SOCIAL MEDIA SENTIMENT ANALYSIS BASED ON AFFECTIVE-BEHAVIOURAL-COGINITIVE MODEL OF ATTITUDES

Dewamuni Adikaramge Chamodi Madhushani

(179333J)

Degree of Master of Science in Computer Science

Department of Computer Science and Engineering

University of Moratuwa
Sri Lanka

May 2020

SOCIAL MEDIA SENTIMENT ANALYSIS BASED ON AFFECTIVE-BEHAVIOURAL-COGINITIVE MODEL OF ATTITUDES

Dewamuni Adikaramge Chamodi Madhushani

(179333J)

Dissertation submitted in partial fulfilment of the requirements for the degree Master of Science in Computer Science Specializing in Data Science, Analytics and Engineering

Department of Computer Science and Engineering

University of Moratuwa
Sri Lanka

May 2020

DECLARATION

I declare that this is my own work and this dissertation does not incorporate without acknowledgement any material previously submitted for a Degree or Diploma in any other University or Institute of Higher Learning and to the best of my knowledge and belief this does not contain any material previously published or written by another person except where the acknowledgement is made in the text.

Also, I hereby grant to University of Moratuwa the non-exclusive right to reproduce and distribute my dissertation, in whole or in part in print, electronic or other medium. I retain the right to use this content in whole or part in future works (such as articles or books).

Name: D. A. C. Madhushani

Signature: ……………………………..   Date:...

The above candidate has carried out research for the Masters dissertation under my supervision.

Name of the Supervisor: Dr. Surangika Ranathunga

Signature: ……………………………….   Date:...

ABSTRACT

Sentiment Analysis is the study of classifying a given text based on its sentiment (positive/ negative polarity) of the expression. Sentiment analysis is being widely used to analyse the public opinion towards a given entity. Today in Web 2.0, social media is a popular platform to express one's opinions and beliefs. Therefore, researchers are keen on investigating how social media sentiment analysis can be improved to benefit interested entities. Most of the sentiment analysis research has been conducted on identifying the polarity (i.e.: positive, negative or neutral) and emotions (i.e.: happiness, sadness, disgust, anger, fear and surprise).

Comparatively, less focus has been given to study how expressions can be classified based on psychological aspects of attitude. The objective of the proposed research is to move beyond the mere polarity, and to investigate whether we can get an in-depth understanding of the expressed attitude. For this, we have used the ABC (Affective, Behavioural and Cognitive) model of attitude introduced in consumer psychology.

In this research a new dataset was compiled by extracting Tweets on a specific topic and manually annotating them based on the attitude by domain experts. This research discusses how existing tools and technologies of Sentiment Analysis can be applied for this problem domain. Various preprocessing and feature extraction techniques were evaluated against a set of machine learning algorithms including Ensemble and Deep Learning models. Additionally, this research aims to contribute to reduce the gap between machine learning and consumer psychology and thereby proving the possibility of applying machine learning across different domains.

DEDICATION

I dedicate this Masters dissertation to my beloved parents, Mrs. Rupika Wijesekara and Mr. Shantha Adikaram. I hope that this achievement will be one step closer to completing the dream that you had for me all those many years ago when you chose to give me the best education you could.

ACKNOWLDGEMENT

First and foremost, I would like to express my heartfelt gratitude to my supervisor, Dr. Surangika Ranathunga for providing abundant guidance, support and encouragement throughout this research. Special thanks to Ms. Irosha Amarabandu, Research Officer, Department of Psychiatry, Faculty of Medical Sciences, University of Sri Jayewardenepura for giving her invaluable support to conduct this research. Also, I would like to thank my colleagues for sharing knowledge, support and constant encouragement. I express my gratitude to University of Moratuwa for providing me with necessary resources and knowledge to make this project a success.

Last but not least, I express my love and gratitude to my beloved family for their continuous and unparalleled love, support and understanding.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF EQUATIONS

LIST OF ABBREVIATIONS

ABC	Affective Behavioural Cognitive
BOW	Bag of Words
CBOW	Continuous Bag of Words
CNN	Convolution Neural Networks
EDA	Exploratory Data Analysis
FS	Feature Selection
HMM	Hidden Markov Model
LDA	Latent Dirichlet Allocation
LSI	Latent Semantic Indexing
LSTM	Long Short-Term Memory
ML	Machine Learning
NLP	Natural Language Processing
POS	Part of Speech
RNN	Recurrent Neural Networks
ROS	Random Over-sampling
SA	Sentiment Analysis
SC	Sentiment Classification
SMOTE	Synthetic Minority Oversampling
SVM	Support Vector Machines
TF-IDF	Term Frequency-Inverse Document Frequency
TPE	Tree of Parzen Estimators

1. INTRODUCTION

1.1 Sentiment Analysis

Sentiment analysis (SA) is the process of computationally identifying and classifying opinions expressed in a text, especially in order to determine whether the writer's attitude towards a particular topic, product, political party, policy or any entity is positive, negative, or neutral [1].

In SA there are 3 main types of classification problems—document level analysis, sentence level analysis and aspect level analysis [1]. Document level analysis considers the full text of expressed opinion to determine the sentiment. Whereas in sentence level, the sentiment expressed in each sentence of the text is separately analysed. Aspect level breaks down the sentence into a more granular level to identify individual aspect entities within the sentence, and the sentiment expressed on each of these aspects [1].

Even though sentiment analysis commonly polarises attitudes into negative, positive or neutral, it also has the ability to interpret more complex emotions such as happiness, sadness, excitement etc. in a given context. All these applications fall under the umbrella of SA. SA is mainly conducted on data that is obtained from web pages, books, news reports, blogs and microblogs [2].

In the past few years, the number of studies on social media SA has been increased [2]. Availability of free and open platforms such as social media allows individuals to independently and easily express their opinions and beliefs. Hence it has become a goldmine for analysts to mine for public opinions. This is widely being used in areas such as business, politics, health and socio-economic sector to evaluate public opinions and stances. Especially for businesses, being able to capture the sentiment behind the responses of the public helps marketers to determine if their marketing initiatives are driving the actions that they planned for, while also giving them feedback for adapting their strategies in case their touch points are not resonating with the targeted consumers.

