Classifier Ensemble for Opinion Extraction of Movie Review Using NLTK

A DISSERTATION

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE AWARD OF DEGREE

OF

MASTER OF TECHNOLOGY

IN

COMPUTER SCIENCE & ENGINEERING

Submitted By:

RITU RAJ

2K17/CSE/15

Under the supervision of

Dr. RAJESH KUMAR YADAV

(Assistant Professor)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College Of engineering)

Bawana Road, Delhi-110042

JUNE, 2019

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College Of engineering)

Bawana Road, Delhi-110042

DECLARATION

I, Ritu Raj Roll No. 2K17/CSE/15 student of M.Tech (Compter Science & Engineering), hereby declare that the Project Dissertation titled "***Classifier Ensemble for Opinion Extraction of Movie Review Using NLTK***" is submitted by me to the Department of Computer Science & Engineering, Delhi Technological University in partial fulfillment for the requirement of the award of degree of Master of Technology for the requirements of the award of degree of Master of Technology, is original and not copied from any source without proper citation. This work has not previously formed the basis for the award of any Degree, Diploma Associateship, Fellowship or other similar title or recognition.

Place: DTU, Delhi
Date: 25-06-2019

Ritu Raj
(2K17/CSE/15)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College Of engineering)

Bawana Road, Delhi-110042

CERTIFICATE

I hereby certify that the Project Dissertation titled *"**Classifier Ensemble for Opinion Extraction of Movie Review Using NLTK**"* which is submitted by Ritu Raj, Roll No. 2K17/CSE/15, Department of computer Science & Engineering, Delhi Technological University, Delhi in partial fulfillment for the requirement of the award of degree of Master of Technology (Computer Science and Engineering) is a record of a project work carried out by the student under my supervision. To the best of my knowledge this work has not been submitted in part or full for any Degree or Diploma to this University or elsewhere.

Place: Delhi

Date: 25-06-2019

(Dr. Rajesh Kumar Yadav)
SUPERVISOR
Assistant Professor

Dept. of Computer Engineering

Delhi Technological University

ABSTRACT

Opinion Extraction or Sentiment Analysis is a contextual text mining that identifies and extracts subjective information from source information, helping a business to acknowledge its brand, product or service social sense. The social platform is in a boom which gives a large audience an overview. Opinion is made in connection with everything on the Internet. This work focuses on the extraction of opinions from films also known as sentimental analysis. A classification ensemble based on the supervisioned machine learning algorithm is generated using the NLTK Toolkit to produce a fresh classifier. This improves overall accuracy. The results obtained are demonstrated further with using matplotlib.

ACKNOWLEDGEMENT

I am most thankful to my family for constantly encouraging me and giving me unconditional support while pursuing this research.

I am extremely grateful to **Dr. Rajesh Kumar Yadav** Asst. Professor, Department of Computer Science Engineering, Delhi Technological University, Delhi for providing invaluable guidance and being a constant source of inspiration throughout my research. I will always be indebted to her for the extensive support and encouragement she provided.

I also convey my heartfelt gratitude to the research scholars at Delhi Technological University, for their valuable suggestions and helpful discussions throughout the course of this research work.

Ritu Raj

Roll No. 2K17/CSE/15

TABLE OF CONTENTS

List of figures

List of Tables

LIST OF ACRONYMS

A	Accuracy
Adj	Adjective
Adv	Adverb
BoVW	Bag of Visual Words
CNN	Covolution Neural Network
Emoti	Emoticon
F	F-measure
POS	Part-of-speech
LR	Logistic Regression
ML	Machine Learning
NB	Naive Bayesian
NLP	Natural Language Processing
P	Precision
R	Recall
RF	Random Forest
SM	Social Media
SVM	Support Vector Machine
URL	Uniform Resource Locator
Vb	Verb

CHAPTER 1
INTRODUCTION

With the fast development of the World Wide Web, people often transmit emotions via social media, blogs, ratings and reviews over the Internet. The Internet is a huge source of information and, thanks to this growth in text information, it is necessary to understand the main idea of expressing opinions and calculating thoughts for business exploration. To identify new business strategies, owners employ to discover the rating in the market [1]. In the field of competition many researchers are getting employed to help decode the business for improvement. Data scientists are the latest buzz in the field of employment related to data mining and extraction.

Opinion Extraction is one of natural language processing's most trendy apps. The main reason is that bipolar text characteristics are usually identified by a sentence or a review. For instance, checking whether the book's writer is male or female, checking whether certain posts on a social site are being bullied or not, etc [2]. Different social platforms provide the user with a means to provide reviews which, based on the positives or negative, can have a significant effect on the overall business. Responses, opinions, values and forecasts are often indicative of the status of people as they consist of conceptions expressed in a subjective language [3].

Sentiment analysis or Opinion extraction in a nutshell is the analysis of different feelings such as opinion, attitude, thoughts, emotions etc. These task is backed by Natural Language Processing toolkits. Natural language processing is a branch of data science in which with the systematic processing of text data helps in understanding and deriving information from it. By integrating NLP and its different components one can organize the huge chunk of data and perform a wide range of task to solve problems from different aspect of field such as speech recognition, opinion extraction, cyber bullying, automatic summarization, topic segmentation, stock markets etc.

Natural language processing tasks is solved using machine learning techniques. Machine learning refers to the technique in which the vast chunk of data is dealt to gain some insight from it by applying an algorithm along with it. The basic concepts of Machine learning and related algorithm is described in the next segment.

This thesis work is divided into five chapters. The first chapter is the introduction in which the online scrutinizing of data is important for the business is described. In this some concepts of machine learning is also explained along with the objective is detailed. The second chapter is based on the Related work . In this some previous works done on this background is described and cited well with the authors. The third chapter is the proposed methodology.

In this how the the data is extracted and will be executed is described. Basically in this part the ensembling concepts and the pseudo code is explained. The fourth chapter is the implementation and result. In this all the setup of the libraries required is explained. Further the result on the review is generated and the graphical representation of he code using Matplotlib is detailed. The last chapter is the conclusion. In this the future scope of the work done is described.

1.1 Main Concepts of Machine Learning

Machine learning is a technique of programming which goes beyond the traditional form programming style. This can be with the simple exmaple. Suppose a programme has to be designed to check the grammar of English. Using traditional style it will be coded using all the rules that are required. Now if the same grammar checking algorithm has to be designed for Hindi language then sperate coding has to be done. This is a huge labor of both manpower and time. Now with the unconventional style of coding that is machine learing a data need to be gathered, features on which the result has to be obtained is designed, model is chosen and on that training and testing set can be obtained.

To understaond the above concepts in deatil it is explained as further in points. Some of the fundamental concepts are explained below that forms the base of our work. Machine learning predicts things based on patterns it has been trained with. In this context, classification is the process where computers group data together based on predetermined variables, also called features. When dealing with a classification problem, it is important to always keep in mind that 100% accuracy cannot be achieved. ML is used to obtain classifications quickly and automatically with as much precision as possible [4].

1. Data Collection : In the beginning, data can be collected in the form of a text or excel sheet. The better the variety, density and the quantity of data, the better the learning prospects for the machine. The better the data is gained by a computer.

2. Data Preparation : In this setp data is pre-processsed before extraction of the features and training it on any particlular algorithm. It takes time to determine the quality of data.

3. Model Training: In this step the appropriate algorithm is chosen. A model is used which is a mathematical representation of a real-world process that is obtained by combining features and one algorithm of ML. The processed data is spilt into two parts. The training set and the testing set. Training set is a subset of the input and is used to train the model. Testing set part of data help in determining how well an algorithm performs.

Some machine learning algorithms are described below which were used in the study as well in the implemention of the project. Broadly speaking machine learning can be categoried into Supervised, Unsupervised and Reinforcement.

1.1.1 Decision Tree

Decision trees helps in the problems related to classification. This is a simple technique for interpretation and in prediction of a qualitative response. It is a type of Supervised learning algorithm. This algorithm inherits the name from its structure. It can work on both the continuous as well as discrete data. Each node that is not a leaf is associated with a feature test, also known as split. It works in such a manner that it creates as many distinct group as possible [5].

The feature space is divided into a number of simple regions R1 , R2 , ..., Rn and the set of decisions are represented in a tree. For exmple in the given diagram below Fig 1.1 the regions are divided as the leaf node.

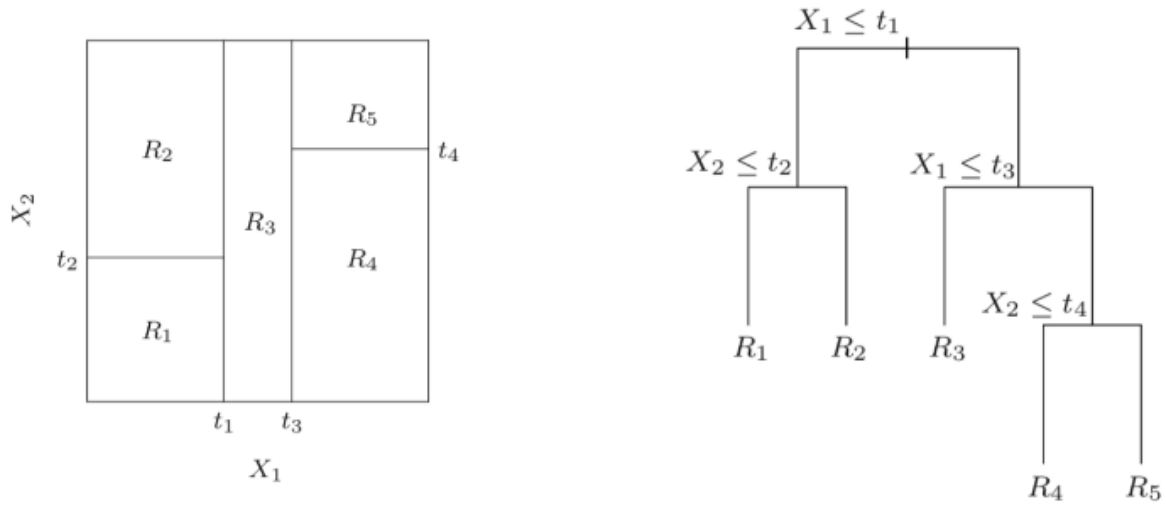
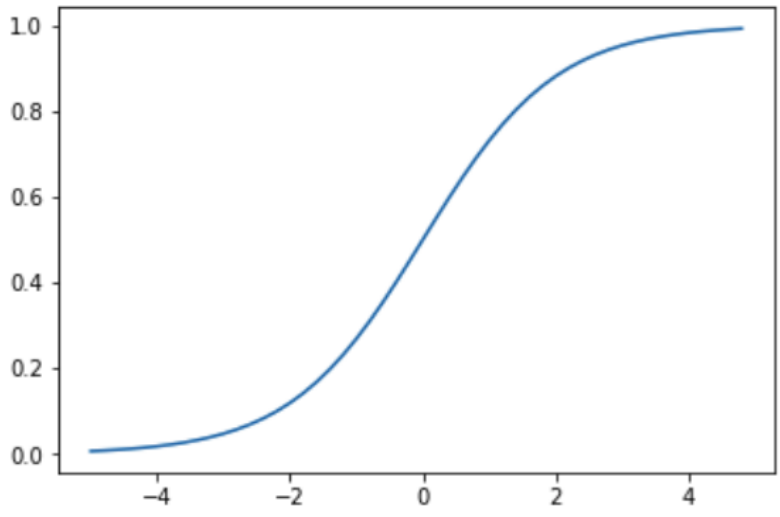


Fig 1.1 - Decision Tree feature space divided into regions.

