Automatic Detection of Emotions in Twitter Data - A Scalable Decision Tree Classification Method

Jaishree Ranganathan
University of North Carolina at Charlotte
Charlotte, NC, USA
jrangan1@uncc.edu

Nikhil Hedge
University of North Carolina at Charlotte
Charlotte, NC, USA
nhedge@uncc.edu

Allen S. Irudayaraj
University of North Carolina at Charlotte
Charlotte, NC, USA
airudaya@uncc.edu

Angelina A. Tzacheva
University of North Carolina at Charlotte
Charlotte, NC, USA
aatzache@uncc.edu

ABSTRACT

Social media data is one of the promising datasets to mine meaningful insights with applications in business and social science. Emotion mining has significant importance in the field of psychology, cognitive science, and linguistics etc. Recently, textual emotion mining has gained attraction in modern science applications. In this paper, we propose an approach which builds a corpus of tweets and related fields where each tweet is classified with respective emotion based on lexicon, and emoticons. Also, we have developed decision tree classifier, decision forest, and rule-based classifier for automatic classification of emotion based on the labeled corpus. The method is implemented in Apache Spark for scalability and BigData accommodation. Results show higher classification accuracy than previous works.

CCS CONCEPTS

• **Information systems** → *Data mining*; • **Computing methodologies** → *Machine learning approaches*;

KEYWORDS

Data Mining, Emotion Mining, Social Media, Supervised Learning, Text Processing

ACM Reference Format:

Jaishree Ranganathan, Nikhil Hedge, Allen S. Irudayaraj, and Angelina A. Tzacheva. 2018. Automatic Detection of Emotions in, Twitter Data - A Scalable Decision Tree Classification Method. In *Proceedings of ACM conference on Hypertext and Social Media (RevOpID '2018)*. ACM, New York, NY, USA, 10 pages. https://doi.org/10.475/123_4

1 INTRODUCTION

Twitter is one of the popular social networking site with more than 320 million monthly active users and 500 million tweets per day. Tweets are short text messages with 140 characters, but are powerful source of expressing emotional state and feelings with the society of friends. According to author Fox [5] emotion is discrete and consistent response to internal or external events that have a significance for the organism. Emotion is one of the aspects of our lives that influences day-to-day activities including social behavior, friendship, family, work, and many others. There are two theories related to human emotions: discrete emotion theory and dimensional model. Discrete emotion theory states that different emotions arise from separate neural systems, dimensional model states that a common and interconnected neuro-physiological system is responsible for all affective states [30].

Textual emotion mining has quite lot of applications in today's world. The applications include modern devices which sense person's emotion and suggest music, restaurants, or movies accordingly, product marketing can be improved based on user comments on products which in turn helps boost product sales.

Other applications of textual emotion mining are summarized by Yadollahi et.al [30] and include: in customer care services, emotion mining can help marketers gain information about how much satisfied their customers are and what aspects of their service should be improved or revised to consequently make a strong relationship with their end users [7]. User's emotions can be used for sale predictions of a particular product. In e-learning applications, the intelligent tutoring system can decide on teaching materials, based on user's feelings and mental state. In Human Computer Interaction, the computer can monitor user's emotions to suggest suitable music or movies [26]. Having the technology of identifying emotions enables new textual access approaches such as allowing users to filter results of a search by emotion. In addition, output of an emotion-mining system can serve as input to other systems. For instance, Rangel and Rosso [22] use the emotions detected in the text for author profiling, specifically identifying the writer's age and gender. Last but not least, psychologists can infer patients' emotions and predict their state of mind accordingly. On a longer period of time, they are able to detect if a patient is facing depression or stress [3] or even thinks about committing suicide, which is extremely useful, since he/she can be referred to counseling services [12]. Though this automatic method might help in detecting psychology related issues, it has some ethical implications as it is concerned with human emotion and their social dignity. In such cases it is always ethical to consult human psychiatrist along with the automatic systems developed.

RevOpID '2018, July 2018, Baltimore, Maryland, USA

ACM ISBN 123-4567-24-567/08/06.
https://doi.org/10.475/123_4

Emotion classification is automated using supervised machine learning algorithms. Supervised learning involves training the model with labeled instances and the model classifies the new test instances based on the training data set. Most of the previous works in this area of emotion mining [27] and [1] have used manual labeling of training data set. Authors Hasan et. al. [8] use hash-tags as labels for training data set. This work focuses on automatically labeling the data set and then use the data for supervised learning algorithms.

The previous works [27] [8] [1] have developed text classification algorithms like k-nearest neighbor and support vector machines. In this paper, we use decision tree, decision forest and rule-based decision table majority classifiers for automatic emotion classification.

In this paper, we focus on classifying emotions from tweets and developing a corpus based on the National Research Council - NRC lexicon [19] [18], National Research Council - NRC hashtag lexicon [17] [16] and emoticons. The National Research Council - NRC Emotion Lexicon is a list of words and their associations with eight emotions (anger, fear, anticipation, trust, surprise, sadness, joy, and disgust) and two sentiments (negative and positive). We experiment with several classifiers, including decision tree, random forest,and rule-based classifiers including decision table majority and prism, and choose the ones with the highest accuracy.

The reminder of the paper is structured as follows: section II related work; section III describes the methodology of data collection, pre-processing, emotion classification, feature augmentation, emotion class labeling, emotion classification, and spark; section IV we discuss the experiments, results, and evaluation; section V concludes the work.

2 RELATED WORK

This section briefly describes previous works on classifying emotion from text.

2.1 Emotion Mining From Text

Authors Kim et.al [10] proposed a comparative study for two major models in emotion mining - Discrete and Dimensional Model. For Discrete model classifier, they used Wordnet Affect lexicon and dimension reduction techniques like Latent Semantic Analysis, Probablistic Latent Semantic Analysis (LSA). To build classifier for dimensional model they used database of English affective words .According to their work no method overperforms the others on all the emotions considered.

