A COMPREHENSIVE REVIEW OF VARIOUS SOCIAL MEDIA SENTIMENT ANALYSIS TECHNIQUES

SHUBHANGI MISHRA, VANSHIKA BHATNAGAR, ADITYA ASIJA

SUPERVISOR: DR PULKIT MEHNDIRATTA

DECEMBER 6, 2020

ABSTRACT

The entire world is transforming quickly under the present innovations. The Internet has become a basic requirement for everybody with the Web being utilized in every field. With the rapid increase in social network applications, people are using these platforms to voice them their opinions with regard to daily issues. Sentiment analysis (or opinion mining) is a common dialogue preparing task that aims to discover the sentiments behind opinions in texts on varying subjects.

Social media platforms and micro blogging websites are the rich sources of user generated data. Through these resources, users from all over the world express and share their opinions about a variety of subjects. The analysis of such a huge amount of user generated data manually is impossible, therefore an effective and intelligent technique is needed which can analyze and provide the polarity of this textual data. Multiple tools and techniques are available today for automatic sentiment classification for this user generated data. The purpose of this study is to explore the different techniques to identify its importance as well as to raise an interest for this research area.

INTRODUCTION

Social Computing is an innovative and growing computer field for the analysis and modeling of social networking that takes place in various forums. It is used to produce smart and collaborative applications to achieve effective results. The wide range of social networking sites allows people to share their feelings or ideas about a particular event, product or issue. The extraction of such informal and coherent data greatly helps to draw conclusions in various fields. However, the poorly constructed format of the ideas available on the web makes the mining process a challenge.

The textual information available on the web is highly divided into two categories: factual data and emotional data. Factual data is the term used to refer to various issues and workings of organizations or events. While emotional data are subordinate terms, describing the ideas or beliefs of an individual for a particular business, product or event. Emotional analysis is the process of knowing and distinguishing the different emotions that are conveyed online by people to find the author's approach to a particular product, topic or event into broadly: positive, negative or neutral. Emotional analysis has three main components of the study as follows: the owner of the senses which means the subject, the actual feelings i.e. belief and object, i.e. the topic the subject shared. Emotional analysis is performed at various levels ranging from the broader level to the finer level. High-level emotional analysis determines the emotions of an entire manuscript or document. Finer emotions analysis, and focuses on symptoms. The emotional analysis of Twitter data is done at the sentence level that goes between the broader and the finer. In the process of analyzing emotions, the emotions present in the text are of two types: Direct and comparative. The direct emotions in the text are independent of other things in the same sentence. However, the sense of comparison in the text indicates the comparison of different objects within the same sentence.

Emotional analysis strategies are used in various scenarios such as disaster relief and relief assistance, marketing and trade speculation, political election assessments, advertising markets, scientific research, customer loyalty testing, job creation, human health care and understanding of student learning experiences.

Twitter is a fast-growing micro-blogging site. In addition, Tweets are mostly public and are limited to 140 characters which make it easy to identify emotions in the text. Twitter Sentiment Analysis (TSA) handles the issue of analysis of the messages posted on Twitter in context of the sentiment expressed by tweets. The initial step of TSA process incorporates collection of tweets and marking them by their expressed sentiment. The next step involves extraction of feature sets needed for classifier training. The feature selection & combination of features shows a great impact on the performance of the classifier. Both the labelled data and the chosen features are sent to the algorithm and utilized to build a classifier model. In the last step, the classifier allots labels to testing data (unlabelled tweets). Selection of the suitable pre-processing methods comes out to be a great step in improving the classification accuracy.

In this paper, we introduce the process of analyzing the sentiments of Twitter data. Mostly, four approaches are used for this purpose Lexicon based techniques, Machine Learning based techniques, hybrid techniques (which combines lexicon based and machine learning based approach) and lastly ensemble techniques (which combines multiple machine learning algorithms).

METHODS AND EXPERIMENTAL DESIGN

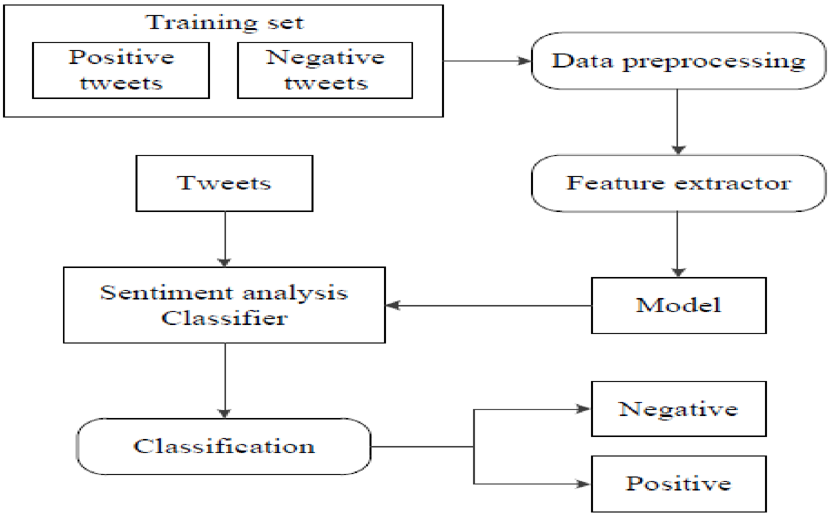


Fig 1 – Overview of Sentiment Analysis System

DATA COLLECTION & EXTRACTION METHODS

Data can be collected using four different methods which are as follows:

a. **Collecting the data from Repositories** such as UCI, Friendster, Kdnuggets, and SNAP

b. **Getting the data through search API and stream API**. Search API is used to collect Twitter data on the basis of hashtags and stream API is used to stream real time data from Twitter

c. **Collecting the data using automated premium tools** such as Radian6, Sysmos, Simplify360, Lithium and non-premium tools such as Keyhole, Topsy, Tagboard and SocialMention

d. **Extraction of data** (in our case tweets) **from Twitter API** using python libraries like tweepy and twython

