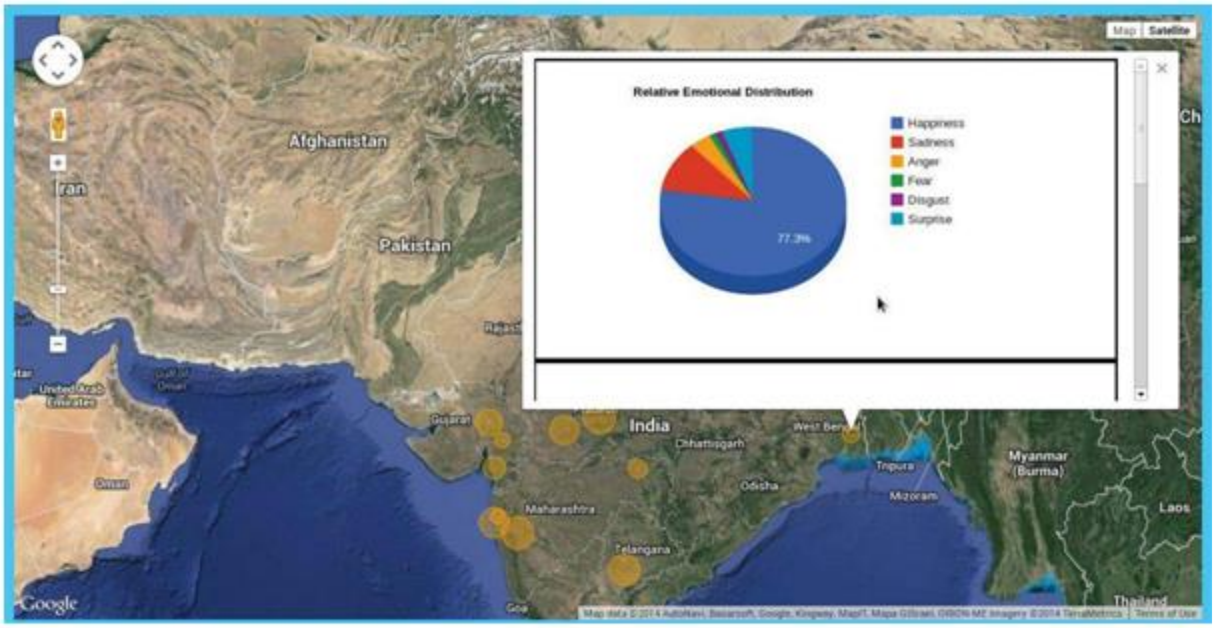


Fig. 4: Location-wise emotion-analysis of major cities in India

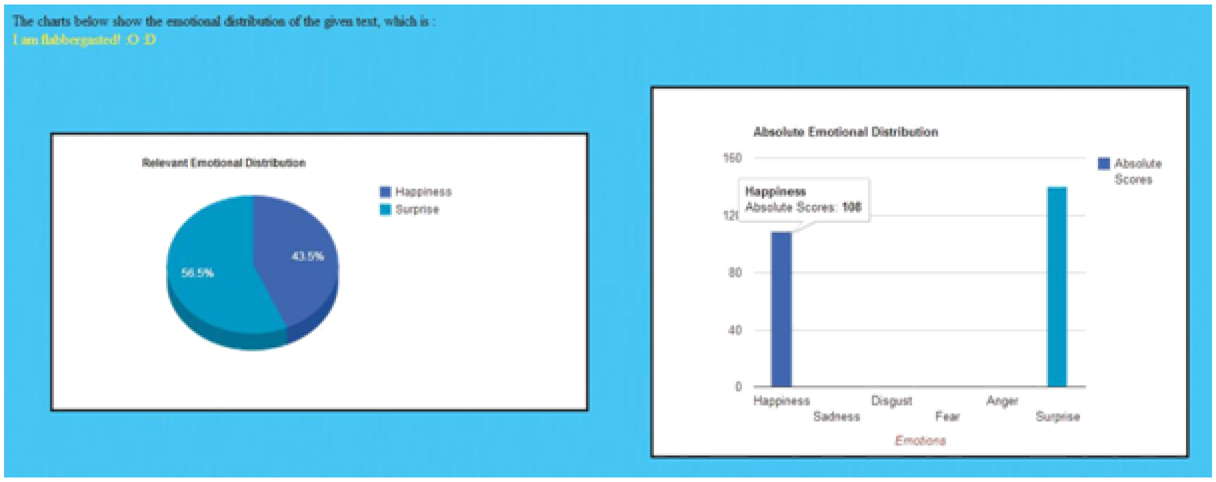

The charts below show the emotional distribution of the given text, which is :
I am flabbergasted! :O :D