How social media SA differs from any other sentiment analysis is due to its unique features such as informal language, sarcasm, short words, hashtags, emoticons and abbreviations in its large vocabulary. These features make social media SA quite a challenging subject. Hence numerous studies have been carried out to improve the accuracy by trying to tackle these challenges.

1.2 Types of attitudes

Attitude refers to people's evaluation of virtually any aspect of their social world. People can have favourable or unfavourable reactions to issues, products, ideas, objects, specific behaviour or entire social groups. As defined Allport [3], "Attitude is a mental or neural state of readiness, organized through experience, exerting a directive influence upon the individual's response to all objects and situations with which it is related."

There are various models of attitudes discussed in social science and psychology. Generally, attitudes can be divided into positive, negative and neutral. This is known as classifying based on polarity [4]. This is the attitude model widely used in classification tasks as it can be applied to any given context. Nevertheless, there are many other models that are lesser known in the SA paradigm but they are widely applied in the field of consumer psychology to better understand and categorise consumer attitudes. Mainly there are 4 structural models of attitudes discussed in consumer psychology [5], namely,

1. Tricomponent Attitude Model
2. Multiattribute Attitude Model
3. The Trying-to-Consume Model
4. Attitude-Toward-the-Ad Model

Tricomponent model discusses the 3 aspects- affective, behavioural and cognitive, which adheres with emotional, physical and mental aspects respectively. Multiattribute Attitude Model explains consumer's attitudes as a formation of his/her perceptions and beliefs on the attributes of the attitude object. The Trying-to-Consume model focuses on the consumer's effort to consume despite the actual outcome that may vary due to personal and environmental factors. Attitude-Toward-the-Ad Model studies how the feelings and judgements are formed in consumers as a result of being exposed to an advertisement [5].

The main objective of all these structures is to understand the underlying dimensions of human attitudes, and eventually predict the outcomes or behaviours using that knowledge [6]. The difference between these models is how they capture these dimensions and their interrelations.

Tricomponent Attitude Model, which is also known as the ABC model of attitude is the most abstract and widely used model [7]. Therefore, this research will focus only on this model.

1.2.1 Tricomponent Attitude Model (ABC model of attitude)

Human behaviour can be viewed as a combination of mental, emotional, and physical dimensions. This is popularly referred to as the "think-feel-do" perspective [6]. Hence, in this model, attitudes are categorised into 3 types– Affective, Behavioural (also known as Conative) and Cognitive [7].

1. **Affective** – feelings/ emotions toward the attitude object (emotional aspect).
2. **Behavioural**– tendencies to do certain things with respect to the attitude object (physical aspect).
3. **Cognitive** – beliefs and knowledge about what is or is not true with respect to the attitude object (mental aspect).

Each of these attitudes can be further categorized into negative or positive attitudes (i.e. negative affective attitude or positive affective attitude, negative behavioural attitude or positive behavioural attitude, etc.). A few examples the author observed with regard to the mobile phone domain have been given in Table 1.1.

Table 1.1: Example of classification of attitudes based on ABC model

Type of Attitude	**Polarity**	
	Positive	**Negative**
Affective	"I love the new iPhone 11"	"I hate the new iPhone 11"
Behavioural	"I am definitely going to buy the new iPhone 11!"	"I shall probably wait for iPhone 12 without buying iPhone 11"
Cognitive	"Yay! finally the new iPhone 11 comes with extended battery life."	"iPhone 11 camera is not satisfactory."

1.3 Problem Statement and Motivation

SA is applied to understand the attitude of consumers towards a particular product, service, marketing campaign, new release, etc. Generally, in SA attitude is polarized into positive and negative attitudes.

As shown in Figure 1.1, many factors affect the buying behaviours and choices of consumers. More than the factual demographic features such as income and age, consumer psychology suggests that buying behaviour largely relies on consumers' attitudes and expectations [8]. Therefore, it is not sufficient to determine only the conventional polarity of the attitude towards a specific product or service. With the advancement of this research area, businesses shall be able to get a deeper understanding about views of their consumers beyond the mere polarity, thereby being able to change unfavourable attitudes into favourable attitudes and to retain favourable attitudes. As shown in Figure 1.2, Figure 1.3 and Figure 1.4, some companies may try to persuade customers to consume their product by addressing all dimensions of attitudes.

But attitudes shall not be confused with emotional analysis. Attitude is a mental state of readiness learned and organized through experience [3] whereas emotion is a state of physiological arousal accompanied by changes in facial expressions, gestures, posture, or subjective feelings [9]. Even though emotional sentiment analysis has its own branch of study under SA [2], attitude SA is given less focus beyond the binary classification of the polarity. There has not been any research done on SA using the ABC model of attitude.

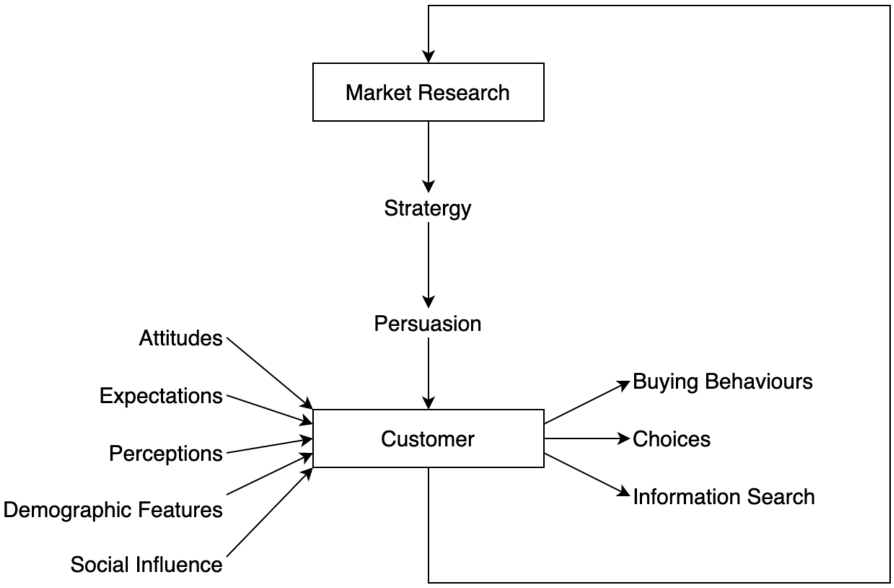


Figure 1.1: What affects the customers buying behaviours and choices

Source: Adapted from [8]