1.1.2 Logistic Regression

Logistic Regression is another classification technique that can be used in both Supervised as well as Unsupervised learning. This is used in estimation of discrete values which is based on the given sets of independent values. It predicts the probability and its output lies in the range of 0 to 1 [6]. It is a sigmoid function, which takes any input x ∈ R and outputs a value in [0, 1] , this is shown in Fig1.2.



y-axis (σ(x))

x- axis (x)

Fig 1.2-Sigmoid Function graph for Logistic Regression.

The logistic function σ(x) is defined as follows:

$$\sigma(x) = exp(x)/ (1+exp(x)) = 1/ (1+exp(-x)) \quad (1)$$

Taking t as a linear function in two dimensions, where $x = \beta 0 + \beta 1x$, then the logistic function can be rewritten as:

$$p(x)= 1/ (1+exp(-(\beta_0 +\beta_1 x))) \quad (2)$$

In the above equation p(x) is a probability that ranges from 0 to 1. In a classification problem, with y ∈ {0, 1}, p(x) can be seen as the probability that a new point belongs to class 1:

$$p(x) = P(G=1| X=x) \quad (3)$$

Therefore, predicting the class is equivalent to finding the value of β0 and β1 . The logit (natural logarithm of the odds) is introduced as:

$$g(p(x)) = logit\ (p(x)) = ln\ (p(x)/ (1- p(x))) = \beta_0 + \beta_1 x$$
$$p(x)/1 - p(x) = exp(\beta_0 + \beta_1 x) \quad (4)$$

1.1.3 Naive Bayes classifiers

Naive Bayes is one of the simplest classification technique that is based on Bayes theorem. It is based on the assumption that predictors are independent. In the general form the Bayes theorem is:

$$P(A|B) = P(B|A)P(A)/ P(B) \quad (5)$$

Here A and B are events . P (A|B) is the posterior probability of class given predictor, P(B) is the prior probability and P (B|A) is the likelihood conditional probabilities, while P(A) and P(B) are marginal probabilities.

Considering a partition $A1 , ..., An\ of\ \Omega\ (A\ i\ A\ j\ 6 = \varnothing\ \forall i\ 6 = j\ and\ i=1\ A\ i\ =\Omega)$ the previous equation can be rewritten:

$$P(A_i|B) = P(B|A_i)P(A_i) / P(B) = P(B|A_i)\ P(A_i) / \sum^{n}_{j=1} P(B|A_j)\ \ P(A_j)\ \ \forall i = 1,\ldots.,n \quad (6)$$

One of the main algorithms among Naive Bayes classifiers is Multinomial Naive Bayes, that is one of the most popular in the field of text analysis. Comparing to the previous algorithm, learning in such a model is faster than for a logistic regression classifier. It introduces multinomial distribution to compute the probabilities in situations where there are more than two possible outcomes. Although it is a simple method, it is very competitive in data mining after an appropriate preprocessing. In this context the features are the frequency of words (as will be explained in 4.2). It assumes that the value of a particular feature is independent from the value of others. This implies that correlations between features are not considered. Hence, it is called naive.

Considering a set of predictors *X* as *(x1 , ..., xd)* and the class y, the goal is to find the probability of class y given the vector of features X [6].

In a two classes problem, say 0 and 1 as labels, the algorithm predicts class one if $P(Y = 1|X)$ > 0.5, class zero otherwise. The idea of the algorithm is the same in multi classification problems, in which the class with the largest probability is chosen.
Hence,

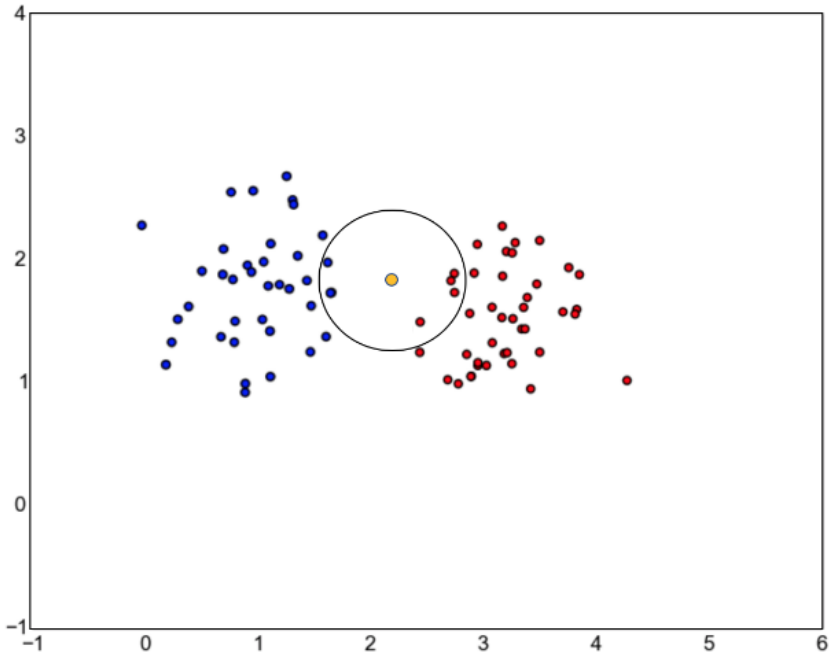
$$y = argmax_{y} P(Y = y) \prod_{i=1}^{d} P(x_i \mid y) \qquad (7)$$

1.1.4 K- nearest neighbors

KNN is a popular choice in the field of industry for classification . However it can be used for both regression as well as classification problem. K nearest neighbors is a simple algorithm that determines the k nearest neighbor from its origin point using a distance function based on the majority vote caste to it.

To understand how this algorithm works consider the below diagram. In this for k=6, six neighbors are considered. It can be seen in the circle that four points are red, while two are blue. So, the new point is classified by majority vote as red as in Fig 1.3.

x2- plane



x1- plane

Fig 1.3- KNN point classification exmple.

In this the concept of distance is used to formulate the k- neighbors [7]. The Euclidean distance is widely used approach to calculate distance between two given points.

$$d(\vec{x},\vec{y}) = \left(\sum_{i=1}^{n} |x_i - y_i|^p\right)^{1/p} , \tag{8}$$

In the above formula the distance between x and y is given. Here p is a postive number to obtain three different popular distances.

In this ,p = 1 gives the Manhattan distance, p = 2 gives the Euclidean distance and p = ∞ gives the Chebychev distance.

Mahalanobis is another famous measure that evaluates the distance between a point P and a distribution D. This is done by measuring how many standard deviations are from P to the mean of D. So, the Mahalanobis distance of an observation *~x = (x1 , x2 , . . . , xN)* T from a set of observations with mean *μ = (μ1 , μ2 , . . . , μN)* T and covariance matrix S is defined in a vector notation as:

$$D_M(\vec{x}) = \sqrt{(\vec{x} - \vec{\mu})^T S^{-1} (\vec{x} - \vec{\mu})} \quad (9)$$

Another technique to calculate distance is the cosine similarity. It is a measure of similarity between two non-zero vectors. It can be derived from the definition of dot product as follows:

$$\vec{x} \cdot \vec{y} = \|\vec{x}\| \|\vec{y}\| \cos(\theta) \implies \cos(\theta) = \frac{\vec{x} \cdot \vec{y}}{\|\vec{x}\| \|\vec{y}\|} \quad (10)$$

From the previous formula it is clear that the cosine similarity measures the angle between the two vectors.

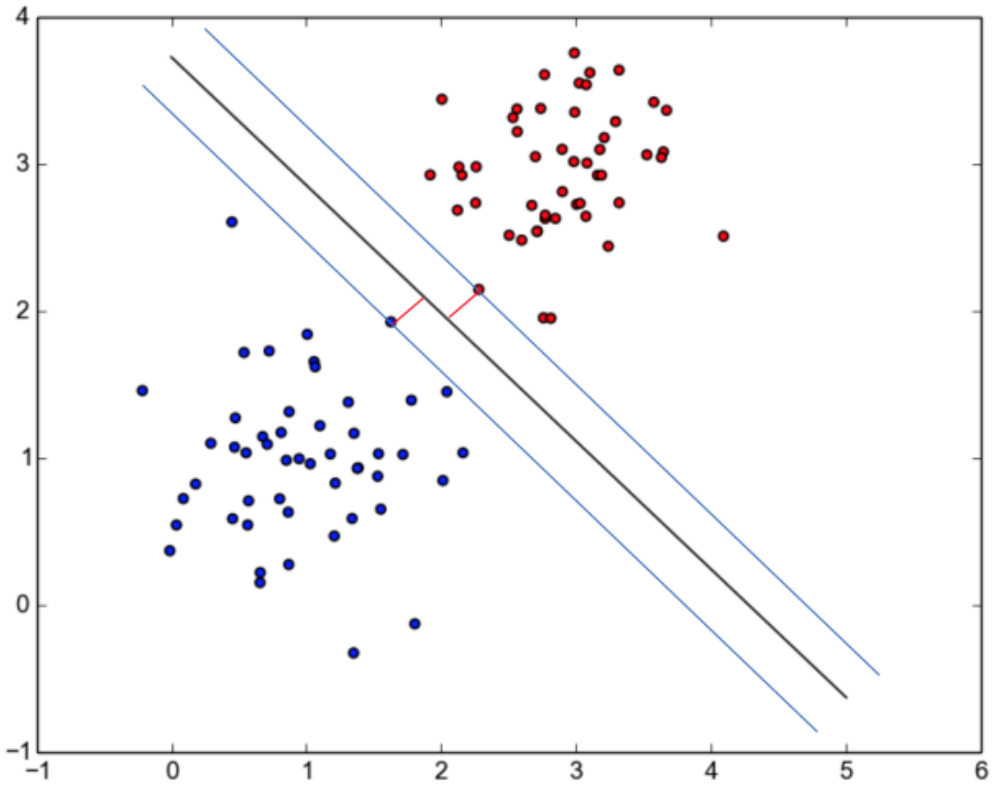
1.1.5 .Support Vector Machines

Support Vector Machine (SVM) is a classification technique employed in a supervised machine learning algorithm. In this algorithm each data item is represented in a n-dimensional space depending on number of features where each feature value is equal to the value on that coordinate. Suppose we had only two features as weight and height. This would be represented in two dimensional space.