Authors Neviarouskaya et.al [20] created a system called 'Affect Analysis Model' which is a rule based system for recognition of textual emotion. This system utilizes the database that contains emoticons, acronyms, abbreviations, affect words, interjections and modifiers along with the manually labeled emotion and intensity for each of the instances. Based on this database and rules the Affect Analysis Model identifies the given text or sentence emotion along with its intensity.

Authors Ma et.al [13] assesses the affective context from text messages. They detect emotion from chat and other dialogue messages and employ animated agents capable of emotional reasoning based on textual interaction. They used keyword spotting technique for calculating emotion estimation in text. This is a system that divides a text into words and performs an emotional estimation for each of the words, as well as a sentence-level processing technique, i.e., the relationship among subject, verb and object is extracted to improve emotion estimation. They used WordNet - Affect database to assign weights to the words according to the proportion of synsets.

2.2 Emotion Mining From Twitter Data

Twitter is one of the popular social networking site where individual can post message sharing the personal feelings and express emotion. The following works concentrate on emotion mining from Twitter data.

Authors Wang et al. [27] built a dataset from Twitter, containing 2,500,000 tweets and use hashtags as emotion labels. In order to validate the hashtag labeling, they randomly select 400 tweets to label them manually. Then they compared manual labels and hashtag labels which had acceptable consistency. They explored the effectiveness of different features such as n-grams, different lexicons, part-of-speech, and adjectives in detecting emotions with accuracy close to 60%. Their best result is obtained when unigrams,bigrams, lexicons, and part-of-speech are used together.

Authors Xia et al. [29], propose distantly supervised lifelong learning framework for Sentiment Analysis in social media text. They use following two large-scale distantly supervised social media text datasets to train the lifelong learning model: Twitter corpus (English dataset) [25], and Chinese Weibo dataset collected using Weibo API. This work focuses on continuous sentiment learning in social media by retaining the knowledge obtained from past learning and utilize the knowledge for future learning. They evaluate the model using nine standard datasets, out of which 5 are English language datasets and 4 are Chinese datasets. The main advantage of this approach is that it can serve as a general framework and compatible to any single task learning algorithms like naive bayes, logistic regression and support vector machines.

Authors Hasan et al. [8] also validate the use of hashtags as emotion labels on a set of 134,000 tweets. They compared hashtag labels with labels assigned by crowd-sourcing and by a group of psychologist's. It is found that crowd labels are not consistent within themselves; On the other-hand psychologist's labels are more consistent with hashtags. They developed a supervised classifier, named "EmoTex" which uses the feature set of unigrams, list of negation words, emoticons, and punctuation's and runs k-nearest neighbors (KNN) and support vector machines on the training data achieved 90% accuracy for multi-class emotion detection.

Authors Bollen et.al [1] analyze temporal tweets to identify emotions. They try to find the correlations on overall emotion of tweets over a period corresponding to some global event at that moment.

Authors Dos Santos and Gaitti [4] perform Sentiment Analysis of short texts using deep convolutional neural network model. They propose a model called Character to Sentence Convolutional Neural Network - CharSCNN. This model extracts relevant features from words or sentences using convolutional layers. Authors evaluate the model using movie review sentences [24] and Twitter messages [6]. They achieve accuracy close to 86%.

Authors Roberts et.al [23] created manually labeled corpus by using data extracted from Twitter based on 14 emotion evoking

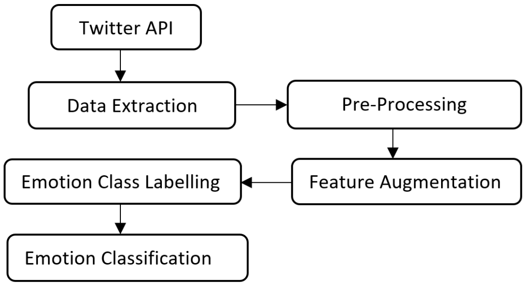


Figure 1: Methodology

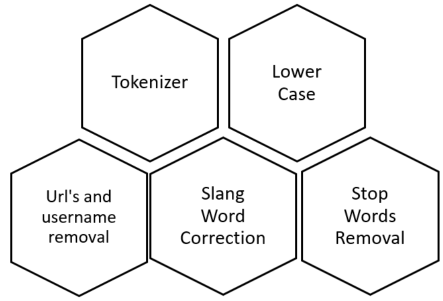
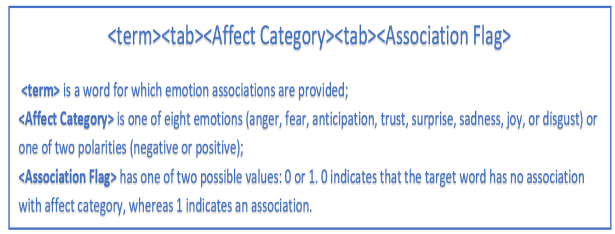


Figure 2: Pre-Processing



<term><tab><Affect Category><tab><Association Flag>

<term> is a word for which emotion associations are provided;
<Affect Category> is one of eight emotions (anger, fear, anticipation, trust, surprise, sadness, joy, or disgust) or one of two polarities (negative or positive);
<Association Flag> has one of two possible values: 0 or 1. 0 indicates that the target word has no association with affect category, whereas 1 indicates an association.

Figure 3: NRC Word Level Annotation

topics. They classified the data into seven basic emotion categories, for each of the emotion classification a separate binary support vector machine was used.They achieved best performance over the emotion 'fear'. Macro-average precision and recall for the 7 emotions is 0.721 and 0.627 respectively.

Authors Purver et.al [21] used Twitter data labeled with emoticons and hash-tags to train supervised classifiers. They used support vector machines with linear kernel and unigram features for classification. Their method had better performance for emotions like happiness, sadness, and anger but not good in case of other emotions like fear, surprise, and disgust. They achieved accuracy in the range of 60%.

3 METHODOLOGY

3.1 Data Collection

In data collection step we used Twitter streaming API [14] to collect the data with the following attributes TweetID, ReTweetCount, TweetFavouriteCount, TweetText, TweetLanguage, Latitude, Longitude, TweetSource, UserID, UserFollowersCount, UserFavoritesCount, UserFriendsCount, UserLanguage, UserLocation, UserTimeZone, IsFavorited, IsPossiblysensitive, IsRetweeted, RetweetedStatus, UserStatus, MediaEntities. We collected around 520,000 tweets as raw data. The dataset is available to other researchers to download upon request. Fig. 1. shows the overall model of the proposed methodology.