Fig. 5: Detecting emotions in a piece of text

VI. CONCLUSION

In this paper, we have addressed the problem of classifying text into the six basic *Emotion-Categories*, rather than just la-beling them as positive or negative. Through our research and a self-generated reliable bag of emotional words (*EWS*), we can now effectively quantify various emotions in any block of text. We have also automatically generated a labeled training-set (without manually labeling the tweets) of emotionally-biased tweets using a keyword-matching approach, which was then used to train various classifiers. Moreover, we have also intro-duced the concept of Surety Factor to suggest the reliability of our output and the degree of usefulness and correctness of our results. Finally, we visualized our results using pie-charts, bar-graphs and maps, and demonstrated the various applications of our analysis. In future, a system could be established for automatically updating the bag-of-words which we created, on the basis of new tweets and data analysed. Using our approach, many interesting apps can be created, such as an add-on to a social-networking site displaying the recent mood of each of your friends. Also, our analysis of Twitter can be extended to the development of a real-time system, analyzing mood-swings and emotions on Twitter.

REFERENCES

1. *Rakesh C Balabantaray, Mudasir Mohammad, and Nibha Sharma. Multi-class twitter emotion classification: A new approach. International Journal of Applied Information Systems, 4(1):48–53, 2012.*
2. *Cecilia Ovesdotter Alm, Dan Roth, and Richard Sproat. Emotions from text: machine learning for text-based emotion prediction. In Proceedings of the conference on human language technology and empirical methods in natural language processing, pages 579–586. Association for Computational Linguistics, 2005.*
3. *Lisa Pearl and Mark Steyvers. Identifying emotions, intentions, and attitudes in text using a game with a purpose. In Proceedings of the naacl hlt 2010 workshop on computational approaches to analysis and generation of emotion in text, pages 71–79. Association for Computa-tional Linguistics, 2010.*
4. *Paul Ekman. An argument for basic emotions. Cognition & emotion, 6(3-4):169–200, 1992.*
5. *Janyce Wiebe, Theresa Wilson, and Claire Cardie. Annotating expres-sions of opinions and emotions in language. Language resources and evaluation, 39(2-3):165–210, 2005.*
6. *Paulo Roberto Gonc¸alves Segundo. The language of evaluation: Ap-´*
7. *praisal in english (martin, jr & white, rrr). Linha D'Agua, (21):133–137, 2008.*
8. *Carlo Strapparava and Rada Mihalcea. Learning to identify emotions in text. In Proceedings of the 2008 ACM symposium on Applied computing, pages 1556–1560. ACM, 2008.*
9. *Carlo Strapparava and Rada Mihalcea. Semeval-2007 task 14: Affective text. In Proceedings of the 4th International Workshop on Semantic Evaluations, pages 70–74. Association for Computational Linguistics, 2007.*
10. *Wenbo Wang, Lu Chen, Krishnaprasad Thirunarayan, and Amit P Sheth. Harnessing twitter" big data" for automatic emotion identification. In Privacy, Security, Risk and Trust (PASSAT), 2012 International Con-ference on and 2012 International Confernece on Social Computing (SocialCom), pages 587–592. IEEE, 2012.*
11. *Saima Aman and Stan Szpakowicz. Identifying expressions of emotion in text. In International Conference on Text, Speech and Dialogue, pages 196–205. Springer, 2007.*
12. *Clare H Liu. Applications of twitter emotion detection for stock market prediction. PhD thesis, Massachusetts Institute of Technology, 2017.*
13. *Dario Stojanovski, Gjorgji Strezoski, Gjorgji Madjarov, and Ivica Dim-itrovski. Emotion identification in twitter messages for smart city applications. In Information Innovation Technology in Smart Cities, pages 139–150. Springer, 2018.*
14. *"Tweepy". http://www.tweepy.org/.*
15. *"Thesaurus.com". http://www.thesaurus.com/.*
16. *Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. The Stanford CoreNLP natural language processing toolkit. In Association for Computational Linguistics (ACL) System Demonstrations, pages 55–60, 2014.*
17. *"Stanford Tokenizer". https://nlp.stanford.edu/software/tokenizer.shtml.*
18. *Ian H Witten, Eibe Frank, Mark A Hall, and Christopher J Pal. Data Mining: Practical machine learning tools and techniques. Morgan Kaufmann, 2016.*
19. *"Sequential minimal optimization". https://en.wikipedia.org/wiki/ sequential minimal optimization.*
20. *"J48". http://weka.sourceforge.net/doc.dev/weka/classifiers/trees/ j48.html.*

.