Now these two coordinate is known as the Support vectors as in Fig1.4. Now a line is identified that splits the data between these two group. For this a margin is chosen such that it is fathest to the ither point ans closest to the group points. This line is known as the classifier.

Support vector machine is a generalization of a classifier called the maximal margin classifier [8]. This simple version of SVM, that requires classes be separable by a linear boundary, helps to understand how this complex algorithm works.

X2- plane



X1- plane

Figure 1.4 - SVM marginal classifiers

1.2. Research Objective

The problem of sentiment analysis is quite popular in recent times due to the explosion of data online . This analysis is very severe as the data reaches to large population and it has a direct impact in terms of sensitivity or marketing. In the area of sensitivity various researchers is being done on bullying or non-bullying.

This project deals with the marketing field of analysis. In this the data is of utmost important . Here in this we are working on the opinion extraction of the movie review. Researchers have explored this part using various techniques to help in determining the problem. Here taking the base work as done previously that is collecting data and bipolar identification of the movie review is done. In this new ensembling approach along with the concept of object pickling is integrated. Also for the data visualization graph is plotted on each test review using Matplotlib library. For instance, if I say “The movie was OK but not that awesome” . What do you think is the sentiment of this sentence. It looks neutral as it was expected to be

awesome but it was just fine. Now try this “The movie was OK”. Now the meaning of the sentence is quite different as now we don’t know what was author expecting of the movie.

Chapter 2

LITERATURE SURVEY

Sentiment analysis or Opinion extraction can be approached in three ways. They are categorised into knowledge based techniques, hybrid techniques, and statistical techniques methods.

Knowledge-based programming can split information by categories of effects oriented on the current of cleared phrases like happy, depressed, scared and exhausted. Not only do few source bases list apparent affect words, but they also attribute a likely "affinity" to specific feelings to arbitrary words. Statistical methods rely on the components of machine learning, such as latent-based semantic assessment, supporting vector machines (SVM), word bag (BoW) and Semantic Point guidance [7].

More conventional method that attempts to identify the holder of a feeling (i.e., the individual that retains that affective state) and the target (i.e., the entity that the effect is thought about). The grammatical connections of words are used for the context of opinion mining and get the function the speaker has opined about. Grammatical reasons for interdependence are acquired through profound text parsing. In some respects, such as by analyzing content that does not explicitly deploy associated information but is implicitly connected to other ideas, hybrid methods leverage both machine learning and knowledge representation aspects such as ontologies and semantic networks.

There are numerous free source software instruments that deploy statistics, machine learning, and natural language processing (NLP) methods to process large-scale information set opinion assessment, including internet articles, web pages, online discussion groups, online reviews, web blogs, and social media. Search-based instrument can assist identify what market consumers really need. Therefore, the larger the information being processed provides the better outcome to satisfy the client. In another type, domain-based organization uses publicly accessible data sets to obtain the semantic and affective data connected with ideas of natural language. Analysis of opinions

In this chapter various research paper that have been studied related to the subject of concern has been briefed. The topic of sentiment analysis or opinion extraction has been of major concern lately and its very important to understand the core of it. In this thesis movie reviews have been considered and so many previous related papers on this topic have been analysed.

Various paper have already implemented the algorithm to detect the efficient accuracy on the matter, In this thesis the already paper have been extended to provide an ensemble technique using pickling concepts.

With the use of pickling of the object the work load can be reduced by 80% of the original load time. This will enhance the overall time complexity of the system. Only the first time while creating the object time will be large, this will depend upon how large datasets will be. Taking large datasets is always beneficial as it will improve the accuracy of the system. However, its important that datasets taken should be refined and appropriate for the content analysis. The area of mining and data analysis of utmost importance when NLP is done. This forms the basis for the work required and various tools are used to get the work done as per the required objective.

In this the amalgamation of various paper that has been studied is done. Below most of the important read has been detailed. This thesis is derived from this papers core concepts only.With the easy availability of data social media is highly popular among researchers. Many work have been proposed earlier to analyse the trend on these platforms related to text classification. In the work given by Tun Thura Thet, Jin-Cheon Na and Christopher S.G. Khoo a linguistic approach of determining the sentiment of a clause from the prior sentiment scores assigned to individual words by taking into consideration the grammatical dependency structure of the clause is done. This work is useful in multiple review sentiment analysis. Unlike many other work beside computation of the bipolar detection it also calculates the sentiment strength. Like it determines the sentence as the most positive or the most negative. The future scope of this work can be expanded to many online forums such as critic review, twitter analysis etc. [9].

A Classifier ensemble was designed using Naive Bayes and Genetic Algorithm (GA) in the paper given by M.Govindarajan which is based on coupling algorithm . In this approach new hybrid algorithm is designed to evaluate the performane of the movie review. Here the optimum usage of the classifier delivers the best performance. The hybrid of Genetic algorithm and Naive Bayes gives the higher percentage of accuracy. The result obtained from the experiment is as, The Genetic algorithm approach gives better performance with respect to accuracy. This paper also provide an working module on how the hybrid algorithm gives better results than the individual algorithm [10].

In one of the papers given by Grégoire Mesnil, Tomas Mikolov, Marc'Aurelio Ranzato and Yoshua Bengio and ensemble of generative and discriminative techniques have been used for sentiment analysis of movie reviews using RNN and n-gram model . This paper also work on giving an impactful ensembling algorithm. In this approach three segments of the reviews are combined, they are ,first is based on the language model, second is based on the continuous representation of sentences and the last one is based on the weights defined by tf-idf bag of words representation of the document. Every single module helps in the contribution of the analysis of the movie review. This results in the state of the art picture of the combine algorithm [11].

A standard performance evaluation metrics of Accuracy, F-measure and Entropy of movie review is determined using popular Naïve Bayes and SVM with the SentiWordNet approach in the paper given by V.K. Singh, R. Piryani, A. Uddin and P. Waila. In this paper it has been obeserved that by using SentiWordNet the classification of movie review is not much effective along with Naive Bayes and SVM. Due to the reviews having the same pattern of expression the Naive bayes and SVM gives a good performance but it can't be said same when the variety of review is considered. Therefore to acheive the highest level of performanve the SentiWordNet adjective scheme is utilised. This gives the best result out of all the four schemes. Adjectives forms the major structure behind the meaning of the reviews. Along with the adjective when combined with the Adverb it becomes much better in terms of performance evaluation. It has been observed that it gives the better performance result in the blog analysis too. [12].

Taking unigrams and n-grams as the feature vector with Support Vector Machines and Naive Bayes approach an accuracy range between 63% - 82% has been found in the paper given by Michelle Annett and Grzegorz Kondrak. The experiment has been succesful in getting the desired result. The basis of this algorithm was derived on how the reults would vary on the features chose. The succseful completion of the analysis also gives the feature is the most significant in sentiment analysis or be it any machine learnig task.

This experiment extends for the blog review for not only on detection of the biploar nature but also in the different categories of rating from one to five. It also extends in taking the user's preferences and their attitude towards movie selection. Prefernce of choice of actors and actressses , the genre of the movie that is comedy, romance, thriller or horror. In the nutshell to understand the users thought is one of the challaneges in the discovery of more appropriate review analysis [13].

In Mahesh Joshi, Dipanjan Das, Kevin Gimpel and Noah A. Smith paper , linear regression from text and metadata is described for a new dataset pairing movie reviews and show that review text can substitute for meta-data, and even improve over it, for prediction . The result obtained in this experiment gives a comparable result is simlilar to that obtained from the actual analysis. It provides the result that the study of both meta data and text features provide more significant analysis of the review. In the testing of the model, it has more of negative reviews but the overall distribution gets balanced. Here the text is able to substitute for the meta data. Lexical n- grams proved to be a strong baseline to beat other approach [14].

In the paper given by V.K. Singh, R. Piryani, A. Uddin and P. Waila presents a new kind of domain specific feature-based heuristic for aspect-level sentiment analysis of movie reviews. In this SentiWordNet scheme is used to compute the document-level sentiment for each movie reviewed and compared the results with results obtained using Alchemy API. It works on the combination of Adverb+Verb with the Adverb+Adjective in the one part and in another it works on the feature-based heuristic technique for the aspect-level sentiment analysis. First part of the work is based on the opinionated value of the sentiment analysis of the movie review along with its linguistic feature . The next part of aspect-level sentiment analysis provides a quick and simple to implement approach. This part is also useful in the determination of the movie recommendation. One of constraint wth the aspect-level is that it only limited to the domain specific [15].

A Naive Bayes, Decision Trees, Maximum-Entropy, and K-Means clustering algorithm is applied on simple bag-of-words model in Kuat Yessenov and Sasa Misailovic paperon Sentiment Analysis of Movie Review Comments. In this text for analysis have been taken from the Digg as the text copora. The reults obtained gives the analysis on how simple bag of words can perform relatively better, it can firther be refined by the choice of features of semnatic and syntactic information of the text. The future work also expands on the use of off-topic and on topic and the usage of WordNet [16].

In another paper given by Alistair Kennedy And Diana Inkpen classifies reviews based on the General Inquirer to identify positive and negative terms, as well as negation terms, intensifiers, and diminishers.In the second method uses a Machine Learning algorithm, Support Vector Machines. This experiment on the movie review with the integration of the valance shifters, it has been found out that the improving effect on the classification is better. In another approach to the reserach postive and negative terms were increased to check on the performance as well as on the intensifiers and the dimishers. In this for the future scope the

weights could be assigned to the system to determine the understatement and the overstatement of the system. Also in future named entities can be used to study on the review anlaysis [17].