3.2 Pre-Processing

The extracted tweet text is pre-processed to make the informal text suitable for emotion classification. We lower case all the letters in the tweet; remove stop words, i.e., the most frequent words in English which will not add value to the final emotion; replace slang words with formal text, example b4 $\longrightarrow$ before, chk $\longrightarrow$ check etc; After pre-processing we have around 200,000 tweets. Fig. 2. shows the steps involved in pre-processing.

3.3 Feature Augmentation

In addition to the attributes extracted in the first step of data collection, we add additional attributes that are effective for emotion identification. After the pre-processing step, we augment the data by adding the following features: FinalEmotion, AngerScore, TrustScore, FearScore, SadnessScore, AnticipationScore, DisgustScore, SurpriseScore, JoyScore, PositiveScore, NegativeScore, AngerWordList, TrustWordList, FearWordList, SadnessWordList, AnticipationWordList, DisgustWordList, SurpriseWordList, AngerEmoticonList, TrustEmoticonList, FearEmoticonList, SadnessEmoticonList, DisgustEmoticonList, SurpriseEmoticonList, AnticipationEmoticonList, AngerHashTagList, TrustHashTagList, FearHashTagList, SadnessHashtagList, AnticipationHashTagList, DisgustHashTagList, SurpriseHashTagList. Next, we include the top four most frequent word score and most frequent verb score for each of the extracted tweets. Inside our processed dataset we found the top most frequently used words are: love, people, message and instant and the most frequent verbs are: get, going and know. The attributes LoveScore, PeopleScore, MessageScore, InstantScore, GetScore, GoingScore, KnowScore are added to each record of the dataset.

3.4 Emotion Class Labeling

To identify the emotion class, we use the National Research Council - NRC lexicon [19], [18]. The Annotations in the lexicon are at word-sense level. Each line has the format: <Term> <AffectCategory> <AssociationFlag> as shown in Fig. 3.

Apart from word level annotation, to increase the weightage of each emotion class assigned to tweet we also use the hashtags and emoticons inside the tweet text. For hashtags, we utilize the National Research Council - NRC Hashtag Emotion Lexicon [17] [16] which is a list of words and their associations with eight emotions (anger, fear, anticipation, trust, surprise, sadness, joy, and disgust). The associations are computed from tweets with emotion-word hashtags such as #happy and #anger. All emoticons were retained in the data collection process and validated while assigning weights to each emotion class for a tweet. Fig. 4. shows the list of emoticons used in this process. Fig. 5. Explains the steps involved in assigning final emotion class.

Emotion	Emoticons
sadness	>:[:-(:(:-c :c :-< :< :-[:[:{
anger	:-|| :@> :(
joy	:) ;) =) :] :P :-P ;P :D ;D :> :3 :-) ;-) :^) :o) ;^) :-D :->
surprise	:-o :-O o_O O_o :$
disgust	D:< D: D8 D; D= DX v.v

Figure 4: Emoticons

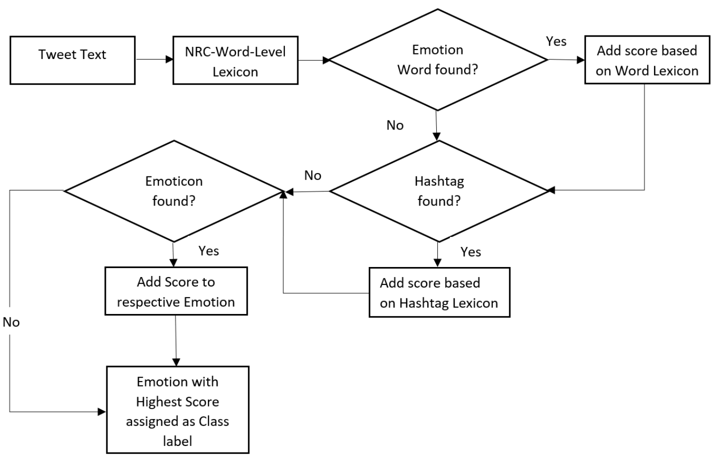


Figure 5: Emotion Labeling

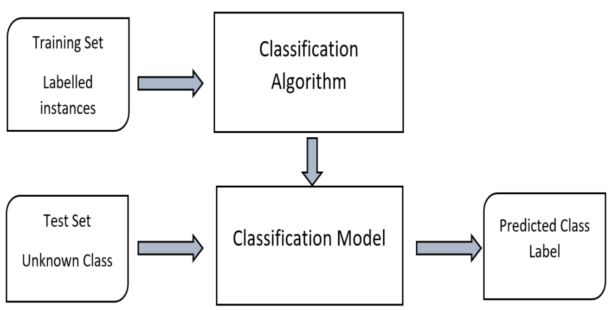


Figure 6: Classification

3.5 Emotion Classification

A systematic technique to build classification model from input data set is called Classification. It is the task of assigning objects to one of several predefined categories called class labels as shown in Fig. 6. Some of the classifiers include naive bayes, support vector machines, neural networks, decision tree classifiers, and rule-based classifiers. In this work, we use decision tree, decision forest, and decision table majority classifier. In our approach we have created the labeled dataset using the process in Fig. 5. This dataset contains close to 174,000 of the labeled instances.