DATA CLEANING & PREPROCESSING METHODS

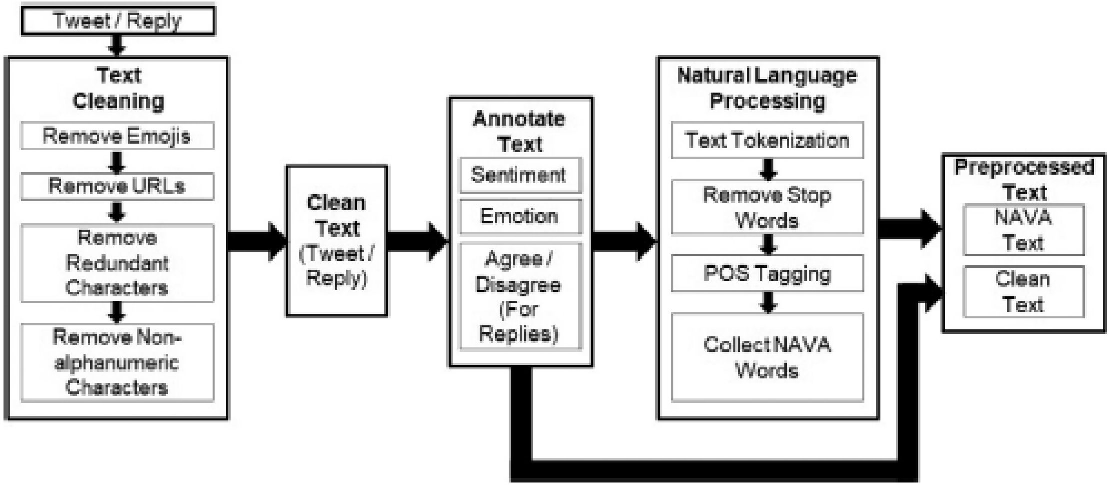


Fig 2 – Data Preprocessing

After collecting the data, it needs to be preprocessed and cleaned, thus allowing feature extraction from the input data. Data preprocessing is one of the major phases within the knowledge discovery process. Despite being less known than other steps like data mining, data preprocessing actually very often involves more effort and time within the entire data analysis process. Raw data usually comes with many imperfections such as inconsistencies, missing values, noise and/or redundancies. Performance of subsequent learning algorithms will thus be undermined if they are presented with low-quality data. Thus by conducting proper preprocessing steps we are able to significantly influence the quality and reliability of subsequent automatic discoveries and decisions. There are certain steps involved in preprocessing which are as follows:

a. Noise and Outlier Detection
b. Missing Feature Values
c. Normalization
d. Discretization
e. Instance Selection
f. Tokenization
g. Part of speech tagging (POS tagging)
h. Removing the hash tags, URLs, user tags, retweets, slang, incorrect spellings
i. Removal of stopwords
j. Stemming
k. Lemmatization
l. Feature Selection

One of the main problems to be tackled is that a variety of algorithms, such as instance-based learners, are sensitive to the presence of **noise**. This has the effect of misclassification which is based, for example, on the wrong nearest neighbor. This is so, because similarity measures can easily be falsified by data with noise in their values. A well-known preprocessing technique for finding and removing the noisy instances in the dataset is the noise filters. As a result, a new improved set of data is generated without noise and the resulting dataset can be used as input to a data mining algorithm. A simple filter approach is the variable-by-variable data cleaning. In this approach, the values that are considered as ‘suspicious’ values according to specific criteria are discarded or corrected. Thus, they can be included in the dataset that will be given as input to a data mining algorithm. In this approach, the criteria include, among others, the following: an expert evaluates the suspicious data as errors or as false labeled or in other cases a classifier predicts those values as ‘unclear’ data.

It is possible the appearance of dataset values called **outliers**. To deal with them creating a sorted list of values, with those values that are higher to be identified as outliers. Below we list methods used to identify outliers, as they are categorized in the following way:

a. Statistics-based methods
b. Distance-based methods
c. Density-based methods

There are classifiers that exhibit robust behavior in datasets with missing values, such as naive Bayes, and therefore the final result of the decision is not affected by the missing values. On the other hand, there are classifiers, such as neural networks and k-NN, that require a careful handling of the incomplete information. There are several methods and techniques for **handling missing data** which can be chosen as follows:

a. Most common feature value of a categorical attribute: Fill in missing values with the value that appears most often in the given dataset.
b. Concept most common feature value of a categorical attribute: Similarly to the above case, except that the missing values are filled by the values of the same class.
c. Mean substitution of a numerical attribute: Fill in missing feature values with the feature’s mean value that is computed using the available values. Alternatively, instead of using the ‘general’ feature mean, it can be used the feature mean of the samples that belong to the same class.
d. Regression method for a numerical attribute or classification methods for a categorical attribute: Develop a regression model for the numerical attribute case; fill in the blanks of the dataset by taking the decision-outcome of the model that is built in using all the other known features as predictors. Similarly, develop the classification model for the categorical case.

e. Method of treating missing feature values as special values: In this case, the unknown elements of the dataset are considered as complete new values for the features that contain missing values.

In various datasets that we have to manage, there are very often large differences between the feature values, such as the maximum and minimum value, for example, 0.001 and 10 000. In general, this issue is not desirable and requires careful intervention to make a scaling down transformation in such a way that all attribute values to be appropriate and acceptable. This process is known as **feature scaling or data normalization** (in the case of data preprocessing) and it is necessary and very important for various classifiers, such as neural networks, SVMs, k-NN algorithms, as well as fuzzy classifiers which cannot perform well if large differences between the feature values occur in the dataset. Normalization is applied to change the values of numeric columns in the dataset in order to use a common scale, without distorting differences in the ranges of values or losing information.

Another important issue to address is that of values of the continuous features of a dataset. Usually, the features of the sample can get continuous values or values from a wide range. This may require a time consuming procedure to handle or, in many cases, may be inefficient for the processes used in the field of predictive data mining. Thus, the main outcome of these and a solution to the above problem is the **discretization** process. The discretization process can actually be very useful to a variety of classifiers, such as decision trees (DTs) or Bayesian classifiers. Two of the most well-known and widely used discretization methods are:

a. Equal size: find the maximum and the minimum value of the attributes, and then divide the found range into k equal-sized intervals.
b. Equal frequency: find the number of values of the attribute and then it separate them into intervals which contain the same number of instances.

The instance selection approach is not only used to handle noise but also to deal with the infeasibility of learning from huge datasets. The training time required by the neural networks and SVMs as well as the classification time of instance-based learners is clearly affected by the size of the dataset. In this case, instance selection can be considered as an optimization problem that attempts to maintain the quality along with minimizing the size of the dataset. Moreover, by increasing the number of instances the complexity of the induced model is raised resulting to the decrease of the interpretability of the results. The sampling is a widely used process since it constitutes a powerful computationally intense procedure operating on a sub-sample of the dataset and is able to maintain or even to increase the accuracy. There is a variety of procedures for sampling instances from a large dataset. The most well-known are the following:

a. Random sampling that selects a subset of instances randomly and

b. Stratified sampling in the case where the class values are not uniformly distributed in the dataset.