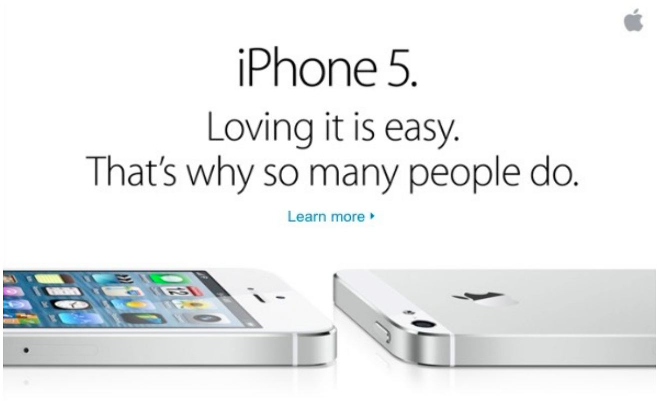
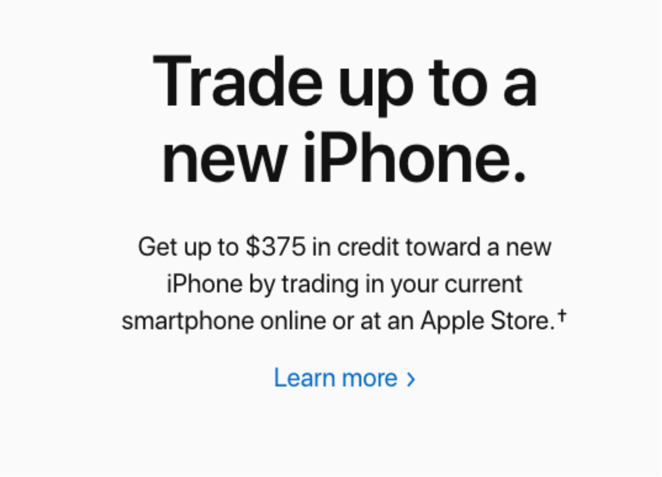


Figure 1.2: Affective advertising



Trade up to a new iPhone.

Get up to $375 in credit toward a new iPhone by trading in your current smartphone online or at an Apple Store.†

Learn more ›

Figure 1.3: Behavioural advertising

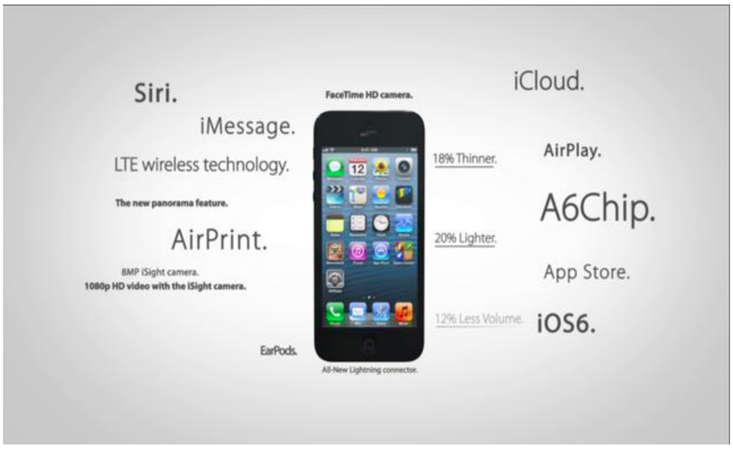


Figure 1.4: Cognitive advertising

1.4 Objectives

- Study the applicability of existing SA techniques in classifying customer attitudes according to the ABC model of attitudes.
- Produce a working model using SA techniques to classify attitudes expressed in Tweets based on the ABC model of attitudes along with a comprehensive evaluation and comparison of results using various techniques. Further classification of attitudes based on the polarity will not be considered in this research.

1.5 Research Scope

Research boundary or the areas that will be addressed and will be ignored is discussed in this section.

- Sarcasm and irony will not be addressed in this solution.
- The main focus of this research is to classify attitudes based on the ABC model. Neutral attitudes will not be considered as it is not a part of the model.
- Only the Affective, Behavioural and Cognitive attitude classes will be considered and not its Negative and Positive polarity.
- A text will be classified based on only one class label. Neither multiple classes will be predicted nor intensity levels/ranks will be calculated.
- This proposed solution is not an improvement of social media SA. The intention is to make use of social media SA to classify attitude sentiment.
- Only SA towards products will be studied in this research.

1.6 Thesis Organization

The remainder of the paper is organized as follows. In Chapter 2, previous work done in the SA domain will be discussed and compared. In Chapter 3 the implementation and the methodologies used will be discussed. Results of the final solution will be presented and evaluated in Chapter 4. Chapter 5 concludes the research.

2. LITERATURE REVIEW

This section contains a literature review conducted on the research that has studied various novel and traditional approaches in the paradigm of SA.

The set of steps illustrated in Figure 2.1. is applicable to any classification model. Likewise, sentiment analysis, which is a type of classification problem adheres to a similar set of steps. Once after the data collection has been carried out, the data set needs to be preprocessed in order to remove any noise and to normalize the text to build a reliable classification model. Feature engineering step is to extract information from the data set and come up with a feature set for the model. All the features that have been extracted will not be needed to build the model. Feature selection algorithms help to sort out the most suitable features that need to be fed into the model. Then the sentiment analysis model can be built using the training dataset. There are numerous classification algorithms that can be used to build a sentiment analysis model. Finally, model can be evaluated using predetermined metrics. These few steps are often repeated until a model with satisfactory results is obtained. During the repeated phases different training sets, preprocessing techniques, feature sets, feature selection algorithms and classification algorithms can be tested out, while the evaluation metrics are kept constant for the purpose of model comparison [10], [1].