Table 1: Information System S

A	B	D	E	C - Class
A1	B1	D1	E1	C1
A1	B1	D1	E2	C1
A2	B1	D1	E1	C2
A3	B2	D1	E1	C2
A3	B3	D2	E1	C2
A1	B1	D1	E2	C1
A1	B2	D1	E1	C1
A1	B3	D2	E1	C2
A3	B2	D2	E1	C2
A1	B2	D2	E2	C2
A2	B2	D1	E2	C2
A2	B1	D2	E1	C2
A1	B2	D1	E2	C1

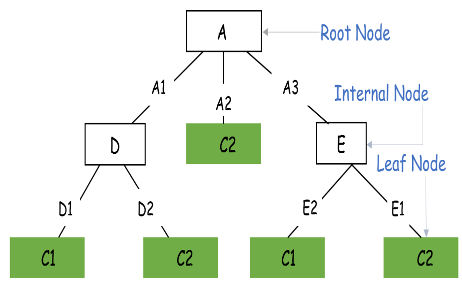


Figure 7: Decision Tree

Most of the previous works for Twitter Emotion Classification have used k-nearest neighbor and support vector machines, which obtained average classification accuracy. In attempt to improve the accuracy, in this work, we develop a decision tree classifier, decision forest, and rule-based classifier to automatically classify the tweet emotion.

3.5.1 Decision Tree Classifier. Decision tree algorithm answers a given classification question using a tree representation. The decision tree has the following nodes: root node, internal nodes, and leaf nodes. Consider the data in Table 1. Sample decision tree for the data in Table 1. is shown in Fig. 7. which depicts the nodes of the tree. The node with no incoming edges and zero or more outgoing edges is the root node. Whereas the internal node has one incoming edge and two or more outgoing edge and leaf node is one which has exactly one incoming edge and no outgoing edge. Class label is assigned to the leaf node.

The Decision Tree induction algorithm pseudo-code is given in Fig. 8

Significant improvements in classification accuracy have resulted from growing an ensemble of trees and letting them vote for the most popular class. In order to grow these ensembles, often random vectors are generated that govern the growth of each tree in the ensemble. A random forest is a classifier consisting of a collection of tree structured classifiers h(x,(θ) k), k=1, ... where the (θ) k are

```
Algorithm Decision Tree Induction Algorithm
TreeGrowth (E, F)
    if stopping.cond(E, F) = true then
        leaf = createNode()
        leaf.label = Classify(E)
        return leaf.
    else
        root = createNode()
        root.test.cond = find_best_split(E, F)
        let V = {v | v is a possible outcome of root.test.cond}
        for each v ε V do
            Ev = {e | root.test_cond(e) = v and e ε E}
            child = TreeGrowth(Ev , F)
            add child as descendent of root
            label the edge (root → child) as v
        end for
    end if
return root
```

Figure 8: Decision Tree Pseudo-Code

```
Algorithm Random Forest
Precondition: A training set S: = (x1, y1),...,(xn, yn),
features F, and number of trees in forest B.
function RandomForest (K, L)
    H←Ø
    for i ∈ 1,...,B do
        K (i) ← A bootstrap sample from S
        hi ← RandomizedTreeLearn(K (i), L)
        H ← H ∪ {hi}
    end for
    return H
end function
function RandomizedTreeLearn(K, L)
    At each node:
        f ← very small subset of L
        Split on best feature in f
    return the learned tree
end function
```

Figure 9: Random Forest Pseudo-Code

independent identically distributed random vectors and each tree casts a unit vote for the most popular class at input x [2]. Fig. 9. shows the algorithm for random forest induction. Pythagorean forest is a visualization of random forest algorithm as shown on Fig.14.

3.5.2 Rule Based Classifier. The classification of records in a rule-based classifier is based on the rule set. Rule set for the classifier

```
Algorithm DecisionTableMajority
DecisionTableMajority(Data, FeatureSet):
    TrainData, TestData ← Data [60:40]
    DecisionTable ← empty
        for each element in FeatureSet:
            Add column to DecisionTable
        end for
        for each record in TrainData:
            if recordAttributes match feature set:
                DecisionTable ← record
        end for
        for each record in TestData:
            if record matches DecisionTable:
                Create dataset of matching records
                PredictedClass ← MajorityClassLabel
            else
                PredictedClass ← MajorityClassDecisionTable
        end for
end
```

Figure 10: Decision Table Majority - Pseudo-Code

model consists of list of rules in a disjunctive normal form as in equation 1.

$$R = (r_1 V r_2 V \ldots V r_k) \tag{1}$$

R , is the Rule set
$\mathbf{r}_i$, is the classification rule

The classification rule r_i is given as below,

$$r_i : (Cond_i) \rightarrow y_i \tag{2}$$

$\mathbf{Cond}_i$, is antecedent
$\mathbf{y}_i$, is consequent

In general, the rule covers a record if the antecedent of rule matches the attributes of the record. To build a rule-based classifier, first step is to extract rule set R as in equation 1.

In this work, we use the decision table majority (DTM) rule-based classification method [11]. A DTM has two components namely schema and body. Schema is a set of features that are included in the table and body is set of labeled instances from the space defined by the features in the schema [11]. The set of features used in the schema are selected by feature selection algorithm. By using features in the schema, DTM classifier searches for exact match for any unlabeled instance. If the features of unlabeled instance match with body then the majority class is returned as the class label, otherwise the majority class of DTM is returned. The pseudo-code for the DTM algorithm is given in Fig. 10. Flow diagram for DTM is detailed in Fig. 11.

3.6 Spark

Apache Spark [15] is a popular open-source Cloud platform for large-scale data processing. Similar to DryadLINQ [31] which is a programming model for large scale distributing computing, spark

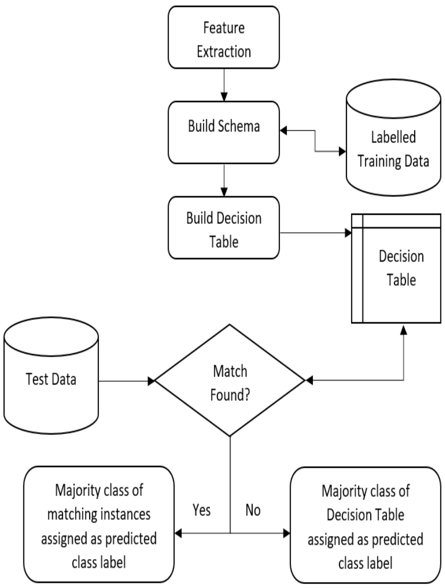


Figure 11: Decision Table Majority - Flow Chart

provides a fundamental data structure called Resilient Distributed Datasets(RDD) [32]. RDD's help achieve faster and efficient MapReduce operations. Spark ecosystem includes the following components: Spark Streaming, Spark SQL, Spark GraphX and Spark Machine Learning Library - MLlib. In this work, we use MLlib [15] to implement our decision tree classifier. We implement decision table majority method in Spark by using scala programming language.