In **tokenization** the longer strings of text is split into smaller pieces, or tokens. Larger chunks of text can be tokenized into sentences; sentences can be tokenized into words, etc.

A **stopword** is a commonly used word (such as “the”, “a”, “an”, “in”) that a search engine has been programmed to ignore, both when indexing entries for searching and when retrieving them as the result of a search query. We would not want these words to take up space in our database, or taking up valuable processing time. For this, we can remove them easily, by storing a list of words that you consider to stop words.

For grammatical reasons, documents are going to use different forms of a word, such as organize, organizes, and organizing. Additionally, there are families of derivationally related words with similar meanings, such as democracy, democratic, and democratization. In many situations, it seems as if it would be useful for a search for one of these words to return documents that contain another word in the set. The goal of both **stemming and lemmatization** is to reduce inflectional forms and sometimes derivationally related forms of a word to a common base form.

It is well-known and widely recognized that several learners, such as the k-NN procedure, is very sensitive relative to irrelevant features. In addition, the presence of irrelevant features can make SVMs and neural network training very inefficient and in many cases impractical. Furthermore, the

Bayesian classifiers do not perform well in the case of redundant variables. To tackle these issues, the **Feature Selection approach** is proposed to be used. The Feature Selection is the procedure of identifying and removing as many irrelevant and redundant features as possible. This reduces the dimensionality of the data and facilitates learning algorithms to operate faster and more effectively. In general, the features can be distinguished as follows:

a. Relevant: The features that have an influence on the class and their role cannot be assumed by the rest.

b. Irrelevant: Irrelevant features that do not have any influence on the class.

c. Redundant: A redundancy occurs whenever a feature can take the role of another (perhaps the simplest way to model redundancy).

In practice, it is not so straightforward to determine feature redundancy when a feature is correlated (perhaps partially) with a set of features. In general, the Feature Selection algorithms consist of two approaches:

a. A selection algorithm that generates the proposed subsets of features to find an optimal subset and

b. An evolutionary algorithm that decides about the quality of the proposed feature subset by returning some 'measure of goodness' to the selection algorithm.

On the other hand, without the application of a proper stopping criterion, it is possible the Feature Selection process to run in an exhaustively way or without termination through the space of subsets. Usually, the stopping criteria are:

a. Either the addition (or the deletion) of any feature that does not offer a better subset

b. Whether an optimal subset has been found according to some evaluation function.

FEATURE EXTRACTION

The preprocessed dataset has many distinctive properties. In the feature extraction method, we extract the aspects from the processed dataset. Later this aspect are used to compute the positive and negative polarity in a sentence which is useful for determining the opinion of the individuals using models like unigram, bigram. Machine learning techniques require representing the key features of text or documents for processing. These key features are considered as feature vectors which are used for the classification task. Some examples features that have been reported in literature are:

a. **Words and Their Frequencies:** Unigrams, bigrams and n-gram models with their frequency counts are considered as features. There has been more research on using word presence rather than frequencies to better describe this feature.

b. **Parts Of Speech Tags:** Parts of speech like adjectives, adverbs and some groups of verbs and nouns are good indicators of subjectivity and sentiment. We can generate syntactic dependency patterns by parsing or dependency trees.

c. **Opinion Words and Phrases:** Apart from specific words, some phrases and idioms which convey sentiments can be used as features.

d. **Position of Terms:** The position of a term with in a text can affect on how much the term makes difference in overall sentiment of the text.

e. **Negation:** Negation is an important but difficult feature to interpret. The presence of a negation usually changes the polarity of the opinion.

SENTIMENT ANALYSIS TECHNIQUES

Sentiment analysis can be defined as a process that automates mining of attitudes, opinions, views and emotions from text, speech, tweets and database sources through Natural Language Processing (NLP). Sentiment analysis involves classifying opinions in text into categories like "positive" or "negative" or "neutral". It's also referred as subjectivity analysis, opinion mining, and appraisal extraction. Sentiment Analysis is a term that includes many tasks such as sentiment extraction, sentiment classification, and subjectivity classification, summarization of opinions or opinion spam detection, among others. It aims to analyze people's sentiments, attitudes, opinions emotions, etc. towards elements such as, products, individuals, topics, organizations, and services. This paper takes in to account four sentiment analysis techniques, namely:

a. Lexicon-based techniques
b. Machine Learning- based techniques
c. Hybrid techniques (which combines lexicon based and machine learning based approach)
d. Ensemble techniques (which combines multiple machine learning algorithms)

Lexicon-based sentiment analysis methods rely on domain specific knowledge represented as a lexicon or dictionary. The process of sentiment calculation is based on identifying and keeping words that hold useful information while removing words that are not related. Lexicon based method uses sentiment dictionary with opinion words and match them with the data to determine polarity. They assigns sentiment scores to the opinion words describing how Positive, Negative and Objective the words contained in the dictionary are.

Lexicon-based approaches mainly rely on a sentiment lexicon, i.e., a collection of known and precompiled sentiment terms, phrases and even idioms, developed for traditional genres of communication. There are two sub classifications for this approach:

a. **Dictionary-based:** It is based on the usage of terms (seeds) that are usually collected and annotated manually. This set grows by searching the synonyms and antonyms of a dictionary. An example of that dictionary is WordNet, which is used to develop a thesaurus called SentiWordNet. The drawback which it faces is that it cannot deal with domain and context specific orientations.

b. **Corpus-Based:** The corpus-based approach have objective of providing dictionaries related to a specific domain. These dictionaries are generated from a set of seed opinion terms that grows through the search of related words by means of the use of either statistical or semantic techniques. Methods based on statistics: Latent Semantic Analysis (LSA). Methods based on semantic such as the use of synonyms and antonyms or relationships from thesaurus like WordNet may also represent an interesting solution.

WordNet is a large lexical database of English with 58058 words and 4 part of speech tags. Nouns, verbs, adjectives and adverbs are grouped into set of cognitive synonyms (synsets), each expressing a distinct concept. Synsets are interlinked by means of conceptual-semantic and lexical relations. Since the verbs have different tenses in documents and nouns have different forms of singular and plural, the size of the dictionary has increased uselessly while in the WordNet dictionary words are in their lemmatized form. Finally, WordNet is a popular and available dictionary. All the words are converted to their lower case form in terms of their letters, to be compatible with the WordNet words. Also all the numbers, punctuation marks and other elements are removed from dataset since they were not that much informative in comparison with other words. Each dataset is described in a two-dimensional matrix of documents and WordNet dictionary.