In Bruno Ohana and Brendan Tierney paper on Sentiment Classification of Reviews Using SentiWordNet presents the results of applying the SentiWordNet lexical resource to the problem of automatic sentiment classification of film reviews . Result obtained on this is similar to other work that are based on the SentiWordNet opinion lexicons. It has been similar to other analysis of movie review that employs maual lexicon concept. This also explain how the accuracy can be improved in the linguistic area. For the future scope it details on the performance comparison of the other lexicons and SentiWordNet [18].

Lianjing Jin, Wei Gong, Wenlong Fu and Hongbin Wu gives the paper that improves the accuracy of Naive Bayes classification using improved information gain ,one of methods of feature extraction, by reducing the impact of low-frequency words using NLTK corpus. This paper incorporates Naive Bayes algorithm on the analysis of the text review. To improve the classification accuracy has been studied in this paper. The information gain act a measure for the naive bayes classification. The observed results tells the by removing the low frequency word the higher accuracy can be attained which enhances the naive bayes alogorithm. In the future scope work work can be done on the field on space complexity and time complexity. [19].

In another paper given by Y. Wang, W. Fu, A. Sui and Y. Ding It has been found that Naive Bayes classifier has a higher accuracy and rate by classifying Movie Reviews in NLTK using Decision Tree classifier, Naive Bayes classifier, Maximum Entropy classifier and K-nearest neighbor classifier. So in this paper the study of these four algorithm is done on the movie review text datasets. Following results have been obtained as the analysis on the review. Naive Bayes gives the best accuracy out of all the four classification algorithm. Fastest in terms of computation is given by KNN algorithm and Decision tree approach provides the most stable on the algorithm [20].

In Paper given by G. S. Brar and A. Sharma includes sentiment analysis of movie reviews using feature-based opinion mining and supervised machine learning. This paper is the study on twitter data. The tweets that are obtained from here a bit complext to unerstand the emotions behind it. There is presence of from sarcastic comments to meme content [21]. The ambiguity in the content makes it a challenge to understand the sentiment behind it. Here

feature vector space is created for text pre-processing on each tweet. In the best step feature is extracted without any presence of hashtags etc. In this analysis Naive Bayes and SVM have the best accuracy on the tested tweet. Feature vector on the tweets give much better accuracy than on the movie review data [22].

In Mr. B. Narendra, Mr. K. Uday Sai, Mr. G. Rajesh, Mr. K. Hemanth, Mr. M. V. Chaitanya Teja and Mr. K. Deva Kumar paper illustrates a comparative study of sentiment analysis of movie reviews using Naïve Bayes Classifier and Apache Hadoop in order to calculate the performance of the algorithms and show that Map Reduce paradigm of Apache Hadoop performed better than Naïve Bayes Classifier. In this the performance is measured with the result that MapReducer framework gives better result than Naive Bayes for both the stop word removal and bi gram detection [23].

Vuk Batanovic, Bosko Nikolic and Milan Milosavljevic give a dataset balancing algorithm that minimizes the sample selection prejudice by eliminating unnecessary systematic distinctions between the feeling classes. The author has suggested a bias on the choice of resource language in this article. The specifically made algorithm can exceed the random undersampling for the algorithm to obtain balanced datsets. Serbian film review was taken for the research. The work can be extended in the future in other fields, such as product reviews, books, etc. in Serbian language, thus growing the availability of datasets [24].

The best reason behind the classifier ensemble is that it is often much more reliable than the individual classifiers that construct it. The ensemble has a greater degree of overall accuracy that depends on each separate classifier's differences as well as on their separate results. Thus, the determination of genuiune classifier to build an ensemble model continues a fight. In addition, not all algorithms are great at detecting all kinds of issues.

A definite classification method should only be permitted to vote for that output class for which it works well in a voting-based classifier system. Some classifiers, for instance, are great for identifying people's names, while some are great for identifying location names. The choice of suitable votes per classifier is therefore an significant study issue. We first present this classifier ensemble as an optimization issue in this article and provide it with two distinct types of alternatives based on single and multi-objective optimization methods. Here we find out whether or not a specific classifier is permitted to vote for a specific class.

Technical developments have progressively altered how realized tourist data, shared by internet media, have become a strong source of information affecting both the value and

performance of tourism. However, there is a level of information on the Internet that makes manual processing almost impossible and requires fresh analytical techniques. Opinion mining is increasing rapidly in examinations to look at semantian relations and significance as a free manual method.

In this paper, distinct approaches to sentiment analysis implemented in tourism are evaluated and evaluated in terms of the used datasets and performance on main metrics of assessment. The paper concludes by highlighting various future study avenues as part of a wider Big Data strategy to further advance sentiment assessment in tourism.

This article looks at the way customer experiences driven by NLP can be gained. Sketching is a key to innovative design. Developers experiment fast with several abstract ideas with easy, concrete tools such as sketches and prototypes of paper. NLP-driven drawing experiences, however, lead to distinctive issues. For instance, abstract linguistic interaction can be difficult to visualize or a wide variety of technically viable smart functions can be designed. Throw away the issues we experienced and explain emerging alternatives such as a fresh wireframe format through a first person account of our design method for generating smart writing support. [24].

A new method has lately been suggested for weighing circumstances based on numerical statistics. The redundancy of two terms and their individual conditions were separated in two freshly suggested algorithms. The distinct abilities and the correlations between the functionalities have been contrasted with these algorithms. They proposed that the terms of members be eliminated, which would be repeated instead of eliminating two terms that might lose a big number of useful terms. In addition to the discriminatory individual characteristics, two requirements for creating composite elements were also integrated. Moreover, an assessment criterion was considered based on the class distribution of terms [25].

It gives helpful insights into how products can be enhanced or sold more efficiently following such a weblog debate. An significant part of this assessment is to determine the impression placed in blogs on certain brands and products. The finding of client information on the Internet has resulted to a distinct attempt and a major battle for businesses that are increasingly concerned to comply with their product issues.

Opinion extraction focus on the task of determining if a part of textutal data has a potivite opinion or negative opinion based on that particular subject. Various reserach earlier in this field have shown sentiment division on the basis of meaningful data. In recent techniques there have been some study where on the value of labelled data of training set the sentiment

analysis is done. In this paper a meanigful information from the backgroud is obtained for the clssification of information and it further extends the usage of available resource.

Results based on testing or experience on different subjects show that our technique to performs better than using background knowledge or training data in isolation, as well as alternative technique to using lexical knowledge with text classification.

CHAPTER 3
PROPOSED METHODOLOGY

One of the biggest unstructured data available is online information. This is a big repository of data provided that the correct content can be extracted from it. One way to do this is by analyzing feelings. Analysis of sentiment has appeared as an significant element in the analysis of text. Feeling is an attitude, thought or judgement that is prompted by an assessment of feelings and feelings, also known as opinion mining, that studies the feelings of people towards certain entities. Compared to more traditional techniques of market studies (e.g. surveys or opinion polls), sentiment analysis has the benefit of being more cost-effective and time-efficient. It is also a non-intrusive technique of extracting views from customers

Opinion extraction was carried out on multiple domains such as social media and shopping reviews to gage client feelings. However, there is still a lack of a extensive research of client feelings in the hotel domain. Through ready-made online software, most analyzes have concentrated on evaluation. So far, no algorithm inherently produced has been used. Our research is aimed at addressing these issues.

3.1 Ensemble Techniques

These are also known as multiple classifier systems in learning module. They are advanced and highly popular technique in the field of machine learning. They are quite helpful in the classification problems [25]. The main reason or the idea behind this technique is that there no perfect algorithm to solve a problem. Ever algorithm and its model has its own weaknesses and limitations. Hence the goal is to obatin the best possible approach to solve the problems with the maximum outcome from it.

Combination of algorithm in the recent times has come into the picture that it provides much accurate guessing than the simple algorithm on the same problems.With the combination of algorithm the negative impact of some algorithm can be balanced and the distribution of accuracy is homogenous. The diagram below in Fig 3.1 gives the visual representation of the ensembling technique on the differnt models.

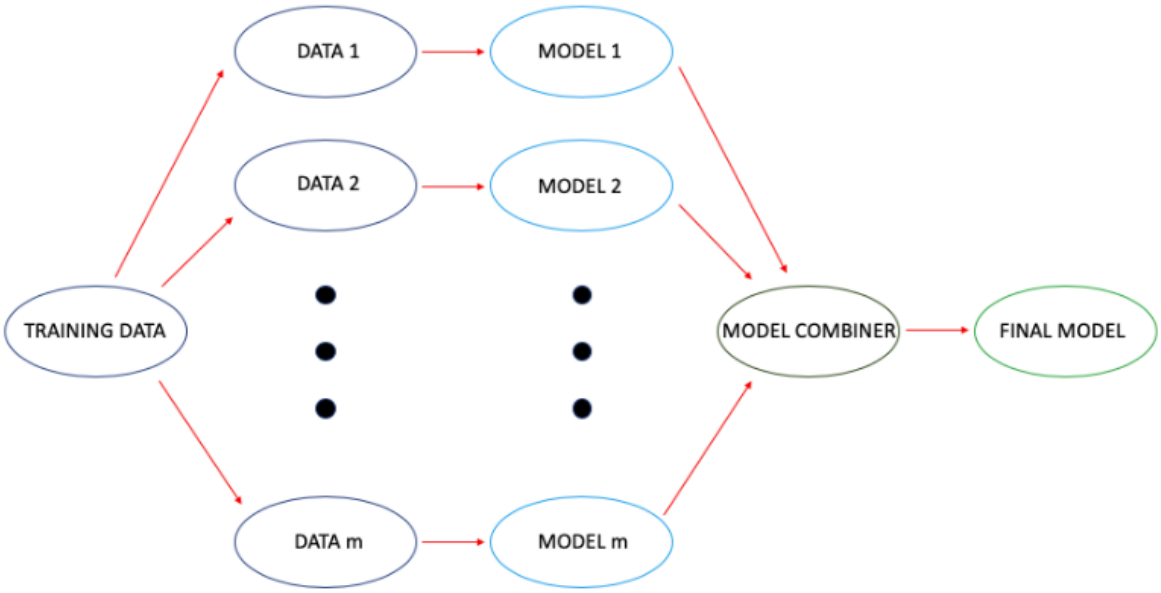


Figure 3.1 - Visual representation of the ensembling technique on the differnt models.

This chapter provides an introduction to ensemble methods, with particular focus on some well-known algorithms in the context of classification problems: voting methods, bagging, boosting and stacking.

Ensemble method on different algorithm can be divided into two categories:

First is the Homogeneous ensembles. They are alos know as the weak learners as they are combination of n base learners, this is generated by base learning algorithms of the same kind, like SVM, decision tree,etc. Random Forest is one of the most representative methods is that explains the concept, also as the name suggests this is composed by a huge number of decision trees.