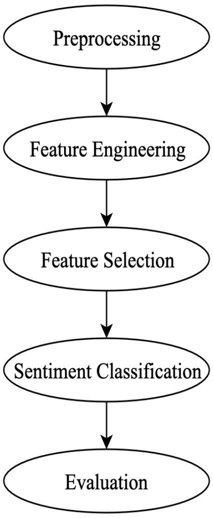


Figure 2.1: Steps of SA Model

Source: Adapted from [10]

2.1 Pre-processing

According to Fayyad et al. [11], the total percentage of noise in any given dataset can amount up to 40%. This noise can cause confusion in machine learning algorithms which will result in erroneous models simply causing a scenario of "garbage in garbage out". Hence, preprocessing is done in order to cleanse and normalize the text. It is one of the most crucial steps in building a sentiment analysis model which is often underestimated. This step is vital especially when dealing with informal documents such as tweets and other social media microblogs. These documents contain unique features such as URLs, slangs, user mentions, emoticons and various other informal language components which makes this domain challenging [10], [12]. There are various preprocessing steps employed in text classification tasks. Some of the most common and important preprocessing techniques that have been successfully incorporated in previous studies done on social media SA will be discussed in this section.

2.1.1 Handling Numbers

In this technique the numbers in the text is simply removed. Whether to apply this technique or not entirely depends on the problem domain. If the domain is number sensitive, it is argued that this technique will reduce the effectiveness of the model. For such instances, Celikyilmaz et al. [13] have used a technique where the numeric texts such as $1.99, 60%, 67 have been replaced by tags such as <DOLLAR-AMOUNT>, <PERCENTAGE> and <NUMBER> respectively. Furthermore, in social media sentiment analysis with informal language, numbers can also represent words e.g.: 4 (for), 2 (to or too). But in most cases, it is believed that it holds no value or sentiment [14].

2.1.2 Handling Punctuations

Removing punctuations is one of the most basic preprocessing techniques. This includes getting rid of all punctuation in the text such as '?', '!' and '.'. Angiani et al. [10] suggested keeping the apexes as they are a part of grammar constructs such as the genitive.

However, repetition of punctuation marks (e.g.: ‘???’, ‘!!!’, ‘...’) in the social media context is usually a means of expressing strong emotions. In such contexts it is not wise to blindly remove the punctuation marks especially when the sole purpose is to understand the emotions expressed [14].

But varied number of punctuation marks (e.g.: ‘???’ vs ‘?????????’) will result in making the dataset sparse. Hence, Effrosynidis et al. [12] suggested replacing such repeated punctuation marks with a representative tag (i.e: ‘multiQuestionMark’) that will preserve the intended sentiment while normalizing the text.

2.1.3 Lowercasing

Converting all the text into one case, usually lowercase, is done in order to make the text uniform and to match occurrences in the training data [10]. But words in all capitals in the social media domain will sometimes be a means of expressing strong emotions [14]. Therefore, Effrosynidis et al. [12] used a special technique to handle capitalized words. In their approach they have added a prefix ‘ALL_CAPS_’ to such words, so that the intended sentiment is preserved.

2.1.4 Replace Slang and Abbreviations

Social media documents are full of slang and unique abbreviations. Angiani et al. [10] used an extension of Python library PyEnchant to replace the slang with its formal meaning (e.g.: replace ‘l8’ with ‘late’). They were also able to replace offensive words with the tag ‘bad_word’. Similarly, Effrosynidis et al. [12] also have manually compiled and used a list of 290 such slang words and their replacements.

2.1.5 Spelling Correction

It is expected to have spelling errors in informal text such as tweets. By using spell correctors these terms can be corrected in order to improve the effectiveness of the classifier [12].

Elongated words are the words with vowels repeated in sequence at least three times. A basic dictionary might not be able to correct elongated words. This will result in having the same word written in two different ways (i.e.: ‘cool’ and ‘coooool’). Therefore, it is important to normalize such words so that the classifier won’t treat them as separate words [10], [12]. One approach is to replace such repeating letters with only two repeated letters (e.g.: “coooool” replaced with “cool”) as most English words have a maximum of only two repeated letters. But this will not completely address situations such “looooove” as it will be replaced “loove” which is not its original term [14]. But most of such trivial cases can be addressed using a spell corrector on top of them.

2.1.6 Handling Negations

Examples of negations are ‘can’t’, ‘didn’t’ and ‘won’t’. Literature has suggested many ways to handle negation. One of the approaches is to replace contractions, by substituting words such as “won’t” and “don’t” with “will not” and “do not” respectively [12].

Another approach is detecting words that imply negation and then adding the prefix “NOT” in every word after them until the occurrence of the first punctuation mark [15]. Another study suggests replacing the negation with antonyms [12]. For an example phrase “not good” will be replaced with the word “bad”.

In [10], a simpler technique was implemented where all the negation terms have been replaced with “not”. For an example “can’t miss” will be replaced with “not miss”. This technique allows the classifiers to preserve a lot of negation bigrams without excluding them due to their low frequency. For an example consider the terms “can’t miss”, “won’t miss”, and “don’t miss”, where all the terms express the negation of “miss”. By replacing the negation term with “not”, the resulting term will be “not miss” in all three scenarios.

2.1.7 Replace URLs, Hashtags and User Mentions

Most of the SA studies done on social media do not consider hashtags (e.g.: #iphone), URLs and user mentions (e.g.: @apple) during the classification process, therefore these terms are omitted during the preprocessing stage as they rarely contain any sentiment [10]. But studies [12] and [15] suggest to replace such terms by tags such as "URL" and "AT_USER" to normalize such terms while preserving the information that will be useful in some cases of classification.

2.1.8 Handling Emoticons

Emoticons are used to express emotions in informal text. Therefore, removing emoticons will result in loss of useful information. Instead these can be replaced with tags. By limiting the emoticon tags only into two categories (i.e. smile_positive and smile_negative) classifiers will be able to increase the weight of such features and to reduce the complexity of the model [10].

E.g.: smile_positive → :) , :P , :D , ;)

smile_negative → :(, :/ , >:(

2.1.9 Tokenization

Tokenization is the process of breaking down a given text into words, phrases, symbols and other meaningful chunks by removing punctuation marks [4].

2.1.10 Part of Speech Tagging

This is another common technique especially in Twitter sentiment analysis [16]. POS tagging helps in identifying adjectives, verbs and adverbs which are usually strong indicators of emotions [17], and also to create feature sets using POS tags [15].