SentiWordNet is a lexical resource devised to support Sentiment Analysis applications. It provides an annotation based on three numerical sentiment scores (positivity, negativity, neutrality) for each WordNet synset. Clearly, given that this lexical resource provides a synset-based sentiment representation, different senses of the same term may have different sentiment scores.

Latent Semantic Analysis (LSA) is one of the basic foundation techniques in topic modeling. It is also used in text summarization, text classification and dimension reduction. It is similar to the cosine similarity. For LSA, we generate a matrix by using the words present in the paragraphs of the document in the corpus. The rows of the matrix will represent the unique words present in each paragraph, and columns represent each paragraph.

The main objective of **machine learning techniques** is to develop the algorithm that optimizes the performance of the system using training data such as examples and/or past knowledge and experiences. The machine learning provides a solution of the sentiment classification problem in two sequential steps:

a. Develop and train the model using training set data i.e. already labeled data.

b. Classifying the unlabeled or unclassified data based on the trained or skilled model.

Machine learning techniques are further classified into **supervised and unsupervised techniques**. To carry out sentiment analysis, typically the supervised machine learning techniques are used as we are dealing with subjective data. Supervised machine learning techniques highly depend on training data which are already labeled data unlike in the case of unsupervised machine learning techniques. Based on the provided training data, the classifier will classify the rest data i.e. test data.

Naïve Bayes classifier is a simple probabilistic classifier that uses the concept of mixture models to perform classification. The mixture model relies on the assumption that each of the predefined classes is one of the components of the mixture itself. The components of the mixture model denote the probability of belongingness of any term to the particular component. Thus, they are also known as generative classifiers. Naïve Bayes classifier is a probabilistic classifier that uses the concept of Bayes Theorem and finds maximum prospect of probability of any word fitting to a particular given or predefined class. The probability P is defined as follows:

$$\underbrace{P(c\,|\,x)}_{\text{Posterior Probability}}=\frac{\overbrace{P(x\,|\,c)}^{\text{Likelihood}}\,\overbrace{P(c)}^{\text{Class Prior Probability}}}{\underbrace{P(x)}_{\text{Predictor Prior Probability}}}$$

$$P(c\,|\,X)\;=P(x_1\,|\,c)\times P(x_2\,|\,c)\times\cdots\times P(x_n\,|\,c)\times P(c)$$

Fig 3 – Naïve Bayes Probabilistic Classifier

The **maximum entropy** relies on probability distribution estimation technique to perform classification. In this technique, firstly the categorized feature sets are converted into definite vectors using any of the encoding schemes. Secondly, this encoded vector is used to compute weights for each of the extracted features that can collectively support in determining the most prospective label for a feature set. It is used for various natural language processing tasks such as text classification. It depends on the probabilistic approach like Naïve Bayes. The fundamental concept of maximum entropy is that if much information regarding the data is not known, the distribution should be extremely uniform. This constraint eliminates the probability of non-uniform distribution. The probability is derived from the categorized training data and denoted as expected values of extracted features as follows:

$$P\ (c \mid d) = \frac{1}{Z\ (d)\{\exp(\sum \lambda i\ fi\ (c,d))\}}$$

Fig 4 – Maximum Entropy Classifier

Support vector machine (SVM) solves the traditional text categorization problem effectively; generally outperforming Naïve Bayes as it supports the concept of maximum margin. The main principle of SVMs is to determine a linear separator that separates different classes in the search space with maximum distance i.e. with maximum margin. If we represent the tweet using t, the hyper plane using h, and classes using a set Cj € {l, -1} into which the tweet has to be classified, the solution is written as follows equivalent to the sentiment of the tweet. The idea of SVM is to determine a boundary or boundaries that separate distinct clusters or groups of data. SVM performs this task constructing a set of points and separating those points using mathematical formulas.

Random Forest classifier is a tree-based classifier. It consists of numerous classification trees that can be used to predict the class label for a given data point based on the categorical dependent variable. For a given data point, each tree votes for a particular class label and the class label gaining the maximum votes will be assigned to that data point. The error rate of this classifier depends on the correlation or association among any two trees in the forest in addition to the strength of definite or individual tree in the forest. In order to minimize the error rate, the trees should be strong and the degree of associativity should be as less as possible. In the classifier tree, the internal nodes are represented as the features, the edges leaving a node are represented as tests on the feature's weight, and the leaves are represented as class categories. It performs classification preliminary from the root node and moves incrementally downward until a leaf node is detected. The document is then classified in the category that labels the leaf node. This algorithm is used in many applications of speech and language processing.

The **hybrid approach** of sentiment analysis exploits both statistical methods and knowledge-based methods for polarity detection. It inherits high accuracy from the machine learning (statistical methods) and stability from the lexicon-based approach.

Zainuddin et al. [1] proposed an aspect-based sentiment analysis (ABSA) framework, which contained two principal tasks. The first task used aspect-based feature extraction to identify aspects of entities and the second task used aspect-based sentiment classification. HCTS, STS, and STC datasets were used to evaluate the performance of the proposed hybrid model. This model incorporated rules after mining them with feature extraction methods. Single and multi-word aspects were identified based on a rule-mining technique with heuristic combination in POS patterns. Moreover, the Stanford dependency parser (SDP) was used to detect dependencies between aspects and opinions. Principal component analysis (PCA), latent semantic analysis (LSA), and random projection (RP) feature selection methods were also adopted in the experiments. The new hybrid model combining the ABSA framework, SentiWordNet lexicons, PCA, and the SVM classifier outperformed the existing baseline for sentiment classifications.

Asghar et al. [2] proposed a hybrid Twitter sentiment system that incorporated four classifiers: a slang classifier (SC), an emoticon classifier (EC), a general purpose sentiment classifier (GPSC), and an improved domain specific classifier (IDSC). The proposed framework identified the sentiment of tweets after detecting the presence of slang and emoticons. The results showed that computing the sentiment score of slang expressions lead to an improved accuracy in the sentiment classification of tweets.

The **ensemble classification** techniques have been widely used in many areas to solve the classification problem. But in case of tweet sentiment analysis, comparatively little work has been done on the use of ensemble classifiers. The performance of the ensemble classification system has been compared with various traditional sentiment analysis methods and most popular majority voting ensemble classification system. An example of ensemble classification system is formed by different base learners like Naïve Bayes classifier, Random Forest classifier, SVMs, and Logistic Regression. The results show that proposed ensemble classifier performs better than stand-alone classifiers and the popular majority voting ensemble classifier.