And the second is the Heterogeneous ensembles. To make a distinction from the previous method different learners are used, so that the base learners are indicated as individual ones, in order.

The model combiner can be obtained in various ways and it can be analyzed later. A common architecture of homogeneous ensembles is shown in figure 5, in which m models are trained on a portion of the training data and then combined together giving the final model.

3.1.1 Voting Method

Voting is one of the simplest technique used for ensembling. Voting methods are build upon a several estimators. It predicts the average and also can be used in the regression. Voting is mainly used for classification techniques. With respect to the single base estimators, thhe ensemble one has usually has a improved performance . Consider the given scenario with a set of T individual classifiers *h1 , . . . , hT* and K classes, each one labelled as c k with k ∈ [1, . . . , K] [4]. For a classifier *ht t ∈ [1, . . . , T]* the outputs associated to an instance x are a K-dimensional k vector (h 1 t , . . . , h K t). h t has two formulations:

First is Class label: h kt ∈ {0, 1} that takes value 1 if the class is of k and 0 otherwise.

Second is Class probability: h kt ∈ [0, 1]. It can be seen as an estimate of the posterior probability P (c k |x) for classifier h t .

There are several ways to combine prediction. First of all, the mode can be used to obtain the value that occurs most often for each classification. This type of voting scheme is called as plurality voting:

$$H(x)=c_{argmax} \sum^{T}_{i=1} h_i^j (x) \qquad (11)$$

In this H(x) is the outcome of the ensemble classifier. Another one is majority voting, where at least half of classifiers choose the similar class. In case of any inspection does not reach this value of 50% percentage then it cannot be classified. This will be a reject option and is clear that in plurality voting it does not exist.

$$H(X)= \begin{cases} c_j & if \ \sum^{T}_{i=1} h^j_i (x) > 1/2 \neq T \\ rejection & otherwise \end{cases} \qquad (12)$$

Here #T determines the counts of models that has used. To understand it more clearly lets take an example, consider a three-classification problem with the class y ∈ {0, 1, 2}. Using this classifiers technique, the prediction classes for the first observation are 1, 0, 0 and 2. It is clear that plurality chooses class 0 as best prediction, while majority voting gives a reject option because class 0 reaches 50%, without exceeding it. Another possibility is to simply use the mean of predictions. It can be done via class label or probability representation. As was said before, in the first case when a model classified an instance in a class j the resulting vector have all entries equal to zero and one in position j. For example, it can be (0, 0, 1, 0)

for third class in a four-classification problem. Suppose to have other two classifiers that give (0, 1, 0, 0) and (0, 0, 1, 0) as class label vectors.

Evaluate the mean implies to sum the elements in the same position and divide by the number of classifiers, giving (0, 1/3, 2/3, 0). For each class it is defined as:

$$H^j(x) = 1/T \sum^T_{i=1} h^j_{i(x)} \quad (13)$$

Now, choosing the higher value in the vector it is clear that the predicted class by the ensemble classifier is the third [26]. An example with the prediction probabilities applied to a three-classification problem is reported. For an observation the vectors of class probabilities are the following:

The first classifier will be [0.34, 0.37, 0.29]

The second classifier will be [0, 0.4, 0.6]

The third classifier will be [0.35, 0.08, 0.55]

Applying the above formula 13 the following results can be the classifier evaluates the mean in each class and the result is [0.4467 0.14 0.4133]. Now, the predicted class is chosen as the higher value in the vector. So, the observation is associated to the first class. The last case of voting methods evaluates a weighted mean, giving more importance to the most accurate classifiers:

$$H^j(x) = \sum^T_{i=1} w_i h^j_{i(x)} \quad (14)$$

Here the total sum of weights is one. In both cases of weighted mean and mean the output H(x) of the ensemble classifier is obtained as the maximum value in the vector composed by H j (x), it is defined as:

$$H(x) = c_{argmax} H^j(x) \quad (15)$$

Therefore, the last ensemble can be the best among voting methods with an accurate selection of weights,

3.1.2 Bagging

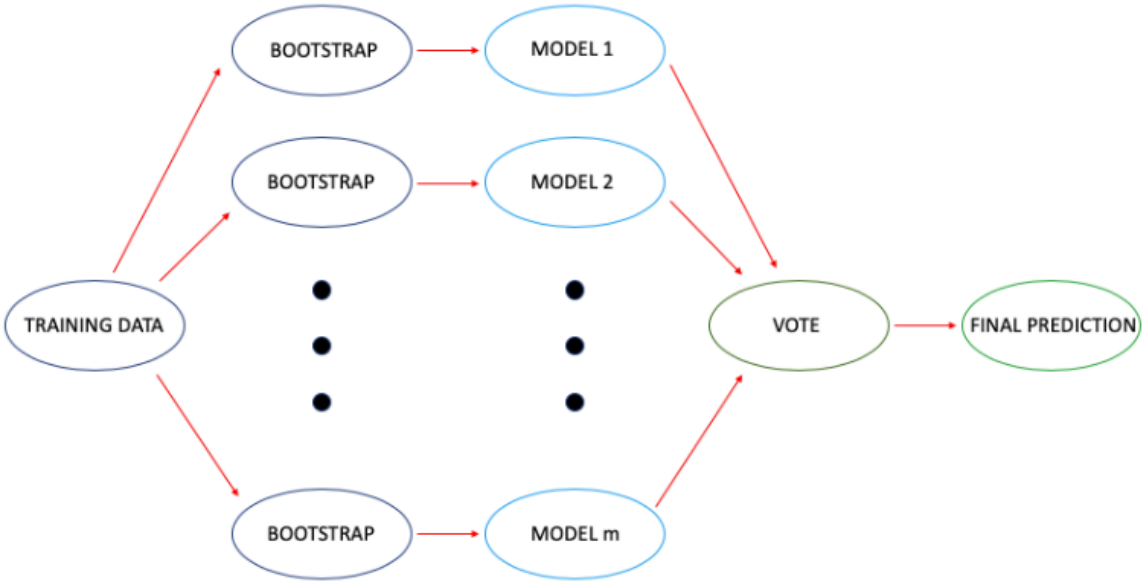


Figure 3.2 - Bootstrap Bagging ensembling model.

Bootstrap Aggregrating is another name for Bagging. The concept of bootstrap is introduced. It is a type of resampling with replacement, in which smaller random samples of the same size (bootstrap samples) are obtained by the original data set. No matter whether if its a classification problem or regression, this takes the input datasets and output with respect to defined training datasets. It is particularly important and frequently used in the matter of decision trees, due to its high variance [27]. Indeed, it is a procedure used in order to decrease the variance of a machine learning algorithm.

For example, the predictions for an average of trees have lower variance than the variance of the individual ones. In bagging the classifiers are built on bootstrap samples of the training set. Finally their results are combined by a plurality vote. Its general idea is shown in figure 3.2. From a data set of N observations, M bootstrap samples are constructed, each one with N observations. Since, it happens with replacement each observation has a probability (p = N) of being selected and (p ∗ = 1 − N) of not being selected. So, the probability of not being selected n times is

3.1.3 Boosting

Boosting ensembling methods proceed in the same spirit as the bagging methods. In this a class of designs are aggregated to obtain a strong learner that performs better. This is a smart

approach, which refers to a family of algorithms able to convert weak learners to stronger ones.

In this procedure the models are grown sequentially: each one is grown using information the most popular boosting algorithm due to Freund and Schapire is “AdaBoost.M1.” In order to understand how this algorithm works an example is proposed. It is a two-class classification problem with N observations and the output class Y ∈ {−1, 1}. A weak classifier M (X), one whose error rate is slightly better than random guessing, takes a vector of predictor variables X and produces a prediction in one of the two classes. Random guessing refers to a method in which the events involved have an equal chance of being chosen.

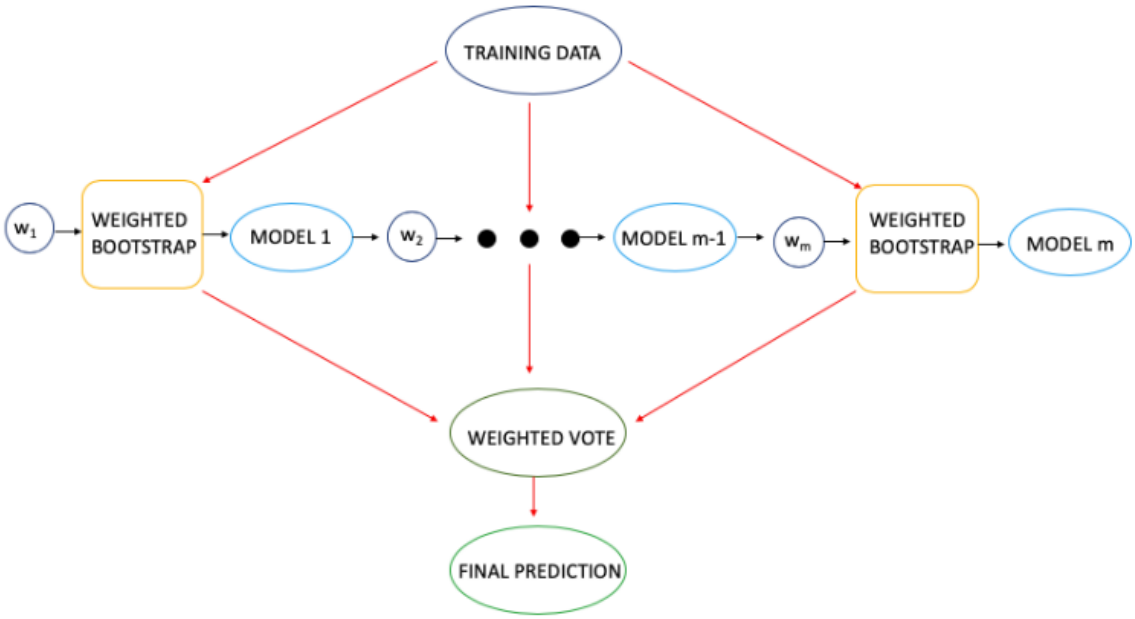


Figure 3.3- Bootstrap Boosting ensembling model.