2.1.11 Stop words removal

Stop words such as "of", "a" and "the" which are commonly occurred in the text are removed because such words don't discriminate for any particular sentiment class. But this should be done carefully as some common words could be useful in specific domains [18].

2.1.12 Stemming and Lemmatization

Task of both these techniques is to bring the terms to their base form. Porter [19] and Lancaster algorithms [20] are two widely known algorithms in text classification used for “stemming”. But Lemmatization has replaced stemming as it yields more accurate results. For example, the words “caring” and “cars” are reduced to “car” in a stemming process whereas lemmatization reduces it to “care” and “car” respectively.

Effrosynidis et al. [12] evaluated the effect of most of the common preprocessing techniques on the three different sentiment analysis classifiers namely, Linear SVC, Bernoulli Naive Bayes and Logistic Regression. In their comparison, they concluded that removing numbers, replacing repeated punctuation marks and stemming as the best preprocessing techniques that improved performance of all three classifiers. They also observed that handling negation by adding the prefix “NOT” in every preceding word, lemmatizing and replacing URLs, hashtags and mentions with tags also yielded fairly good results. Interestingly, while replacing the repeated punctuation marks was observed as one of the best techniques, removing punctuation marks resulted in lowest accuracy in all three classifiers showing their importance in sentiment analysis. Rest of the techniques such as spell correction, lower case and slang replacing were observed to have poor impact or varying impact on the classifier. Angiani et al. [10] who conducted a similar study also found out that stemming had the best effect on the classifier performance. They also observed that replacing negation with “NOT” and replacing emoticons with tags had a fairly good effect on the performance of their selected sentiment analysis classifier, Multinomial Naive-Bayes.

2.2 Feature Extraction

The preprocessed data needs to be converted into a vector of features in such a way that classifier can learn the patterns. Hagenau et al. [21], observed that the results of classification task were not very dependent on the machine learning algorithm but strongly depends on the features selected. Features in SA can be categorised into three types, namely, sentiment features, content specific features and content free features [22]. Content specific features are the ones obtained from the dataset vocabulary (e.g. bag of words and word n-grams). Whereas, content free features consider the document structure (e.g.: part-of-speech features). Sentiment features calculates a sentiment score for each term using a tool such as SENTI-WORDNET.

2.2.1 Bag of words

These features capture the frequency of a word from the vocabulary appearing in a document. This is the most basic technique of feature engineering. Even though it is a simple and easy to use technique, it has its own drawbacks. It views the document as an ordered collection of words, causing the sentences with different meanings but containing the same words to have an identical representation [23].

2.2.2 Word N-grams

Word N-grams give a better solution for word ordering by introducing bi-grams. However, most of the researchers have incorporated both unigrams and bigrams in the feature lists [24], [25].

2.2.3 Word Embedding

Neither of the above mentioned features can capture the semantic information of words. The notion of word embedding is to represent the terms in a vector in such a way that relationships among the terms can be derived using simple algebraic expressions. Word2vec [26] and GloVe [27] are two word embedding models used across SA problems. Barry [23], was able to achieve significantly better performance by using word embedding compared to the baseline model using bag-of-word features.

2.2.4 Word Clusters

Word embeddings can be used to create word clusters. It is an unsupervised representation of words. Several studies done on SemEval 2014 datasets have shown that these features have improved the classification accuracy when added together with other features [28], [29].

2.2.5 Part-of-speech Features

Documents are represented as an array of counts for each POS tags such as verbs, nouns and adjectives [15]. Saif et al. [16] observed that they were able to achieve better results by using the TweetNLP POS tagger compared to the traditional treebanks POS tagger as it is capable of identifying Twitter specific features such as abbreviations, emoticons and interjections.

2.2.6 Emoticons

Emoticons are categorised into classes. This feature set captures frequency of occurrence of each class of this emoticons within the document [24], [17]. In the study [24] it was observed that classifiers that use combination of all features, unigrams, bigrams, POS, emoticons yielded the best results compared to classifiers that only use unigrams and bigrams features.

2.2.7 TF-IDF

Most of the above explained featured are either based on binary (term presence) or term frequencies. TF-IDF calculates a normalised value of significance of a term in a document. Even though binary and frequency based weights have been able to achieve good results, studies have shown that TF-IDF based weights have outperformed them [24].

2.2.8 Statistical and Meta Information on a Message

Some studies have used meta information in additional to the textual data. These features contain information such as length of a document and timestamp [21].

2.3 Feature Selection

Duric and Song [30], listed down 5 criteria that are useful when selecting features for a good SA model.

1. Features should be informative and interpretable.
2. All features should give an overall understanding of the entire corpus.
3. Features should be as domain-dependent as possible.
4. Features must be frequent enough.
5. Features should be discriminative enough for the classifier to understand underlying patterns.

All the features that are derived during the feature engineering phase will not fit to all the above listed criteria. Therefore, there are various statistical techniques to filter out the most suitable features to build an accurate and robust SA model. A few of those techniques are explained in this section.

2.3.1 Frequency cut off

This is the most basic feature selection technique. Here a cut of value is set to filter out the features weighted based on frequency [30]. Some studies have selected only the top 5000 as features when ranked according to the frequency value from the vocabulary [25], [23]. Other researchers have considered only the features that have a minimum frequency of 5 [21].

2.3.2 HMM LSI and LDA

Hidden Markov Model (HMM) and Latent Semantic Indexing (LSI) are two statistical techniques that makes use of the fundamental technique of PCA. These are mainly used in Part of Speech tagging [31]. The HMM is a probabilistic sequence model which assigns a label or class to each unit (e.g.: words, letters, morphemes, sentences) in a sequence, thereby mapping a sequence of observations to a sequence of labels. HMM compute a probability distribution for a possible sequence of labels and then it chooses the best label sequence for a given sequence of units [32]. LSI method transforms the text space to a new axis system which is a linear combination of the original word features. It determines the axis-system which retains the greatest level of information about the variations in the underlying attribute values [2]. Latent Dirichlet Allocation (LDA) is another model similar to LSI that overcomes some of the drawbacks in LSI. Duric and Song [30] in their study used a combination of HMM and LDA which simultaneously models topics and syntactic structures in a collection of documents and were able to achieve competitive results for document polarity classification.