4 EXPERIMENTS AND RESULTS

In this section we describe our experiment and results. We extract the data via Twitter streaming API [14] using Apache Spark [15] scala programming language. The raw data extracted consists of around 520,000 instances. The extracted data is pre-processed as described in Fig. 2. This results in a corpus of tweets and supporting features consisting of around 174,000 instances. As part of feature augmentation additional attributes are added to the existing corpus along with the emotion label. We use the National Research Council - NRC Lexicon [19][18] to label data with emotion class as shown on Fig. 5.

4.1 Decision Tree

The decision tree classifier is built in both WEKA Data Mining Sofware [28] and Apache Spark [15] for comparison and scalability purpose. We perform several experiments with the feature set and select the following features for decision tree classification AngerScore, TrustScore, FearScore, SadnessScore, AnticipationScore, DisgustScore, SurpriseScore, JoyScore, PositiveScore, NegativeScore, LoveScore, PeopleScore, MessageScore, InstantScore, GetScore, KnowScore, GoingScore, Source, UserFollowers, UserFavorite, UserFriends, UserLanguage, isPossiblySensitive, MediaEntities.

Table 2: Weka Decision tree - Confusion Matrix

A	B	C	D	E	F	G	H	
15818	0	2	1	4	1	0	6	A - Sadness
7	10328	2	24	6	7	5	9	B - Joy
9	10	3050	3	4	2	6	6	C - Fear
1	1	0	20201	5	2	2	0	D - Anticipation
3	4	2	7	9130	5	1	3	E - Trust
3	9	8	2	3	2267	1	2	F - Surprise
7	6	14	5	5	5	4082	12	G - Anger
5	11	2	2	1	5	8	4734	H - Disgust

Table 3: Precision,Recall,F-Measure - Weka Decision Tree

Measure	Sadness	Joy	Fear	Anticipation	Trust	Surprise	Anger	Disgust
Precision	0.998	0.996	0.990	0.998	0.997	0.988	0.994	0.992
Recall	0.999	0.994	0.987	0.999	0.997	0.988	0.987	0.993
F-Measure	0.998	0.995	0.989	0.999	0.997	0.988	0.991	0.992

Table 4: Spark Decision tree - Confusion Matrix

A	B	C	D	E	F	G	H	
20117	18	0	226	0	38	0	0	A-Anticipation
0	14930	23	77	0	101	436	0	B - Sadness
354	433	9326	271	12	25	13	0	C - Joy
9	29	172	8696	159	111	0	0	D - Trust
51	327	15	41	4106	153	38	0	E - Disgust
22	19	32	16	58	4058	0	0	F - Anger
204	94	36	1892	90	175	632	0	G - Fear
204	181	41	1784	41	38	15	0	H - Surprise

Table 5: Precision,Recall,F1-Score - Spark Decision tree

Measure	Anticipation	Sadness	Joy	Trust	Disgust	Anger	Fear	Surprise
Precision	0.9597	0.9313	0.9669	0.6687	0.9193	0.8635	0.5573	0
Recall	0.9861	0.9590	0.8938	0.9476	0.8678	0.9650	0.2023	0
F1-Score	0.9727	0.9449	0.9289	0.7841	0.8928	0.9115	0.2969	0

The dataset is split into train and test dataset with the ratio of 60 and 40 respectively. The decision tree model is trained with the train set, the model's accuracy is validated by using the test dataset.

4.1.1 WEKA. We build a decision tree classifier J48 model in WEKA Data Mining Sofware [28] for our Twitter emotion dataset. We achieve accuracy of 99.6% with WEKA's decision tree. The confusion matrix and evaluation measures are shown is shown in Table 2 and Table 3.

4.1.2 Spark. In order to build the decision tree with Spark we use the Machine Learning Library MLLib - 'DecisionTreeClassifier' to train the model. We use scala programming language. We test with both Spark cluster single node instance, and Spark cluster with 6 nodes. The Spark cluster is installed over Hadoop YARN, and the 6 nodes are connected via 10 GigaBits per second Ehternet network. Visualization of the decision tree is shown is Fig. 12. and Fig. 13. With this model we achieve accuracy of 88.45% for emotion classification of Twitter dataset for both single node and 6 node cluster configuration . Table 4. shows the confusion matrix and the evaluation measures are shown in Table 5.

Table 6: Decision Tree Execution Time in Seconds - Spark Single Node, Spark 6 Nodes

Number of Instances	Spark Single Node (Secs)	Spark 6 Nodes (Secs)
174689	59.33	46.57

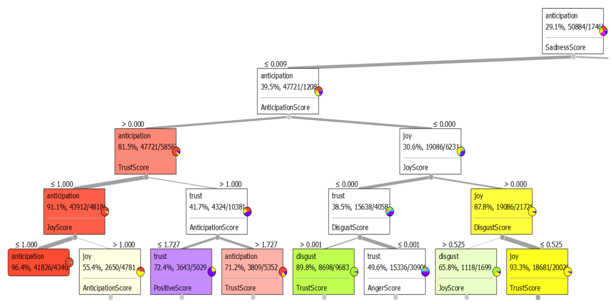


Figure 12: Decision Tree Left Side - Class Emotion - Twitter Dataset

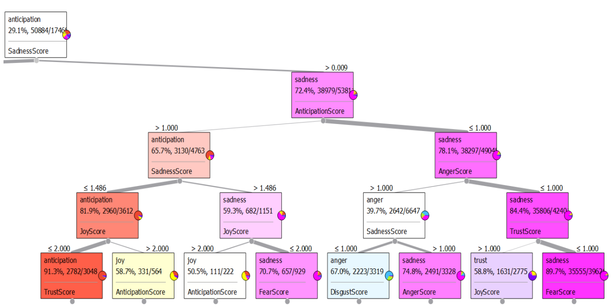


Figure 13: Decision Tree Right Side - Class Emotion - Twitter Dataset

The average execution time results for Spark single node and 6 nodes are shown in Table 6.