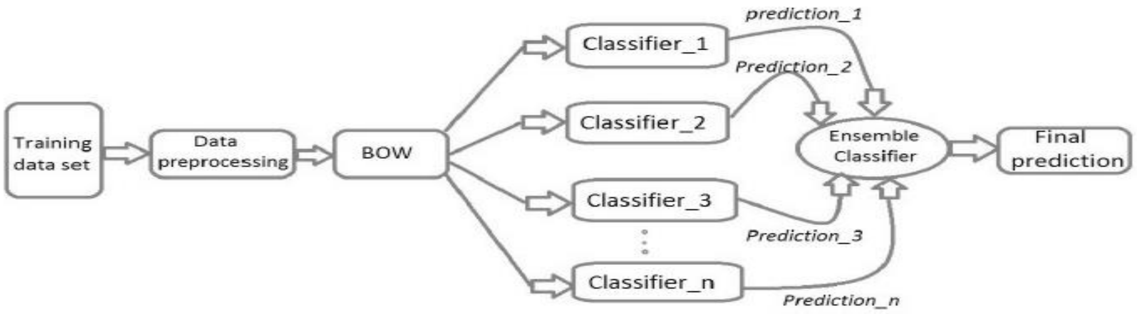


Fig 5 – Ensemble Classifier

LEVELS OF SENTIMENT ANALYSIS

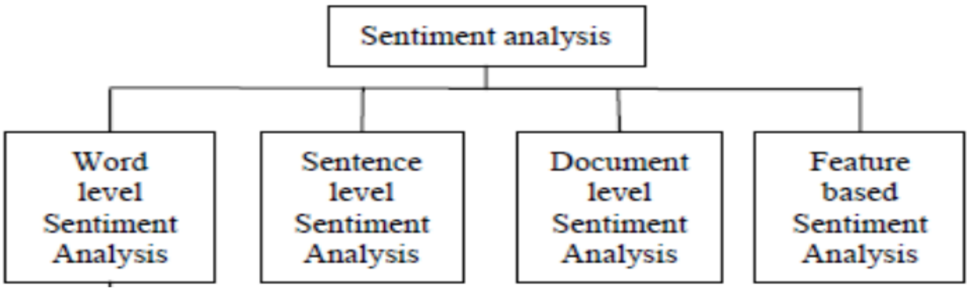

Sentiment analysis
Word level Sentiment Analysis
Sentence level Sentiment Analysis
Document level Sentiment Analysis
Feature based Sentiment Analysis

Fig 6 – Levels of Sentiment Analysis System

a. **Document level:**

- It deals with tagging individual documents with their sentiment. In Document level the whole document is classify either into positive or negative class.
- General Approach: Find the sentiment polarities of individual sentences or words and combine them together to find the polarity of the document.
- Various Tasks involved in this are:
    - Task: Sentiment Classification of whole document

    - Classes: Positive, negative and neutral
    - Assumption :Each Document focuses on a single object (not true in discussion posts, blogs, etc.) and contain opinion from a single opinion holder

b. Sentence or phrase level:

- Sentence-level Sentiment Analysis deals with tagging individual sentences with their respective sentiment polarities. Sentence level sentiment classification classifies sentence into positive, negative or neutral class.
- General approach: find the sentiment orientation of individual words in the sentence/phrase and then to combine them to determine the sentiment of the whole sentence or phrase.
- Various Tasks involved in this are:
    - Task 1: Identifying Subjective/ Objective Sentences
    - Task 2: Sentiment Classification of Sentences Classes: positive and negative

c. Aspect level or Feature level:

- It deals with labeling each word with their sentiment and also identifying the entity towards which the sentiment is directed. Aspect or Feature level sentiment classification concerns with identifying and extracting product features from the source data. Techniques like dependency parser and discourse structures are used in this.
- Various Tasks involved in this are:
    - Task1: Identify and extract object features that have been commented on by an opinion holder
    - Task2: Determining whether the opinions on features are negative, positive or neutral
    - Task 3: Find feature synonyms

d. Word Level:

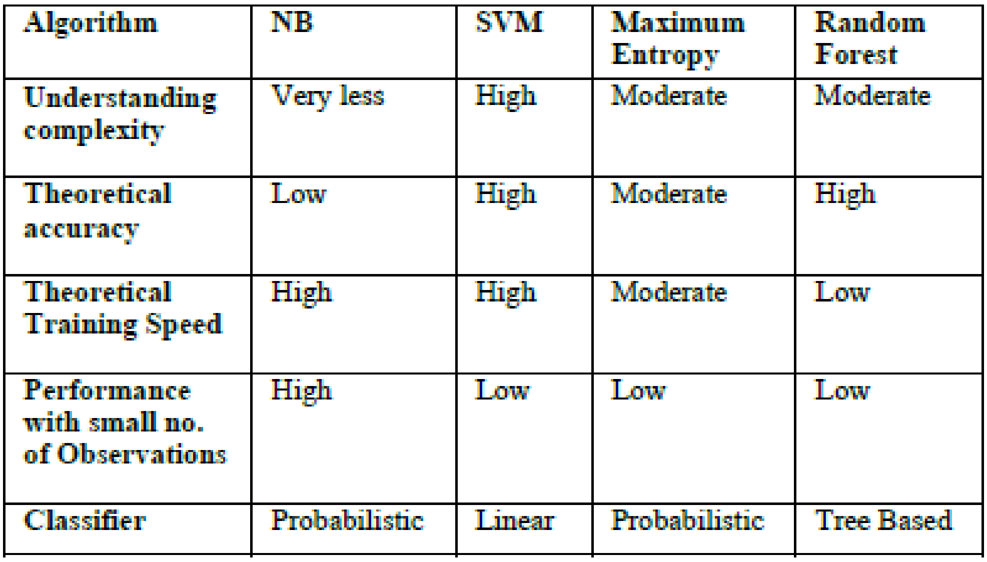
Most recent works have used the prior polarity of words and phrases for sentiment classification at sentence and document levels Word sentiment classification use mostly adjectives as features but adverbs. The two methods of automatically annotating sentiment at the word level are:

i. Dictionary-Based Approaches

ii. Corpus-Based Approaches.

EVALUATION OF SUPERVISED LEARNING ALGORITHMS

From our in-depth study of the above supervised machine learning algorithms used to perform sentiment analysis, we have identified several parameters such as understanding complexity, theoretical accuracy, theoretical training speed, performance with small number of observations and type of the classifier. Understanding complexity refers to the technical difficulties to understand the algorithm. Theoretical accuracy is the theoretical measure of how accurately the algorithm can classify the test set data according to the provided training data. Theoretical training speed refers to how fast the data can be trained. Performance is related to the accuracy of the algorithm. In general, accurate algorithms have good performance.