Having one weak classification algorithm, it is sequentially applied to repeatedly modified versions of the training data, producing a sequence of M weak classifiers. At the first step, classifier M1 is trained in the usual way. Some weights are initialized giving the same importance to every observation (wi = N1). From the second one, until the end, the training data set is modified applying some weights w 1 . . . w n to each observation. Weak classifiers are trained to the data set and the one with the lowest classification error e m is selected. Then, the weight α m for the M m is introduced as

$$\alpha_{m} = 1/2 \log (1-e_m / e_{m)} \quad (16)$$

For a classifier with less than 50% accuracy, where em is greater than 0.5, the weight αm is negative and vice versa. So, it is possible to combine the prediction flipping the sign. It should be noted that, sometimes in literature, the term 12 is omitted in formula 3.12. However, it does not change the logic of

$$w_{m+1}(x_i, y_i) = w_m(x_i, y_i) \exp(-\alpha_m y_i M_m(x_i)) / Z_m \quad (17)$$

where Z m is a normalization constant that guarantees the sum of all instance weights is equal to 1. For example, imagine that a positive weighted classifier misclassifies the observation in exam, the "exp" term in the numerator would be always larger than 1, giving more weight to this misclassified observation in the next iteration. This happens because *yi Mm (xi)* is -1, α m is positive and so also $\alpha_m M_m (X)$ is positive. Otherwise, weights associated to correct predictions are decreased due to the negative term $-\alpha_m M_m (X)$ that, applying the exponential function, gives a number ∈ (0, 1). Summing up, the relative influence of misclassified observations is increased, inducing the next classifier in the sequence to predict better. Then the prediction of each algorithm is combined with the others through a weighted majority vote. The final prediction obtained is

$$F(X) = sign \left(\sum_{m=1}^{M} \alpha_m M_m(X) \right) \quad (18)$$

where α values give more weight to the more accurate classifiers.

3.1.4 Stacking

Stacking is another class of ensemble method that can achieve excellent results in classification. This is different from Boosting and Bagging technique as it deals with the hetrogenous weak learners. Secondly, stacking learns to combine the base models using a meta-model whereas bagging and boosting combine weak learners following deterministic algorithms. A learner, called meta-learner, is trained to combine the individual ones of the previous level.

It is a general procedure composed by different learners (algorithms) and levels [28]. This procedure can be applied over and over again. In figure 3.4 a general procedure of two levels

is shown. The name comes from the fact that the final model is said to be stacked on the top of the others.

Input:

Data set $D = \{(x_1, y_1), (x_2, y_2), \dots, (x_M, y_M)\}$

First-level learning algorithms: $A_1, \dots, A_T$

Second-level learning algorithm: A^*

Figure 3.4 - Input Datasets for two level stacking procedure.

It should be noted that in any ensemble method there is not guarantee that the model created will have better performance than all the algorithms used.

Natural Language Processing (NLP) techniques are many times included with the lexicon based approach to find the meaningful anatomy with the help of finding the semantic relations.

In the above algorithm of input datasets of two level stacking procedure a system is proposed that is capable of accessing the opinions from the user. This results are quite promising in colloquial language predominates.

Below the process of ensemble stacking is explained in the figure 3.5 where the algorithm is combined for the generation of stack based structure.

Process:

1. for t = 1 : T:
2. $h_t = A_t(D)$;
3. end;
4. D' = ∅;
5. for i =1 : M:
6. for j = 1 : T:
7. $z_{i,t} = h_t(x_i)$
8. end;
9. $D' = D' \cup \{ (z_{i,1}, \ldots, z_{i,T}), y_i \}$;
10. end;
11. h' = A*(D');

Figure 3.5 - The process of combining algorithm.

This chapter provides an introduction to ensemble methods, with particular focus on some well-known algorithms in the context of classification problems: voting methods, bagging, boosting and stacking.

Output:

$$H(x) = h'(h_1(x), \ldots, h_T(x))$$

Figure 3.6 - The output procedure of ensemble.

Now with the availability of a new data sets, its time to train them. For the execution of this code postive and negative reviews of statement have been collected. In this some roughly 5500+ postive and 5500+ negative reviews of movie from the IMDB that are further grained have been collected. These review are much accuratefo testing and training purposes.

In the next step, document databse and featuresets are build up from them. Variables are defined as per the need and data is collected and apprpriate function is applied on them.

Table 3.1 - Pseudo Code for converting text into features:

Step 1- Read all the postive and negative movie reviews in a seperate file such pos_review and neg_review from the location.
Step 2- Create a database file of all the reviews into one file with their respective category.
Step 3-Word tokenize the postive and negative reviews from their respective file i.e pos_review and neg_review and apply the part of sepech tag to it. In this adjective, adverb and verb is checked.
Step 4- In the word tokenized, apply frequency distribution method and extract top 5000 words from it. These words would be used to create featuresets. Lets call it as word_feature.
Step 5- This is the most important step in which words is converted into features. Each review is passed through a function where first the review is tokenized. Next foreach word_feature from step 4, it is checked for its presence in the given tokenized review. Here the boolean result is stored for each word. After the completion of loop feature is returned.
Step 6- Repeat Step 5 for each testing review and generate a complete feature set .
Step 7- Shuffle the datasets obtained from step 6.
Step 8- From the feature sets take top 10000 as training set and bottom 10000 as testing set.

Reading of the postive and negative reviews into document can be programmed as below.

Table 3.2. Postive and Negative reviews read.

```
short_positive = open("reviews/positive.txt","r").read()
short_negative = open("reviews/negative.txt","r").read()
documents = []
```

```
for rev in short_positive.split('\n'):
  documents.append( (rev, "pos") )
```

```
for rev in short_negative.split('\n'):
   documents.append( (rev, "neg") )
```

Setting up of data for the feature is done with the extraction of most important words for the classification and this can be epresented as below:

Table 3.3 – Word Tokenization Model

all_words = [] **pos_words = word_tokenize(short_positive)** **neg_words = word_tokenize(short_negatuve)**
for w in short_positive_words: **total_words.append(w.lower())**
for w in short_negative_words: **total_words.append(w.lower())**
total_words = nltk.FreqDist(total_words) **word_features = list(all_words.keys())[:5000]**

Next, need to adjust our feature finding function, mainly tokenizing by word in the document, since we didn't have a nifty .words() feature for our new sample, went ahead and increased the most common words:

Table 3.4 – Featureset Model

```
def find_features(document):
    words = word_tokenize(document)
    features = {}
    for w in word_features:
        features[w] = (w in words)
    return features
```

```
featuresets = [(find_features(rev), category) for (rev, category) in documents]
random.shuffle(featuresets)
```

This process will take 30 to 40 minutes to complete its execution the first time. That too on being run on the core i7 processor. However, once the pickling is incorporated this task can be reduced by more than 70% of the execution time.

CHAPTER 4
IMPLEMENTATION AND RESULTS

4.1 Setup of NLTK Toolkit

The NLTK module is an ideal opinion mining toolkit. The whole natural language processing methodology (NLP) is promoted by this instrument. NLTK will help avoid the feeling assessment aspect through separation of phrases from paragraphs, separation of phrases and tokens, identification of the language marks, and emphasis on primary topics [29].

In order to get started with the NLTK module Python needs to be installed in the system. To get the latest version of Python one can get it from Python.org and download the latest version of Python for Windows, for Mac or Linux following command needs to be executed.

apt-get install python(version number)

Next step is to get NLTK. One of the easiest method to install the NLTK module is by using pip command. On the command prompt type pip install nltk.
After the setup of NLTK, we need to install some of the components for NLTK. In a python file type this code and run.

```
import nltk
nltk.download()
```

After the execution of the code a GUI will pop up like this with red strips, which means it is yet to downloaded. After the installation it will turn into green:

This figure 4.1 given below is the popup window for the installation of the NLTK toolkit libraries. In this all the packages are listed. Here all the packages are installed but the user can select only the required package as per the need of the project.

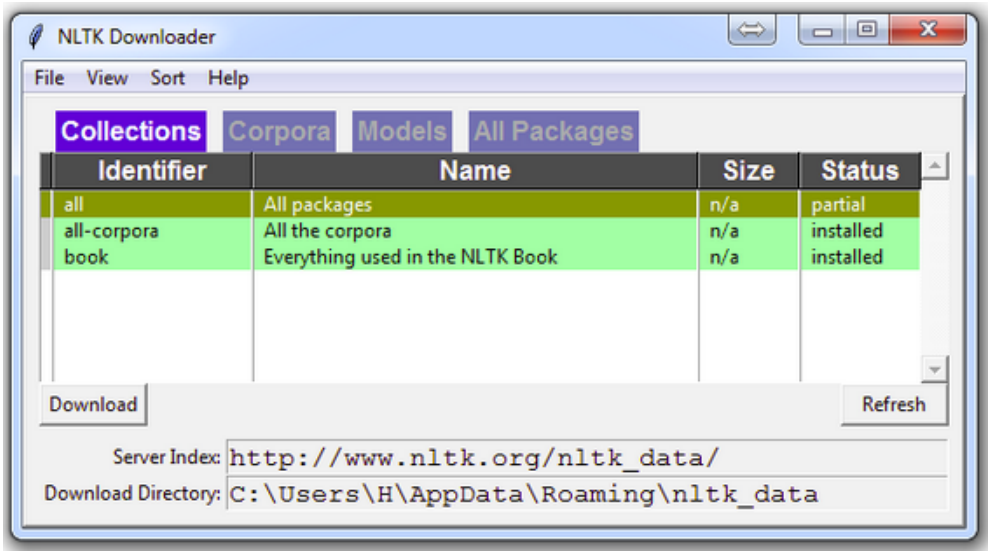


Figure 4.1 - NLTK downloader windoe popup.

When this window appears, select "All" to download all packages and click "Download." All tokenizers, chunkers, algorithms and the entire corporation are downloadable. It can also be selected manually to avoid some space overloading instead of being downloaded all at once. The NLTK module takes about 8 MB and the entire database folder, including chunkers, parsers and the corpora, will take about 2 GB.

NLTK module can bedownloaded without the pop window from the code itself. Following code can be executed. This will download everything without any related dependecy.

```
import nltk
nltk.download()
```

d (for download)

all (for download everything)

After installation of the required libraries, to get started with NKTK toolkit few vocabularies need to be known . They are:

Corpus – The nltk.corpus package of NLTK provides this. This is a body of text with diverse content in it . Corpus is the singular and Corpora is the plural of this. Example: A collection of indian revolution journals.

Lexicon – This is the structure and meanings of phrases. English dictionary, for instance. However, consider that there will be distinct lexicons in distinct areas. For instance: The

significance for the term Bull to a financial employee is the one that determines the market compared to the prevalent English phrase in which the first meaning for the term Bull is an animal. As such, economic investors, physicians, kids, mechanics, and so on have a unique lexicon.