2.3.3 Pointwise Mutual Information

The mutual information measure allows to model the mutual information between the features and the classes. The pointwise mutual information (PMI) Mi(w) between the word w and the class i is determined by the level of co-occurrence between the class i and word w.

$$M_i(w) = \log\left(\frac{F(w) \cdot p_i(w)}{F(w) \cdot P_i}\right) = \log\left(\frac{p_i(w)}{P_i}\right)$$

Equation 2.1: Pointwise Mutual Information

If the Mi(w) is greater than 0, then the word w is positively correlated to the class i, else negatively related [2].

2.3.4 Chi-square

Chi-square (χ2) of the word between word w and class i is defined as follows when, pi(w) is the conditional probability of class i for documents which contain w, Pi is the fraction of all documents containing the class i, and F(w) is the global fraction of documents which contain the word w.

$$\chi_i^2 = \frac{n \cdot F(w)^2 \cdot (p_i(w) - P_i)^2}{F(w) \cdot (1 - F(w)) \cdot P_i \cdot (1 - P_i)}$$

Equation 2.2: Chi-Square

χ2 is better than PMI because it is a normalized value. Therefore, these values are more comparable across terms in the same category [2]. It was observed that using Chi-square in feature selection reduced overfitting and increased the accuracy of the model in various NLP applications [2], [21].

2.4 Sentiment Classification

Sentiment analysis, which is a classification problem, is the process of understanding human emotions expressed in a text. Research on sentiment analysis can be traced back to 1960s. Previous related work on sentiment analysis can be classified as below in Figure 2.2 [2]. Most of these techniques can be applied to all sentiment, emotional and attitude classifications.

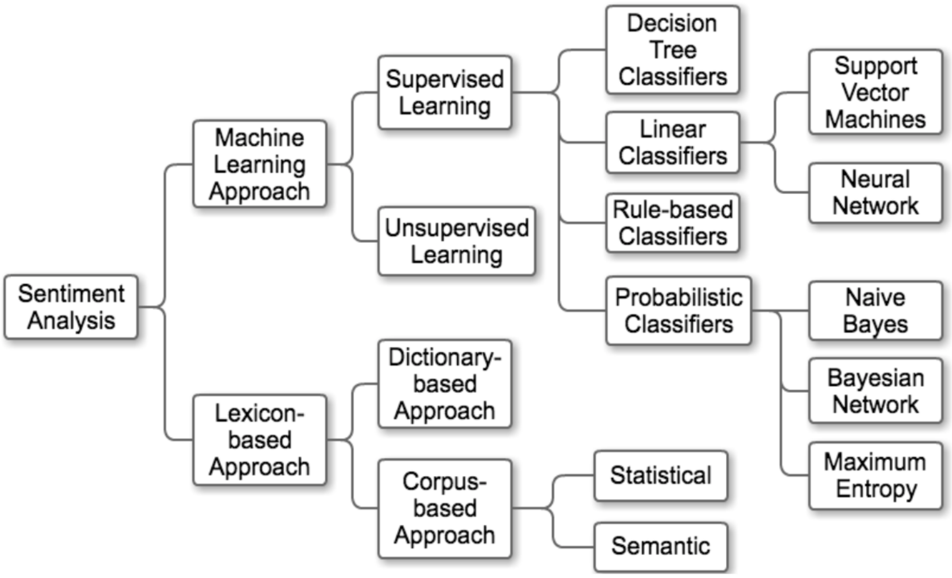


Figure 2.2: Classification of SA techniques

Source: [2]

2.4.1 Lexicon Based Approach (LB)

Lexicon based approach in SA involves the extraction of terms' polarities from labelled sentiment lexicons and the aggregation of such scores to predict the overall sentiment of the given piece of text. The sentiment lexicon can be either manually or semi-automatically generated. Obviously manual approach is more accurate but it consumes more time and has low term coverage. Whereas semi-automatic approach has a high term coverage and less compile time [31]. Dictionary based and Corpus based approaches are a few examples of semi-automatic approach of LB.

2.4.1.1 Dictionary Based Approach

The Dictionary based approach is carried out by a small set of opinion words collected manually with their known polarity. This set is ingrown by searching the words in a thesaurus for synonyms. The new set of words are added to the list and the search is iterated again. This is iterated until no new words are found then can be manually evaluated for accuracy [2]. Dictionary based approach has an advantage over ML because it doesn't require prior training [33]. The disadvantage of the Dictionary based approach is that its inability to find opinion words only related to a given domain [2]. Hence, this is more suitable for generalized contexts with less domain specific words.

2.4.1.2 Corpus Based Approach

Corpus based approach which comes under the Lexicon based (LB) method is more suitable in domain specific context. In Corpus based approach the probability of occurrence of a sentiment word in conjunction with positive or negative set of words (corpus) is determined. Words connected using stop words such as AND, OR, EITHER-OR are considered to be similar whereas words connected using stop words such as BUT are assumed to give opposite opinions [2].

One of the drawbacks is domain specific model is that its inability to be used in another domain. Moreno-Ortiz and Fernández-Cruz [34] proposed “plug-in” approach where the users can assign an appropriate corpus for their method so that the corpus based approach can be used across domains. Compared to manual and dictionary based approaches in LB, corpus-based methods can produce opinion words with relatively high accuracy [1].

LB algorithms robustness across the domains is low which is a significant drawback [31]. SmartSA [31], a LB contextual sentiment analysis for social media, integrates strategies to capture contextual polarity from two perspectives: the interaction of terms with their textual neighborhood (local context) and text genre (global context). Their approach which hybridised a general purpose lexicon with genre-specific vocabulary and sentiment was able to outperform a modern lexicon-based classifier, SentiStrength as well as a machine learning classification.

2.4.2 Machine Learning Approach (ML)

With the current trends and improvements in the field of ML, ML classification techniques are widely being used in sentiment classification and analysis. This approach considers the problem as a regular text classification problem where a set of training records are given to build a model that will predict the label for unseen new data. Numerous studies have been done in the field of ML, but in this study, we will only explore research relevant to SC. Machine Learning can be categorized into two approaches as supervised and unsupervised learning.