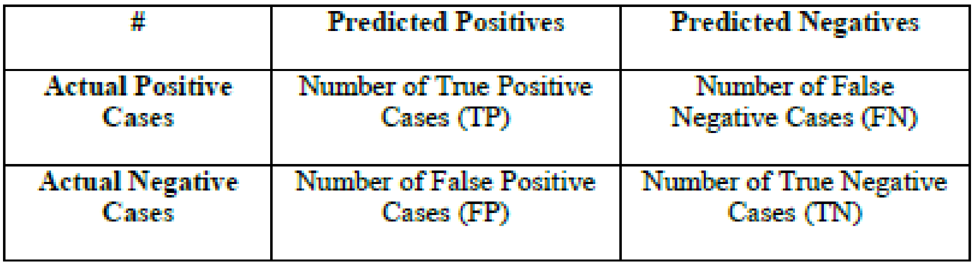


Algorithm	NB	SVM	Maximum Entropy	Random Forest
Understanding complexity	Very less	High	Moderate	Moderate
Theoretical accuracy	Low	High	Moderate	High
Theoretical Training Speed	High	High	Moderate	Low
Performance with small no. of Observations	High	Low	Low	Low
Classifier	Probabilistic	Linear	Probabilistic	Tree Based

Fig 6 – Comparison of Supervised Learning Algorithms

From the parametric comparison shown, we can conclude that Naïve Bayes algorithms are the simplest and easiest to understand and implement compare to Support Vector Machine and Maximum Entropy. However, it suffers from lower accuracy due to its simple Bayesian probability assumption. Whereas Support Vector Machine provides the better accuracy but it doesn’t support the automatic learning of features. Maximum Entropy provides the moderated accuracy but supports the automatic learning of features. Random Forest is based on decision tree method, which gives high accuracy with automatic feature learning. Though, the implementation accuracy of all these algorithms highly depends on the numerous factors such as domain chosen, data source, amount of data and preprocessing method applied on the data.

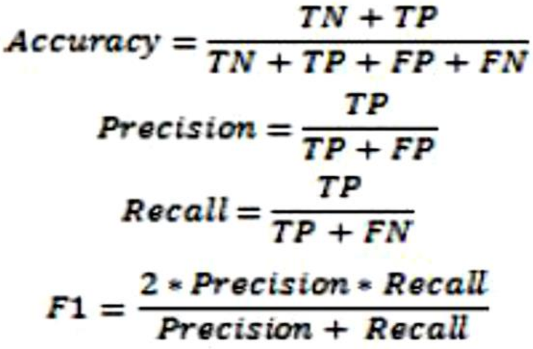
EVALUATION OF SENTIMENT CLASSIFICATION



#	Predicted Positives	Predicted Negatives
Actual Positive Cases	Number of True Positive Cases (TP)	Number of False Negative Cases (FN)
Actual Negative Cases	Number of False Positive Cases (FP)	Number of True Negative Cases (TN)

Fig 7 – Confusion Matrix

A **confusion matrix** is a table that is often used to describe the performance of a classification model (or "classifier") on a set of test data for which the true values are known.



$$Accuracy = \frac{TN + TP}{TN + TP + FP + FN}$$

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall}$$

Fig 8 – Evaluation Parameters

Accuracy is defined as all true predicted cases against all predicted cases. If we receive 100% accuracy, it denotes that the predicted cases are precisely the same as the actual cases.

Precision is defined as the true positive predicted cases against all positive predicted cases.

Recall is defined as the true positive predicted cases against all actual positive cases.

F1 is a harmonic average of the precision and the recall.

CHALLENGES IN SENTIMENT ANALYSIS

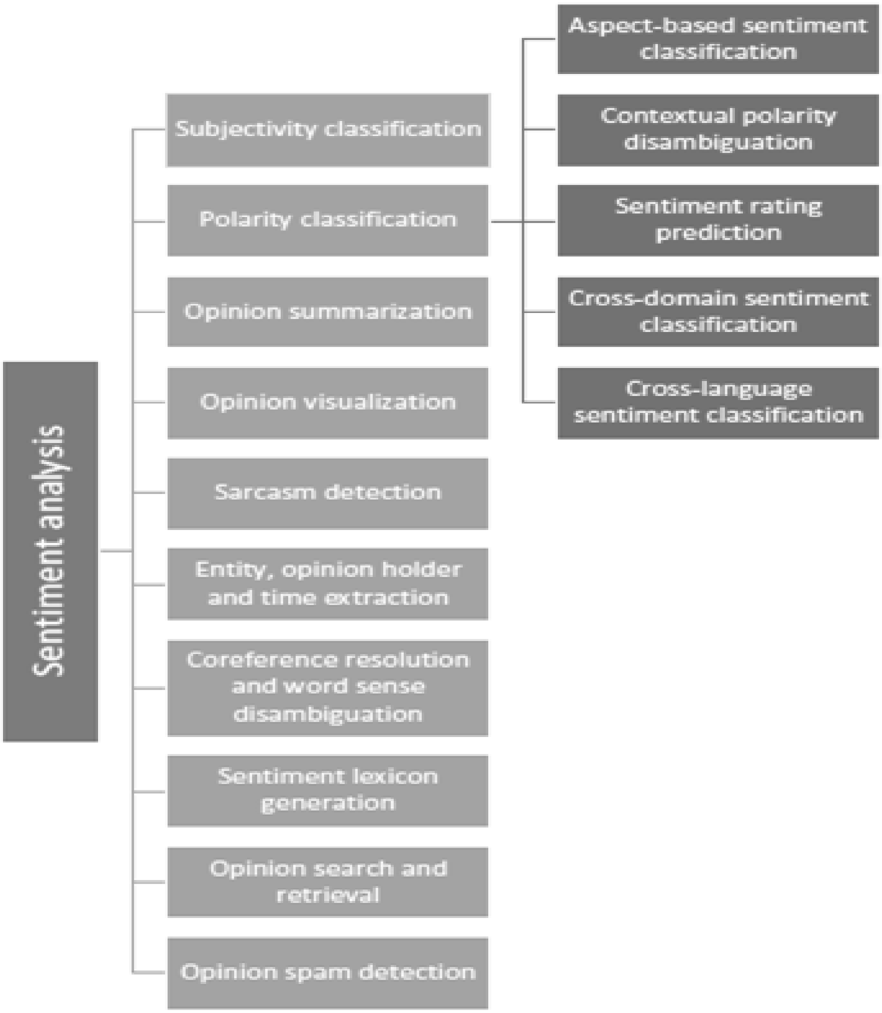

Sentiment analysis
Subjectivity classification
Polarity classification
Opinion summarization
Opinion visualization
Sarcasm detection
Entity, opinion holder and time extraction
Coreference resolution and word sense disambiguation
Sentiment lexicon generation
Opinion search and retrieval
Opinion spam detection
Aspect-based sentiment classification
Contextual polarity disambiguation
Sentiment rating prediction
Cross-domain sentiment classification
Cross-language sentiment classification

Fig 9 – Challenges faced in Sentiment Analysis

Polarity Shift is a most important issue to be addressed in Sentiment Analysis. Polarity Shift means that Polarity (Sentiment) of the sentence is calculated in different way from the polarity actually expressed in the Sentence. This problem is mainly arises due to polarity shifters such as negation and contrast.