Token – This describes the "entity" that is component of the rule-based split-up. For example, if a sentence is "tokenized" into words, each word is a token. Each sentence, if you tokenized the phrases from a paragraph, can also be a token.

These are the words that you will most frequently hear when you enter the room of Natural Language Processing (NLP), but we will cover many more in time. Let's demonstrate an instance of how tokenize something with the NLTK module into tokens.

4.2 Experiment Setup

After the initial setup of NLTK, the problem of this thesis is worked upon. In the previous chapter of Proposed Methodology the problem of opinion extraction on movie revie has been explained. Program is coded in Python, for any data analysis task this language provides huge support. NLTK is super easy to understand as it provide huge framework support for many libraries.

Table 4.1 - Construction of Ensemble classifier

```
Class Vote_Classifier(ClassifierIm):
def __init__(self, *classifiers):
    self._classifiers = classifiers
def classify(self, features):
    vote = []
    for s in self._classifiers:
        t = s.classify(features)
        vote.append(t)
    return mode(votes)
def confidence(self, features):
    vote = []
    for s in self._classifiers:
        t = s.classify(features)
        votes.append(t)
choice_votes = votes.count (mode(votes))
    conf = choice_votes / len(votes)
    return conf
```

After the complete setup of the problem solving approach the final call is made. This is done by integrating the earlier file of text feature and the vote classifer into the new python file.

For example, if the file name in which all the previous work is done is project.py then it can be called from a different file as:

import project as s

With the above call the previous gets imported in the new one. Now the output needs to be read into the new file as:

output = open("review.txt","a")

Rest all the calls for testing purpose can be made as below:

Table 4.2 – Output Call for the ensemble

sentiment_value, confidence = s.sentiment("movie was ok") **output.write(sentiment_value)** **output.write ('\n')**
sentiment_value, confidence = s.sentiment("it was a bad experience") **output.write(sentiment_value)** **output.write ('\n')**
sentiment_value, confidence = s.sentiment(("actors were amazing")) **output.write(sentiment_value)** **output.write ('\n')**
sentiment_value_, confidence = s.sentiment("movie need to work on script.") **output.write (sentiment_value_)** **output.write ('\n')**
sentiment_value_, confidence = s.sentiment("awesome songs and screenplay.") **output.write (sentiment_value_)** **output.write ('\n')**

```
sentiment_value_, confidence = s.sentiment("Nothing was good in this movie")
output.write (sentiment_value_)
output.write ('\n')
```

```
sentiment_value_, confidence = s.sentiment("It was an outstanding piece of work")
output.write (sentiment_value_)
output.write ('\n')
```

```
sentiment_value_, confidence = s.sentiment("Loved everything about it")
output.write (sentiment_value_)
output.write ('\n')
```

```
sentiment_value, confidence = s.sentiment(("actors were amazing"))
output.write(sentiment_value)
output.write ('\n')
```

```
sentiment_value_, confidence = s.sentiment("movie need to work on script.")
output.write (sentiment_value_)
output.write ('\n')
```

```
sentiment_value_, confidence = s.sentiment("awesome songs and screenplay.")
output.write (sentiment_value_)
output.write ('\n')
```

```
sentiment_value_, confidence = s.sentiment("Great work by all the actors")
output.write (sentiment_value_)
output.write ('\n')
```

```
sentiment_value_, confidence = s.sentiment("Poor editing and story")
output.write (sentiment_value_)
output.write ('\n')
```

After callimg on each test case the output file need to be closed as:

output.close()

4.3 Observed Results

In the given observation for the given input run on the different algorithm combined together, following result is obtained. Without pickling the same code is completed in aroud 25-30 minutes but now with the involvement of pickling the same result result can be determind in the 4-5 minutes as the previously created obkect is loaded in the system.

With respect to each review positve or negative review is obtained, However, it can be fighured out that the for few reviews as such “Great work by all the actors”, “Loved everything about it” the results obtained are negative which is incorrect. In the following 10 obesrvation accuracy is about 80%.

Table 4.3 Observation Matrix

Input Data	Output Data
Movie was okay	Positive
It was a Bad experience	Negative
actors were amazing	Positive
movie need to work on script.	Negative
awesome songs and screenplay.	Positive
Great work by all the actors	Negative
Poor editing and story	Negative
Nothing was good in this movie	Negative
It was an outstanding piece of work	Positive
Loved everything about it	Negative

The above output can be further displayed with the Matplotlib libraries. These are the python framework that helps in providing graphical representation of the data. For this Matlpoltlib needs to be installed and imported in the code. This can be executed as below:

Table 4.4 - Matploltlib code structure.

import matplotlib.pyplot as pltt **import matplotlib.animation as anime** **from matplotlib import style**
import style such as style.use(“ggplot”) **# remaining code**
pltt.show() **# To display the graph in the chose style**

This is the graphical representation of the result obtained. This is based on the style “ggplot” which is a part of Matpotlib library. In this graph the doward slope indicates the negative review and upward slope one is the positive. The y-axis represnts the value of reviews, however the x-axis gives the no of movie reviews. The final value of on the graph indicates the overall impact of the movie review.

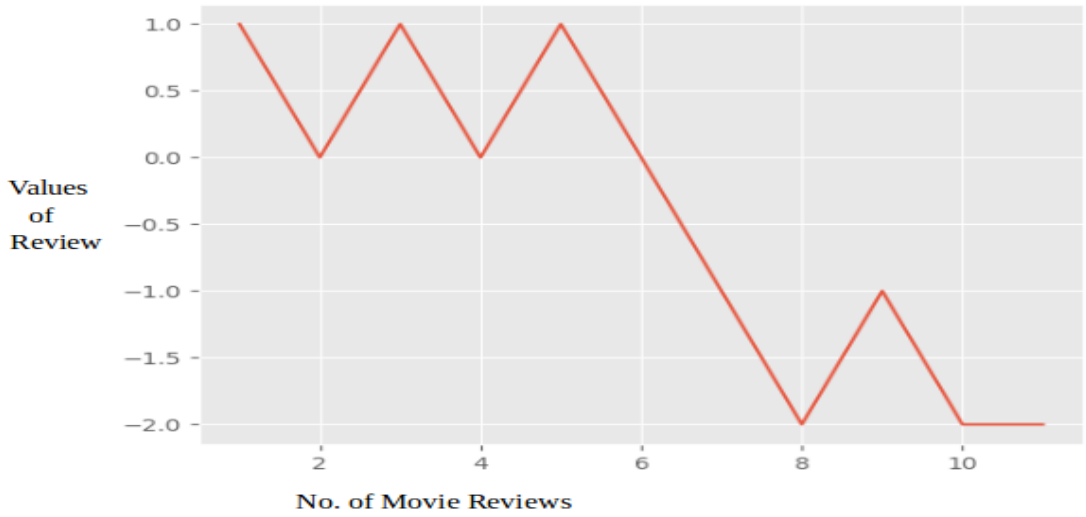


Figure 4.2 - Matplotlib graph representation of the ensemble result.

The graph analysis of the above code for the given set of tested input gives the movie on the negative side. The total value that is obtained after the summation of the y-axis value with the respect of each review gives the result. Each review is plotted at the x-axis.

CHAPTER 5

CONCLUSION AND FUTURE WORK

In this chapter, we first briefly summarize the main work in the thesis. And then gather the findings and make some comments on them. At last, we suggest possible future work in order to better tackle the problem.

5.1 Conclusion

Social media and the Internet have opened up new types of empowerment as well as oppression. Meaningful engagement has transformed into a detrimental avenue where individuals are often vulnerable targets to online ridiculing. Predictive models to detect this cyberbullying in online content is imperative and this research proffered a prototype model for the same. The uniqueness of the proposed enselmble techniquye is that it deals with textual information. The results have been evaluated and compared with various baselines and it is observed the proposed model gives superlative performance accuracy. The limitations of the model arise from the characteristics of real-time social data which are inherently ‘high-dimensional’, ‘imbalanced or skewed’, ‘heterogeneous’, and ‘cross-lingual’. The growing use of micro-text (wordplay, creative spellings, slangs) and emblematic markers (punctuations and emoticons) further increase the complexity of real-time opinion mining of the information.

5.2 Summarization

Our aim in this thesis is to detect whether a text review from different website would be a positive or not. In this dataset is gathered from the different website such as Rotten Tomatoes, IMDB and refined in a way that only short text is chosen.. We had the data instances labeled according to its comment contents as Negative or Positive. With the help of an ensemble model the particular review is tested against the trained datasets. Chapter 1 is the introduction where the concept of machine learning and its model is described and the purpose of the thesis is declared. In chapter 2 we review the related research done before. These papers build the basis of this work done.

The third chapter is the proposed methodology. In this how the the data is extracted and will be executed is described. Basically in this part the ensembling concepts and the psuedo code is explained. The fourth chapter is the implementation and result. In this all the setup of the libraries required is explained. Further the result on the review is generated and the graphical

representation of he code using Matplotlib is detailed. The last chapter is the conclusion. In this the future scope of the work done is described.

5.3 Discussion

According to the results, the contribution of general image features to the classification work is limited. One of the possible explanations is that the patterns of bullying are beyond merely image characteristics. On social media sites, a viewer comment under a post knowing information a lot more than just the post itself which is hard to simulate in a classification model. While trying different techniques to classify the data set and enhance the performance, we discovered a big diversity of the image content and patterns of bullying.

Due to the popularity of Instagram, Facebook, YouTube and many other online portals users post photos or comments with different purposes. Some post to promote products, some post to report news, some are organizations and post to gain popularity among viewers and some are common individuals who post to share experiences of their life. The bully might be intrigued by the identity of a user posting the comment in form of text, photo, or the content of images that the people who comment have strong sentiment about. These factors might require specific common sense knowledge to be recognized, which is sometimes hard for others without it to see and increase the difficulty of this classification problem.

Obviously, images are much more expressive as compared with text and if the image is embedded with text image or text itself represented as image then it further describes this function of expressiveness. We presented a model for cyberbullying detection that work for both, typo-graphic or info-graphic contents as well as simple text or image in order to capture this expressiveness. The proposed model for cyberbullying detection offers uniqueness in a way that it is able to handle various dimensions in the comments like: text, image and text + image to analyze the bullying.

Further, we have explored the use of deep learning technique and word embeddings for performing context-aware analysis of text. The performance results of the proposed model are motivating and improve the generic cyberbullying detection task. We have seen that if emoticons are taken into consideration then it improves the accuracy of entire model by 2.5 to 3 percent. The model works as a visual listening tool for brand management for enhanced social media monitoring and analytics. The main limitation from which the model suffers is that the text recognition for bullying is defined to only English language. As social media is a non-formal way of having communication, a prominent use of mash-up languages, like, a mix

of English and Hindi is widely seen, but such content be it text or text within image could not be processed.