2.4.2.1 Supervised Learning

In supervised machine learning the training sample has labelled data. The model will be created using this data and the unseen data will be classified based on the learned patterns of the known labelled data. Supervised ML algorithms, which is a popular method in SA have been tested on SA and have shown satisfactory results. Naive Bayes classifier, Maximum entropy, Support Vector Machines, Rule based classification, Neural networks are some of such techniques.

A. Naive Bayes

Naive Bayes classifier and Bayes networks are probabilistic classifiers in ML. The Naive Bayes classifier which computes the posterior probability is the simplest and most commonly used classifier. This works on the bag of words constructed in the FS phase. This classifier predicts the label of each word in the set by determining its posterior probability of belonging to that class using the training set. Even though is a simple and easy to implement approach this a very fundamental classifier from which satisfiable accuracy cannot be expected. Multinomial Naive Bayes classifier is commonly used in most of the SM SA problems [10], [16]. It is an instance of a Naive Bayes classifier which uses a multinomial distribution for each of the features.

Naive Bayes is based on the assumption that the features are independent, which is usually not the case in the real world problems. Bayes Network has no such assumptions. It is modelled by a directed acyclic graph where edges represent individual conditional dependencies. Bayes Networks technique is not generally used in SA due to its computational complexities [2].

B. Maximum Entropy (ME)

In [35], the results showed that Maximum Entropy has even out performed Support Vector Machines— a linear classifier (explained in the next section), in supervised sentiment analysis conducted on Czech social media.

Maximum entropy is another probabilistic ML algorithm used in SC that has better performance than both the above mentioned approaches according to [1] and [35]. The principle behind Maximum Entropy is to find the best probability distribution among prior test data [1]. This classifier takes set of X{weights} as parameters These are then used to combine the joint features that are generated from a feature-set using X{encoding}. The encoding maps each C{(featureset, label)} pair to a vector. The probability of each label is then computed using the following equation [2]:

$$P(fs|label) = \frac{dotprod(weights, encode(fs, label))}{sum(dotprod(weights, encode(fs, l))forlinlabels)}$$

Equation 2.3: Pobability Calculation of Maximum Entropy

C. Support Vector Machines (SVM)

In [36], the Support Vector Machine (SVM) was used along with χ2 and PMI as FS methods in microblogging multi-class sentiment classification and was found significantly effective. SVM is a linear classifier which is widely used in SC. The main principle of SVMs is to determine linear separators in the search space which can best separate the different classes. The reason why SVM is commonly used in SC is that the text data are ideally suited for SVM classification because of the sparse nature of text, in which few features are irrelevant, but they tend to be correlated with one another and generally organizable into linearly separable categories [2].

D. Rule Based Classification

In rule based classifiers, the data space is modelled with a set of rules (IF-THEN rules) [1]. VADER introduced in [37] is a Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text. VADER consists of five general rules that embody grammatical and syntactical conventions for expressing and emphasizing sentiment intensity. It yielded significant results outperforming eleven other highly regarded sentiment analysis tools including Naive Bayes, Maximum Entropy, and Support Vector Machine (SVM) and a few lexicon based approaches [37].

E. Neural Networks (NN)

Neural Networks is another linear classifier. It is a model that is based on a collection of natural/artificial neurons (where the neuron is its basic unit) uses for mathematical and computational model analysis [1]. The inputs to the neurons are denoted by the vector $\bar{X}i$ which is the word frequencies in the ith document. There is a set of weights 'A' which are associated with each neuron used in order to compute a function of its inputs f(●). The linear function of the neural network is: $pi = A \cdot \bar{X}i$ [2]. Recently, Neural Networks is becoming a popular area of research in the field of SA due to its advancement [38]. In, [38], they studied whether NN can be combined with other shallow statistical machine learning approaches to build a Probabilistic Neural Network (PNN). Authors came up with 2 models PNN to classify Twitter data and claim to have achieved 92% and 95% accuracy rates which shows significant performances.

Deep Learning (DL) which is a new area of Machine Learning is a type of NN. Simply, DL is NN with several stages of layers [39]. Recurrent Neural Networks (RNN) along with Long Short-Term Memory has been able to solve previously unlearnable tasks by using recognising the temporal order in sequences. LSTM has outperformed other RNN models in natural language processing that requires learning language rules and also in tasks involving context free languages [39]. In [40], the authors attempted to solve the problem of limited contextual information in short messages such as Tweets using DL. They proposed a new deep convolutional neural network that exploits from character-to-sentence-level information and were able to achieve an accuracy of 86.4%

for Stanford Twitter Sentiment corpus. Additionally, DL can be used to address the problem of domain adaptation for models [25]. “Coooolll” is another DL that is built in a supervised learning framework by concatenating the sentiment specific word embedding (SSWE) features with the state-of- the-art hand-crafted features for Twitter sentiment classification that was ranked the 2nd on the Twitter2014 test set of SemEval 2014 Task 9 [35].

2.4.2.2 Unsupervised Learning

Unsupervised learning approach and lexicon based approach were combined in [41] and was able to outperform other machine learning solutions in the majority of cases. Unsupervised learning takes away the burden of labelling the data. Given a set of sentences the unsupervised learning will cluster them based on the similarity of the key words. One of the key advantages of unsupervised learning is its robustness and the ability to use across domains.

2.4.3 Ensemble Learning (EL)

Even though numerous studies have been conducted on SA, there is no single model that all the research has agreed upon as the best performing model for all SA problems. Standalone single classification models can have their own drawbacks. In ensemble learning method, different classification models are trained and combined to solve the same problem. These classification models are known as “base learners” and they can be of the same classification algorithm [42], [43] or combination of different classification algorithms [44], [45], [46]. Most of these studies show that ensemble classifiers out performed single classifiers in SA. However, research contributing to the use of ensemble learning in the domain of SA is comparatively limited and has opportunities for more extensive experiments [42].

EL method can be divided into two categories namely, instance partitioning and feature partitioning [42]. In instance partitioning the training sample is partitioned whereas in the feature partitioning method the feature set is partitioned. Bagging, Boosting and Stacking belong to the instance partitioning method and Random Subspace belongs to the feature partitioning method.