Binary Classification is another important problem to be addressed in which the given review's Polarity is classified only by using "Positive", "Negative" by ignoring the "Neutral". This type of problem mainly arises when the sentiment classification is purely based on machine learning algorithms. Opinion mining that only considers positive and negative will not have good accuracy. Now-a-days the classification is extended by considering 5 possibilities such as "Positive", "Strong Positive"," Negative", Strong Negative" and "Neutral". By increasing the classification category it is possible to improve the accuracy of the opinion mining.

Data Sparsity problem which is caused due to the imposed character limit in micro blog/social media websites. For instance the maximum character limit in twitter is 140. Due to this limitation people will not express their opinion in clear manner. All these three issues are closely related to the accuracy of the sentiment analysis.

Identifying subjective parts of text: Subjective parts represent sentiment-bearing content. The same word can be treated as subjective in one case, or an objective in some other. This makes it difficult to identify the subjective portions of text.

Domain dependence: The same sentence or phrase can have different meanings in different domains. For Example, the word 'unpredictable' is positive in the domain of movies, dramas, etc., but if the same word is used in the context of a vehicle's steering, then it has a negative opinion.

Sarcasm Detection: Sarcastic sentences express negative opinion about a target using positive words in unique way. The sentence contains only positive words but actually it expresses a negative sentiment.

Thwarted expressions: There are some sentences in which only some part of text determines the overall polarity of the document. In this case, simple bag-of-words approach will term it as positive sentiment, but the ultimate sentiment is negative.

Explicit Negation of sentiment: Sentiment can be negated in many ways as opposed to using simple no, not, never, etc. It is difficult to identify such negations.

Order dependence: Discourse Structure analysis is essential for Sentiment Analysis/Opinion Mining.

Entity Recognition: There is a need to separate out the text about a specific entity and then analyze sentiment towards it. A simple bag-of-words approach will label it as neutral, however, it carries a specific sentiment for both the entities present in the statement.

Handling comparisons: Bag of words model doesn't handle comparisons very well.

Internationalization: Current Research work focus mainly on English content, but Twitter has many varied users from across.

APPLICATIONS OF SENTIMENT ANALYSIS

a. **Applications that use Reviews from Websites:**

   Today Internet has a large collection of reviews and feedbacks on almost everything. This includes product reviews, feedbacks on political issues, comments about services, etc. Thus there is a need for a sentiment analysis system that can extract sentiments about a particular product or services. It will help us to automate in provision of feedback or rating for the given product, item, etc. This would serve the needs of both the users and the vendors.

b. **Applications as a Sub-component Technology:**

   A sentiment predictor system can be helpful in recommender systems as well. The recommender system will not recommend items that receive a lot of negative feedback or fewer ratings. In online communication, we come across abusive language and other negative elements. These can be detected simply by identifying a highly negative sentiment and correspondingly taking action against it.

c. **Applications in Business Intelligence:**

It has been observed that people nowadays tend to look upon reviews of products which are available online before they buy them. And for many businesses, the online opinion decides the success or failure of their product. Thus, Sentiment Analysis plays an important role in businesses. Businesses also wish to extract sentiment from the online reviews in order to improve their products and in turn their reputation and help in customer satisfaction.

d. **Applications across Domains:**

Recent researches in sociology and other fields like medical, sports have also been benefitted by Sentiment Analysis that show trends in human emotions especially on social media.

e. **Applications in Smart Homes:**

Smart homes are supposed to be the technology of the future. In future entire homes would be networked and people would be able to control any part of the home using a tablet device. Recently there has been lot of research going on Internet of Things (IoT). Sentiment Analysis would also find its way in IoT. Like for example, based on the current sentiment or emotion of the user, the home could alter its ambiance to create a soothing and peaceful environment.

SIGNIFICANCE OF WORK

In this paper, we first introduced a detailed process of performing an emotional sentiment analysis process to classify the informal Twitter data into positive, negative or neutral categories. Second, we discussed various emotional analysis techniques on Twitter data including knowledge lexicon-based process, machine learning techniques, ensemble and hybrid-based techniques. It has been found that a variety of techniques used for emotional analysis are specific and specific to language. Therefore, future opportunities in the field of sentiment analysis include creating a process to create emotional isolation and extraction that can apply to any data outside the domain. In addition, language differences in data are an important issue that needs to be addressed in the future. Some of the most important challenges of Natural Language Processing (NLP) can also be used as additional advances in sensory analysis, such as the detection of hidden or hidden emotions, satire detection, comparison or association handling and emoticon detection.

REFERENCES

[1] A. Agarwal, R. Singh, and D. Toshniwal, "Geospatial sentiment analysis using twitter data for UK-EU referendum," Journal of Information and Optimization Sciences, vol. 39, no. 1, pp. 303–317, Oct. 2017.

[2] A. Alsaeedi M Khan, "A Study on Sentiment Analysis Techniques of Twitter Data," International Journal of Advanced Computer Science and Applications, vol. 10, no. 2, pp. 361–374, 2019.

[3] A. Deshwal and S. K. Sharma, "Twitter Sentiment Analysis using Various Classification Algorithms," in 2016 5th International Conference on Reliability, Infocom Technologies and Optimization, 2016, pp. 251–257.

[4] A. Hasan, S. Moin, A. Karim, and S. Shamshirband, "Machine Learning-Based Sentiment Analysis for Twitter Accounts," Mathematical and Computational Applications, vol. 23, no. 11, 2018.

[5] A. M. Abirami and V. Gayathri, "A SURVEY ON SENTIMENT ANALYSIS METHODS AND APPROACH," in IEEE Eighth International Conference on Advanced Computing (ICoAC), 2016, pp. 72–76.

[6] A. S. Raghuwanshi and S. K. Pawar, "Polarity Classification of Twitter Data using Sentiment Analysis," International Journal on Recent and Innovation Trends in Computing and Communication, vol. 5, no. 6, pp. 434–439, Jun. 2017.