5.4 Future Work

One way to alleviate the diversity of opinion mining is to find other ways in labeling. In this thesis, we just label positive or negative posts depending on the comments given on the specified website. Bit this could be highly biased. Also the authenticity of the review cannot be verified. Consider a simple example, in which some other rival group may unreasonably exploit a particular movie just for sake of it. This could result in the result that inaccurate.

In our thesis we have just considered the review in text form, irrespective of its authenticated user. To tackle this problem specified user profile can be considered. Also the age and the sex of the user for a particular review can give much detailed analysis of a movie. This will help in understanding which section of the society prefers a movie and to what extend.

The main contributions of this thesis are described below:

1. The scheduled approach is suitable for similar or mixed styles. Naive Bayes classification, Linear support vector classification and Multinomial Naive Bayes are the algorithm investigated for this particular thesis. A linear approach to the problem is used in the suggested technique. This can be further extended to deal with more difficult conditions.

2. The actual classification method based on voting was created by autonomous objective optimisation. However, a multi-independent ensemble method based on optimization is being worked out in the current thesis. Multifunctional and single-algorithm technologies aim to identify either a precise mix of voting by classifier or the maximum voting power to form a classifier.

3. The most common features of these techniques are known for no understanding and/or linguistic skills of a domain based. It's therefore simple to reproduce the techniques proposed for a resource poor language. For every language, such as Hindi, Urdu, Bengali etc., techniques can be extended.

4. The derived works apply to all types of classification problem like etiquette, NER, language etiquette, answer panel blog question, etc. To address these kinds of classifying ensemble issues using single and multi-objective optimisations is a new contribution to our best understanding. This methodology is also a recent challenge to tackle any kind of challenges in the region of natural language processing, particularly in NER.

5. Another reason for the motivation of assembly based technologies is to provide clients with a sequence of distinct replies with a high degree of accuracy or solutions with large percentages of information and solutions. Depending on the course of the issue or the particular requirement, suitable solutions can be acquired.

6. In this study we suggest a new way of integrating the classification systems which are accessible. Thus our suggested method can further improve the precision and efficiency of the works.

REFERENCES

[1] E. Aboujaoude, MW. Savage, V. Starcevic ,WO. Salame, Cyberbullying: review of an old problem gone viral.J Adolesc Health.2015;57(1):10–18, 2015.

[2] MA. Campbell,Cyber bullying: An old problem in a new guise?. Journal of Psychologists and Counsellors in Schools 15(1):68-76 ,2005.

[3] C. H. Behav, Tokunaga Following you home from school: a critical review and synthesis of research on cyberbullying victimization, 26:277–287, 2010.

[4] National Forum on Youth Violence Protection, Centers for Disease Control and Prevention. Youth violence: technology and youth protecting your child from electronic aggression, http://www.cdc.gov/violenceprevention/pdf/ea-tipsheet-a.pdf. Accessed 11 September, 2017.

[5] PK. Smith, J. Mahdavi, M. Carvalho, S. Fisher , S. Russell, N. Tippett, Cyberbullying: its nature and impact in secondary school pupils. J Child Psychol Psychiatry. 49(4):376–385 ,2008.

[6] S. Hinduja & J. W Patchin, Cyberbullying Identification, Prevention, and Response. Cyberbullying Research Center www.cyberbullying.us , 2014

[7] J. Brownlee, https://machinelearningmastery.com/best-practices-document-classification-deep-learning.

[8] Dadvar, Maral, and K. Eckert. "Cyberbullying Detection in Social Networks Using Deep Learning Based Models; A Reproducibility Study." ,2018.

[9] K. Dinakar, R. Reichart, and H. Lieberman. Modeling the detection of textual cyberbullying. In The Social Mobile Web, 21-30, 2011.

[10] Hosseinmardi, Homa, et al. "Prediction of cyberbullying incidents in a media-based social network." 2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM). IEEE, 2016.

[11] A. Mohammed, M. Elmogy, and H. Elbakry. "Content-based image retrieval using local features descriptors and bag-of-visual words." *Int J Adv Comput Sci Appl* 6.9, 212-219, 2015

[12] K. B. Kansara and N. M. Shekokar. A framework for cyberbullying detection in social network. 2015.

[13] Kumar, A. & Sachdeva, N. Multimed Tools Appl (2019). https://doi.org/10.1007/s11042-019-7234-z.

[14] Agrawal, Sweta, and Amit Awekar. "Deep learning for detecting cyberbullying across multiple social media platforms." *European Conference on Information Retrieval*. Springer, Cham, 2018.

[15] Woo Jun, https://github.com/jowoojun/kaggle-Toxic-Comment-Classification-Challenge.

[16] K. Dinakar, B. Jones, C. Havasi, H. Lieberman, and R. Picard. Common sense reasoning for detection, prevention, and mitigation of cyberbullying. ACM Trans. Interact. Intell. Syst., 2(3):18:1-18:30, Sept. 2012}.

[17] Hinduja, Sameer, and Justin W. Patchin. "Bullying, cyberbullying, and suicide."Archives of suicide research 14.3 (2010): 206-221.

[18] Kokkinos, Constantinos M., Nafsika Antoniadou, and Angelos Markos. "Cyberbullying: an investigation of the psychological profile of university student participants." Journal of Applied Developmental Psychology 35.3 (2014): 204-214.

[19] K. Dinakar, B. Jones, C. Havasi, H. Lieberman, and R. Picard. Common sense reasoning for detection, prevention, and mitigation of cyberbullying. ACM Trans. Interact. Intell. Syst., 2(3):18:1-18:30, Sept. 2012.

[20] M. Dadvar, F. M. G. de Jong, R. J. F. Ordelman, and R. B. Trieschnigg. Improved cyberbullying detection using gender information. In Proceedings of the Twelfth Dutch-Belgian Information Retrieval Workshop (DIR 2012), Ghent, Belgium, pages 23-25, Ghent, February 2012. University of Ghent.

[21] V. Nahar, S. Unankard, X. Li, and C. Pang. Sentiment analysis for effective detection of cyber bullying. In Web Technologies and Applications, pages 767-774. Springer, 2012.

[22] V. Nahar, S. Unankard, X. Li, and C. Pang. Semi-supervised learning for cyberbullying detection in social networks. In Databases Theory and Applications, LNCS'12, pages 160-171, 2014.

[23] A. K. K. Reynolds and L. Edwards. Using machine learning to detect cyberbullying. Machine Learning and Applications, Fourth International Conference on, 2:241-244, 2011.

[24] M. Ptaszynski, P. Dybala, T. Matsuba, F. Masui, R. Rzepka, K. Araki, and Y. Momouchi. In the service of online order tackling cyberbullying with machine learning and affect analysis. 2010.

[25] B. Nandhini and J. Sheeba. Cyberbullying detection and classifcation using information retrieval algorithm. In Proceedings of the 2015 International Conference on Advanced Research in Computer Science Engineering & Technology (ICARCSET 2015), page 20. ACM, 2015.

[26] A. Kontostathis, K. Reynolds, A. Garron, and L. Edwards. Detecting cyberbullying: query terms and techniques. In Proceedings of the 5th Annual ACM Web Science Conference, pages 195-204. ACM, 2013.

[27] B. S. Nandhini and J. Sheeba. Online social network bullying detection using intelligence techniques. Procedia Computer Science, 45:485-492, 2015.

[28] Yin, Dawei, et al. "Detection of harassment on web 2.0." Proceedings of the Content Analysis in the WEB 2 (2009): 1-7.

[29] Marathe, S. Sunil, and K. P. Shirsat. "Approaches for Mining YouTube Videos Metadata in Cyberbullying detection." *International Journal of Engineering Research & Technology, International Journal of Engineering Research & Technology (IJERT)* 4.05 (2015): 680-684.

[30] Q. Huang, V. K. Singh, and P. K. Atrey. Cyber bullying detection using social and textual analysis. In Proceedings of the 3rd International Workshop on Socially-Aware Multimedia, pages 3-6. ACM, 2014.

[31] R. Kelly, A. Kontostathis, and L. Edwards. "Using machine learning to detect cyberbullying." Machine Learning and Applications and Workshops (ICMLA), 2011 10th International Conference on. Vol. 2. IEEE, 2011.

[32] X. Zhi, and S. Zhu, "Filtering offensive language in online communities using grammatical relations." Proceedings of the Seventh Annual Collaboration, Electronic Messaging, Anti-Abuse and Spam Conference. 2010.

[33] Zerr, Sergej, et al. "Privacy-aware image classification and search."Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval. ACM, 2012.

[34] L. Henry, K.Dinakar, and B. Jones. "Let's gang up on cyberbullying." *Computer* 44.9 ,93-96, 2011.

[35] Vanhove, Thomas, et al. "Towards the design of a platform for abuse detection in OSNs using multimedial data analysis." 2013 IFIP/IEEE International Symposium on Integrated Network Management (IM 2013). IEEE, 2013.

[36] Rybnicek, Marlies, Rainer Poisel, and Simon Tjoa. "Facebook watchdog: a research agenda for detecting online grooming and bullying activities." 2013 IEEE International Conference on Systems, Man, and Cybernetics. IEEE, 2013.

[37] Zerr, Sergej, et al. "Privacy-aware image classification and search."Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval. ACM, 2012.

[38] Zhao, Rui, et al. "Deep learning and its applications to machine health monitoring." *Mechanical Systems and Signal Processing* 115 (2019): 213-237.

[39] Uysal, A. Kursat, and S. Gunal. "The impact of preprocessing on text classification." *Information Processing & Management* 50.1 (2014): 104-112.

[40] P. Jeffrey, R. Socher, and C. Manning. "Glove: Global vectors for word representation." *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. 2014.

[41] Noah Snavely, http : // www . Cs . Cornell . Edu / courses / cs1114 /2013sp /sections /S06_convolution .pdf .

[42] Kushal Vyag, https://kushalvyas.github.io/BOV.html

[43] P. Kralj Novak, J. Smailovic, B. Sluban, I. Mozetic, Sentiment of Emojis, PLoS ONE 10(12): e0144296, doi:10.1371/journal.pone.0144296, 2015.

[44] A. Kumar, , P. Dogra, and V. Dabas,V. August, Emotion analysis of Twitter using opinion mining. In 2015 Eighth International Conference on Contemporary Computing (IC3)(pp. 285-290). IEEE, 2015