4.2 Decision Forest - Random Forest

4.2.1 WEKA. We build a decision forest - random forest classifier model in WEKA Data Mining Sofware [28] for our Twitter emotion dataset. We achieve accuracy of 88.8% with WEKA's DecisionForest. The confusion matrix is shown in Table 7. The evaluation results with precision, recall and F-measure is given in Table 8.

A visualization of the decision forest - random forest - is the pythagorean forest, as shown on Fig.14.

In Table 5 and Table 8, we see that precision and recall of 'fear' and 'surprise' emotion are lowest compared to 'anticipation' and 'sadness'. We infer that the number of instances of training data for emotion 'fear' and 'surprise' is low, compared to the rest of the class labels, so the training model does not capture many correlations in the features. Fig.15. shows a tree map based on the number of instances in each emotion class. Therefore increasing the number of instances in the training set would improve the classifier accuracy.

Table 7: Weka Decision Forest - Confusion Matrix

A	B	C	D	E	F	G	H	
15650	1	0	0	81	0	100	0	A-Sadness
34	9686	0	475	155	0	28	10	B - Joy
735	52	0	190	1860	0	173	80	C - Fear
19	0	0	20146	18	0	29	0	D - Anticipation
33	163	0	274	8452	0	88	145	E - Trust
203	55	0	193	1797	0	19	28	F - Surprise
20	22	0	19	20	0	3987	68	G - Anger
352	34	0	56	51	0	114	4161	H - Disgust

Table 8: Precision,Recall,F-Measure - Weka Decision Forest tree

Measure	Anticipation	Sadness	Joy	Trust	Surprise	Disgust	Anger	Fear
Precision	0.943	0.918	0.967	0.68	0.0	0.926	0.879	0.0
Recall	0.997	0.989	0.932	0.923	0.0	0.873	0.964	0.0
F-Measure	0.969	0.952	0.95	0.783	0.0	0.899	0.919	0.0

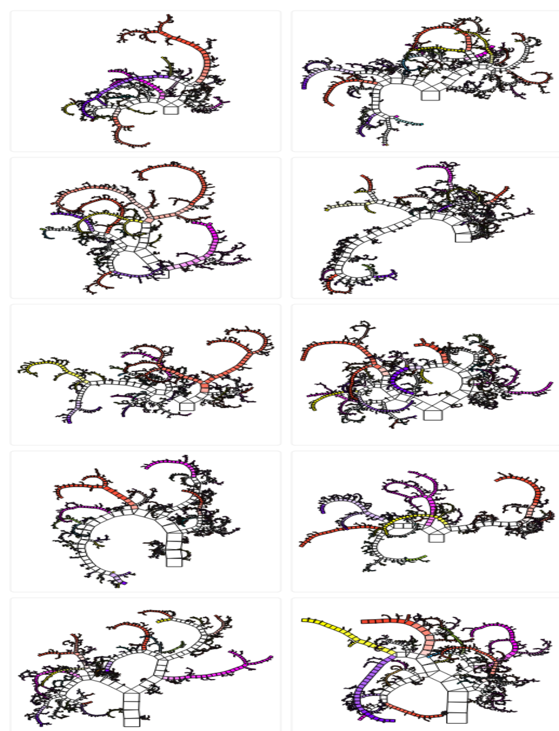


Figure 14: Decision Forest - Pythagorean - Class Emotion - Twitter Dataset

4.3 Decision Table Majority

We implement the decision table majority as a rule-based classification method [11] for comparison purpose. We analyze several rule-based classification methods with our Twitter dataset for their accuracy, running time, and feasibility of implementation on a cloud clustered environment including: ZeroR, OneR [9], Decision Table

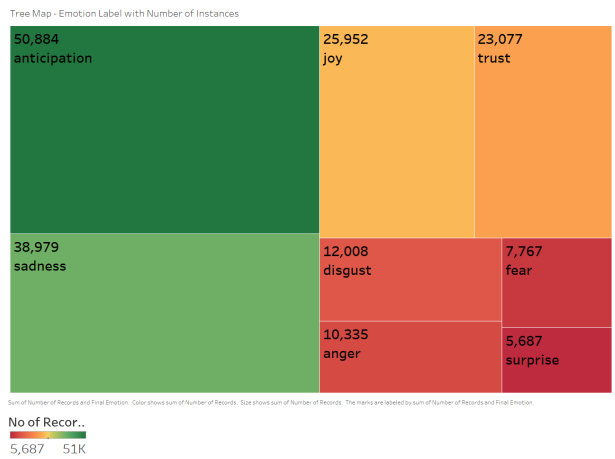


Figure 15: Tree Map - Corpus - Emotion Class Label Distribution

Table 9: Rule Based Classifier - Analysis

Algorithm	Accuracy	Running Time (Seconds)
ZeroR	28.92	0.28
OneR	49.76	0.89
Decision Table	96.45	212.78

Table 10: Weka Decision Table Majority - Confusion Matrix

A	B	C	D	E	F	G	H	Class
15498	0	9	302	0	0	17	6	A-Sadness
174	9861	0	332	0	8	0	13	B - Joy
134	0	2797	156	0	0	3	0	C - Fear
217	52	0	19942	1	0	0	0	D - Anticipation
118	8	0	221	8808	0	0	0	E - Trust
38	6	0	88	2	2161	0	0	F - Surprise
128	0	2	138	0	2	3861	5	G - Anger
110	0	0	180	0	4	5	4469	H - Disgust

Table 11: Weka Precision,Recall,F-Measure - Decision Table Majority

Measure	Sadness	Joy	Fear	Anticipation	Trust	Surprise	Anger	Disgust
Precision	0.944	0.993	0.996	0.934	1	0.994	0.994	0.995
Recall	0.979	0.949	0.905	0.987	0.962	0.942	0.934	0.937
F-Measure	0.961	0.971	0.948	0.959	0.981	0.967	0.963	0.965

[11]. The Accuracy results are shown in Table 9. Based on the results, we see the decision table majority classifier produces the best accuracy for our Twitter emotion dataset.