[7] Ankit and N. Saleena, "An Ensemble Classification System for Twitter Sentiment Analysis," in International Conference on Computational Intelligence and Data Science (ICCIDS 2018), 2018.

[8] C. Dhaoui, C. Webster, and L. P. Tan, "SOCIAL MEDIA SENTIMENT ANALYSIS: LEXICON VERSUS MACHINE LEARNING," Journal of Consumer Marketing, 2017.

[9] D. Krishna Madhuri, "A Machine Learning based Framework for Sentiment Classification: Indian Railways Case Study," International Journal of Innovative Technology and Exploring Engineering, vol. 8, no. 4, pp. 441–445, 2019.

[10] F. A. Pozzia, E. Fersinib, E. Messinab, and B. Liuc, “International Journal on Recent and Innovation Trends in Computing and Communication,” in Sentiment Analysis in Social Networks., 2017, pp. 1–11.

[11] F. Neri, C. Aliprandi, F. Capeci, M. Cuadros, and T. By, “Sentiment Analysis on Social Media,” in ACM International Conference on Advances in Social Networks Analysis and Mining, 2012.

[12] F. Poeczea, C. Ebster, and C. Strauss, “Social media metrics and sentiment analysis to evaluate the effectiveness of social media posts Author links open overlay panel,” in The 9th International Conference on Ambient Systems, Networks, and Technologies (ANT 2018), 2108, pp. 660–666.

[13] F. Poeczea, C. Ebster, and C. Strauss, “Social media metrics and sentiment analysis to evaluate the effectiveness of social media posts,” in 9th International Conference on Ambient Systems, Networks and Technologies, 2018.

[14] G. X. Z Jianqiang, “Comparison Research on Text Pre-processing Methods on Twitter Sentiment Analysis,” IEEE Access, vol. 5, pp. 2870–2879, Feb. 2017.

[15] I. Gupta and N. Joshi, “Enhanced Twitter Sentiment Analysis Using Hybrid Approach and by Accounting Local Contextual Semantic,” Journal of Intelligent Systems, vol. 29, no. 1, pp. 1611–1625, Sep. 2019.

[16] K. Mishev, A. Gjorgjevikj, I. Vodenska, L. T. Chitkushev, and D. Trajanov, “Evaluation of Sentiment Analysis in Finance: From Lexicons to Transformers,” IEEE Access, vol. 8, Jul. 2020.

[17] K. Sailunaz and R. Alhajj, “Emotion and sentiment analysis from Twitter text,” Journal of Computational Science, vol. 36, 2019.

[18] M. Ahmad, S. Aftab, S. S. Muhammad, and S. Ahmad, “Machine Learning Techniques for Sentiment Analysis: A Review,” INTERNATIONAL JOURNAL OF MULTIDISCIPLINARY SCIENCES AND ENGINEERING, vol. 8, no. 3, Apr. 2017.

[19] M. Ebrahimi, A. H. Yazdavar, and A. Sheth, “Challenges of Sentiment Analysis for Dynamic Events,” EEE Intelligent Systems, vol. 32, no. 5, pp. 70–75, Sep. 2017.

[20] M. H. Abd El-Jawad, R. Hodhod, and Y. M. K. Omar, “Sentiment Analysis of Social Media Networks Using Machine Learning,” in 14th International Computer Engineering Conference (ICENCO), 2018.

[21] M. M. M Desai, “Techniques for Sentiment Analysis of Twitter Data: A Comprehensive Survey,” in International Conference on Computing, Communication and Automation, 2016, pp. 149–154.

[22] M. R. Huq, A. Ali, and A. Rahman, “Sentiment Analysis on Twitter Data using KNN and SVM,” International Journal of Advanced Computer Science and Applications, vol. 8, no. 6, pp. 19–25, 2017.

[23] M. Trupthi, S. Pabboju, and G. Narasimha, “SENTIMENT ANALYSIS ON TWITTER USING STREAMING API,” in IEEE 7th International Advance Computing Conference, 2017.

[24] M. Z. Asghar, F. M. Kundi, S. Ahmad, A. Khan, and F. Khan, "T-SAF: Twitter sentiment analysis framework using a hybrid classification scheme," Expert Systems, vol. 35, no. 1, 2018.

[25] N. Pagar and B. S. Satpue, “Survey Paper on Hybrid Approach for Twitter Sentiment Analysis using Supervised Machine Learning Algorithms,” International Journal of Engineering Research & Technology (IJERT), vol. 9, no. 1, pp. 513–515, Jan. 2020.

[26] N. Zainuddin, A. Selamat, and R. Ibrahim, "Hybrid sentiment classification on twitter aspect-based sentiment analysis," Applied Intelligence, pp. 1-15, 2017.

[27] P. Yang and Y. Chen, “survey on sentiment analysis by using machine learning methods,” in IEEE 2nd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC), 2017.

[28] R. Khan, P. Shrivastava, A. Kapoor, A. Tiwari, and A. Mittal, “SOCIAL MEDIA ANALYSIS WITH AI: SENTIMENT ANALYSIS TECHNIQUES FOR THE ANALYSIS OF TWITTER COVID-19 DATA,” Journal of Critical Reviews, vol. 7, no. 9, pp. 2761–2774, Aug. 2020.

[29] R. Xia, J. Jiang, and H. He, “Distantly Supervised Lifelong Learning for Large-Scale Social Media Sentiment Analysis,” IEEE TRANSACTIONS ON AFFECTIVE COMPUTING, vol. 8, no. 4, 2017.

[30] S Kiritchenko , W E. Hipson , R J. Coplan , S M. Mohammad, “SOLO: A Corpus of Tweets for Examining the State of Being Alone,” in Proceedings of the 12th Language Resources and Evaluation Conference, 2020, pp. 1567–1577.

[31] S. Ahuja and G. Dubey, “Clustering and Sentiment Analysis on Twitter Data,” in 2nd International Conference on Telecommunication and Networks (TEL-NET 2017), 2017.

[32] S. Liao, J. Wang, R. Yu, K. Sato, and Z. Cheng, “CNN for situations understanding based on sentiment analysis of twitter data,” in The 8th International Conference on Advances in Information Technology, 2016.

[33] S. Naz, A. Sharan, and N. Malik, “sentiment classification on twitter data using support vector machine,” in ACM International Conference on Web Intelligence (WI), 2018.

[34] Vishal A. Kharde , , S Sonawane, “Sentiment Analysis of Twitter Data: A Survey of Techniques,” International Journal of Computer Applications, vol. 139, no. 11, pp. 5–15, Apr. 2016.