4.3.1 WEKA. We build a decision table majority classifier model in WEKA Data Mining Software [28] for our Twitter emotion dataset. We achieve accuracy of 96.45% with WEKA's decision table majority. The confusion matrix is shown in Table 10. The evaluation measures are shown in Table 11

(AngerScore=>0 AND TrustScore=>3 AND FearScore=>0 AND SadnessScore=>0 AND AnticipationScore=>1 AND DisgustScore=>0 AND SurpriseScore=>0 AND JoyScore=>4 AND PositiveScore=>4 AND NegativeScore=>0 AND LOVE_SCORE =>0→ **FinalEmotion=joy**)

Figure 16: Decision Table - Sample Rule

Table 12: Spark Decision Table Majority - Confusion Matrix

A	B	C	D	E	F	G	H	Class
14591	0	0	1120	0	0	0	0	A-Sadness
0	9160	0	963	0	0	0	0	B - Joy
0	0	2614	572	0	0	0	0	C - Fear
0	0	0	20090	0	0	0	0	D - Anticipation
0	0	0	694	8581	0	0	0	E - Trust
0	0	0	221	0	1980	0	0	F - Surprise
0	0	0	586	0	0	3783	0	G - Anger
0	0	0	536	0	0	0	4385	G - Disgust

4.3.2 Spark. The schema of decision table is the features in the data which contribute to maximum accuracy. We use filter based feature selection algorithm in WEKA Data Mining software [28]. Some of the algorithms to extract the features for decision table Significance are: attribute evaluator, chi-squared attribute evaluator, Gain ratio attribute evaluator, greedy stepwise attribute evaluator, and filter attribute evaluator. Among the listed algorithms, gain ratio attribute evaluator is most appropriate for the given dataset. We use the top 11 features from the entire Gain Ratio list. This selection is based on the accuracy yielded by using the selected features on the decision table majority algorithm.

The list of selected features for decision table majority algorithm using gain ratio feature selection algotithm is: AngerScore, TrustScore, FearScore, SadnessScore, AnticipationScore, DisgustScore, SurpriseScore, JoyScore, PositiveScore, NegativeScore,LoveScore and FinalEmotion.

In order to build the decision table majority classifier with Spark, we design the schema based on the above features. Train data with class labels are loaded as decision table based on the schema. First the decision table is loaded. Then the decision table induction matches the test data as per the decision table schema. A sample rule produced by the decision table is shown on in Fig.16. If matching records are identified then the algorithm returns the class with largest number of matching instances. Otherwise the algorithm returns the default class, which is usually the class with highest number of records in the Decision Table schema. According to Fig.15 we observe that the default class in our data is 'anticipation'.

This method produces classification accuracy of 93.28% for our Twitter emotion Dataset. The confusion matrix is shown in Table 12, and the Table 14 shows the evaluation measures of precision, recall and F-measure for each of the emotion class labels.

Decision table majority is implemented in Apache Spark [15] using Scala programming language. We test in a single node cluster, and 6 nodes cluster configuration. Results show that the execution time is faster in 6 node cluster when compared to a single node. The average execution times are shown in Table 13.

The Table 15, shows the accuracy obtained by the three models used in our work.

Table 13: Decision Table Majority - Average Execution times in Seconds - WEKA, Spark Single Node, Spark 6 Node Cluster

Number of Instances	Spark Single Node (Secs)	Spark 6 Node (Secs)
174689	62.42	37.39

Table 14: Spark Precision,Recall,F1-Score - Decision Table Majority

Measure	Sadness	Joy	Fear	Anticipation	Trust	Surprise	Anger	Disgust
Precision	1	1	1	0.8106	1	1	1	1
Recall	0.9287	0.9048	0.8204	1	0.9251	0.8995	0.8656	0.8910
F1-Score	0.9630	0.95	0.9013	0.8954	0.9611	0.9471	0.9281	0.9424

Table 15: Result comparison

Model	Accuracy
Decision Tree	88.45% - 99.6%
Decision Forest	88.8%
Decision Table Majority	93.28% - 96.45%

5 CONCLUSIONS

In this work, we perform automatic detection of emotions in Twitter dataset. We utilize the National Research Council - NRC Emotion Lexicon to label the Emotion class for our data. We examine several classifiers and choose the decision tree and decision forest (random forest) as well as the decision table majority methods. These methods have not been used before for Twitter emotion classification. We report higher classification accuracy than any previous works. Our accuracy is 88.45% - 99%, compared to 60% - 90% for previous works, which mostly use the support vector machines and k-nearest neighbor classifiers. We implement the data collection, pre-processing, feature augmentation, and the proposed classifiers on both WEKA and Apache Spark system over Hadoop cluster for scalability purpose. Our Spark implementation is able to scale to BigData sets, as data is divided into partitions and is processed in parallel at each cluster node. Applications of this work include detection of emotions for: improving customer satisfaction, e-learning, psychological health care, and designing intelligent phones and devices which recognize user emotion. In the future, we plan to perform actionable pattern mining on our Twitter Emotion dataset to suggest ways to alter the user emotions from negative to positive sentiment.

ACKNOWLEDGMENTS

The authors would like to thank Saif Mohammed from National Research Council for providing the National Research Council - NRC Emotion Lexicon. Authors would like to thank Jonathan Halter for his support and administration of the Apache Spark on Hadoop cluster at UNCCharlotte.

REFERENCES

[1] Johan Bollen, Huina Mao, and Alberto Pepe. 2011. Modeling public mood and emotion: Twitter sentiment and socio-economic phenomena. *Icwsm* 11 (2011), 450–453.
[2] Leo Breiman. 2001. Random forests. *Machine learning* 45, 1 (2001), 5–32.
[3] Munmun De Choudhury, Michael Gamon, Scott Counts, and Eric Horvitz. 2013. Predicting depression via social media. *ICWSM* 13 (2013), 1–10.
[4] Cicero dos Santos and Maira Gatti. 2014. Deep convolutional neural networks for sentiment analysis of short texts. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*. 69–78.
[5] Elaine Fox. 2008. *Emotion science cognitive and neuroscientific approaches to understanding human emotions.* Palgrave Macmillan.
[6] Alec Go, Richa Bhayani, and Lei Huang. 2009. Twitter sentiment classification using distant supervision. *CS224N Project Report, Stanford* 1, 12 (2009).
[7] Narendra Gupta, Mazin Gilbert, and Giuseppe Di Fabbrizio. 2013. Emotion detection in email customer care. *Computational Intelligence* 29, 3 (2013), 489–505.
[8] Maryam Hasan, Emmanuel Agu, and Elke Rundensteiner. 2014. Using hashtags as labels for supervised learning of emotions in twitter messages. In *ACM SIGKDD Workshop on Health Informatics, New York, USA*.
[9] Robert C Holte. 1993. Very simple classification rules perform well on most commonly used datasets. *Machine learning* 11, 1 (1993), 63–90.
[10] Sunghwan Mac Kim, Alessandro Valitutti, and Rafael A Calvo. 2010. Evaluation of unsupervised emotion models to textual affect recognition. In *Proceedings of the NAACL HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text*. Association for Computational Linguistics, 62–70.
[11] Ron Kohavi. 1995. The power of decision tables. In *European conference on machine learning*. Springer, 174–189.
[12] Kim Luyckx, Frederik Vaassen, Claudia Peersman, and Walter Daelemans. 2012. Fine-grained emotion detection in suicide notes: A thresholding approach to multi-label classification. *Biomedical informatics insights* 5 (2012), BII–S8966.
[13] Chunling Ma, Helmut Prendinger, and Mitsuru Ishizuka. 2005. Emotion estimation and reasoning based on affective textual interaction. In *International Conference on Affective Computing and Intelligent Interaction*. Springer, 622–628.
[14] Kevin Makice. 2009. *Twitter API: Up and running: Learn how to build applications with the Twitter API.* " O'Reilly Media, Inc.".
[15] Xiangrui Meng, Joseph Bradley, Burak Yavuz, Evan Sparks, Shivaram Venkataraman, Davies Liu, Jeremy Freeman, DB Tsai, Manish Amde, Sean Owen, et al. 2016. Mllib: Machine learning in apache spark. *The Journal of Machine Learning Research* 17, 1 (2016), 1235–1241.
[16] Saif M Mohammad. 2012. # Emotional tweets. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics-Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation*. Association for Computational Linguistics, 246–255.
[17] Saif M Mohammad and Svetlana Kiritchenko. 2015. Using hashtags to capture fine emotion categories from tweets. *Computational Intelligence* 31, 2 (2015), 301–326.
[18] Saif M Mohammad and Peter D Turney. 2010. Emotions evoked by common words and phrases: Using Mechanical Turk to create an emotion lexicon. In *Proceedings of the NAACL HLT 2010 workshop on computational approaches to analysis and generation of emotion in text*. Association for Computational Linguistics, 26–34.
[19] Saif M Mohammad and Peter D Turney. 2013. Crowdsourcing a word–emotion association lexicon. *Computational Intelligence* 29, 3 (2013), 436–465.
[20] Alena Neviarouskaya, Helmut Prendinger, and Mitsuru Ishizuka. 2007. Textual affect sensing for sociable and expressive online communication. In *International Conference on Affective Computing and Intelligent Interaction*. Springer, 218–229.
[21] Matthew Purver and Stuart Battersby. 2012. Experimenting with distant supervision for emotion classification. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, 482–491.
[22] Francisco Rangel and Paolo Rosso. 2016. On the impact of emotions on author profiling. *Information processing & management* 52, 1 (2016), 73–92.
[23] Kirk Roberts, Michael A Roach, Joseph Johnson, Josh Guthrie, and Sanda M Harabagiu. 2012. EmpaTweet: Annotating and Detecting Emotions on Twitter.. In *LREC*, Vol. 12. Citeseer, 3806–3813.
[24] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*. 1631–1642.
[25] Duy Tin Vo and Yue Zhang. 2016. Don't Count, Predict! An Automatic Approach to Learning Sentiment Lexicons for Short Text. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, Vol. 2. 219–224.
[26] S Voeffray. 2011. Emotion-sensitive human-computer interaction (HCI): State of the art-Seminar paper. *Emotion Recognition* (2011), 1–4.

[27] Wenbo Wang, Lu Chen, Krishnaprasad Thirunarayan, and Amit P Sheth. 2012. Harnessing twitter" big data" for automatic emotion identification. In *Privacy, Security, Risk and Trust (PASSAT), 2012 International Conference on and 2012 International Confernece on Social Computing (SocialCom)*. IEEE, 587–592.

[28] Ian H Witten, Eibe Frank, Mark A Hall, and Christopher J Pal. 2016. *Data Mining: Practical machine learning tools and techniques.* Morgan Kaufmann.

[29] Rui Xia, Jie Jiang, and Huihui He. 2017. Distantly Supervised Lifelong Learning for Large-Scale Social Media Sentiment Analysis. *IEEE Transactions on Affective Computing* 8, 4 (2017), 480–491.

[30] Ali Yadollahi, Ameneh Gholipour Shahraki, and Osmar R Zaiane. 2017. Current state of text sentiment analysis from opinion to emotion mining. *ACM Computing Surveys (CSUR)* 50, 2 (2017), 25.

[31] Yuan Yu, Michael Isard, Dennis Fetterly, Mihai Budiu, Úlfar Erlingsson, Pradeep Kumar Gunda, and Jon Currey. 2008. DryadLINQ: A System for General-Purpose Distributed Data-Parallel Computing Using a High-Level Language.. In *OSDI*, Vol. 8. 1–14.

[32] Matei Zaharia, Mosharaf Chowdhury, Tathagata Das, Ankur Dave, Justin Ma, Murphy McCauley, Michael J Franklin, Scott Shenker, and Ion Stoica. 2012. Resilient distributed datasets: A fault-tolerant abstraction for in-memory cluster computing. In *Proceedings of the 9th USENIX conference on Networked Systems Design and Implementation*. USENIX Association, 2–2.