- *Bagging*- The classifier combination strategy is through some averaging process (usually the majority vote) [42].
- *Boosting*- This is an iterative learning process with a weighted combination strategy where the weights are determined are during the iterations based on the accuracies of the base learners [42].
- *Stacking*- This method forms a high level meta classifier combining the results of the base learners [44].
- *Random Subspace*- The combination strategy of this method is similar to Bagging but instead of different datasets, it uses different feature sets [42].

Application of EL in SA for SM has been underexplored in the literature [47], [48]. A few exceptions are [45], [48] and [47]. In [47], AdaBoost, a boosting method was compared against standalone SVM and Multinomial Naive Bayes (MNB) classifier in Twitter SA problem. AdaBoost classifier with MNB base learners showed the best results in this experiment. In another Twitter SA where MNB, Random Forest and SVM were combined through the bagging method also produced better results than the standalone classifiers [48]. Interestingly, a LB (SentiStrength) and ML (SVM) methods were combined in [45] using the bagging method and it outperformed standalone classifiers as well as several other ensemble classifiers in Twitter SA problem. Due to the lack of literature on EL applied on SM SA, some research conducted on application of EL in general SA was studied. Out of boosting, bagging and random subspace, Wang et al. [42] observed that random subspace gave the best accuracy. Whitehead and Yaeger [43] who observed similar results explained that since sentiment analysis problems usually have thousands of features, random subspace which is a feature partitioning method is more suitable to address this problem. In [44], which compared stacking method against bagging in SC observed that stacking outperformed bagging. Catal and Nangir [46] combined several classification algorithms as base learners using the bagging method and observed that it yielded better results than the ensemble classifier with only SVM base learners for social media sentiment analysis. Ensemble methods give high accuracy at the expense of computational time. But this computational expense is only during the training phase [43].

2.5 Evaluation

Confusion matrix shows the counts of actual vs predicted instances for each class. Table 2.1 shows a confusion matrix for a binary classification.

Table 2.1: Confusion Matrix

		Actual	
		Positive	**Negative**
Predicted	**Positive**	True Positive (TP)	False Positive (FP)
	Negative	False Negative (FN)	True Negative (TN)

Source: Adapted from [42]

Most common SA evaluation technique used in the literature is the accuracy which is the percentage of correctly classified instances [42]. Some studies have also employed metrics such as recall and precision. As defined Nasukawa and Yi [49], "Precision is the ratio of correct cases within the system outputs. Recall is the ratio of correct cases that the system assigned compared to the base of all cases where a human analyst associated either positive or negative sentiments manually". Recall, precision and accuracy can be calculated using the confusion matrix as follows.

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

Equation 2.4: Accuracy

$$Precision\ (Positive\ Class) = \frac{TP}{TP + FP}$$

Equation 2.5: Precision

$$Recall\ (Positive\ Class) = \frac{TP}{TP + FN}$$

Equation 2.6: Recall

Furthermore, Sokolova and Lapalme [50], after comparing 24 performance measures, proposed that for sentiment classification problems where the datasets are imbalanced, F-score measure is a good evaluation metric. F-score is a weighted average of precision and recall.

$$F - score = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

Equation 2.7: F1 Score

2.6 Types of Sentiment Classification

It was studied that the most common type of sentiment classification available on literature is using the sentiment polarity (i.e. positive and negative) [30], [51]. Some studies have improved this by one additional level by adding “Neutral” class. When there is neither positive nor negative sentiment expressed in the text, it is categorised as “Neutral”. Furthermore, in the studies [24] and [52], the authors have added two additional classes “Bipolar” and “Off-Topic” respectively. In some cases, a single text can contain both positive and negative sentiments, “Bipolar” class was suggested in [24] to categorise such text. It was observed that even though the queried keyword (target object) appears in the text, it might not necessarily refer to the exact topic of interest. For an example, if the objective is to analyse the sentiment towards Apple products, one may use the keyword “Apple” to filter out text. But this will also fetch texts addressed towards “apple” fruit. To address such cases, Ceron et. al. [52] suggested the class “Off-Topic” that will reduce noise in the data. Classification based on polarity can be introduced as the most basic level of sentiment analysis.

Nevertheless, in order to thoroughly understand the sentiment of a user, one would need to understand beyond the mere polarities [53]. Some studies have shown interest to investigate beyond the binary polarity. Bouazizi and Ohtsuki [53], used sentiment classification based on emotions. In this research the authors pointed out the difference between the following two negative tweets.

1. “Damn damn.. no iPhone support for windows XP x64. There are some workarounds, but I can’t figure this out.”
2. “Noooooooooooo! My iPhone glass cracked :(”

The first tweet expresses fury towards the product due to some limitations, whereas the second tweet expresses a feeling of sadness caused due to some damage faced to the product. In a business perspective the first tweet is more useful than the second as it expresses a customer dissatisfaction due to a product limitation. Hence, the authors bring forward the importance of clearly distinguishing between such cases and using a more granular level to classify the sentiment texts. The list of emotions they used in the study are, “happiness”, “sadness”, “anger”, “love”, “hate”, “sarcasm” and “neutral”. Similarly, Balabantaray et al. [54] used the 6 emotions proposed by Ekman [56] namely, “happiness”, “sadness”, “anger”, “disgust”, “surprise” and “fear” in their sentiment classification study. Ekman model consists of distinctly identifiable facial expressions of emotions and it is widely being used in many emotion detection problems including sentiment analysis.

Almashraee et. al. [55] discussed two different categories of sentiments, namely, rational and emotional. Rational sentiment describes reasoning and beliefs (e.g.: “This camera is good.”) Whereas the emotional sentiment describes a psychological state of mind (e.g.: “I trust this camera”). The authors studied that the majority of social media users use rational words to express opinions. In order to identify the underlying emotion of these texts, they computed a similarity score between rational words and emotional words using Plutchik’s Wheel of Emotions. Plutchik’s Wheel of Emotions introduced by the psychologist Robert Plutchik [57] is another ontology used in SA to categorise the sentiments. Plutchik defined eight basic emotions “joy”, “trust”, “fear”, “surprise”, “sadness”, “anticipation”, “anger”, and “disgust”. As shown in Figure 2.3, the wheel depicts the relations with each of these emotions and how the intensity increases as you move towards the center of the wheel [55].

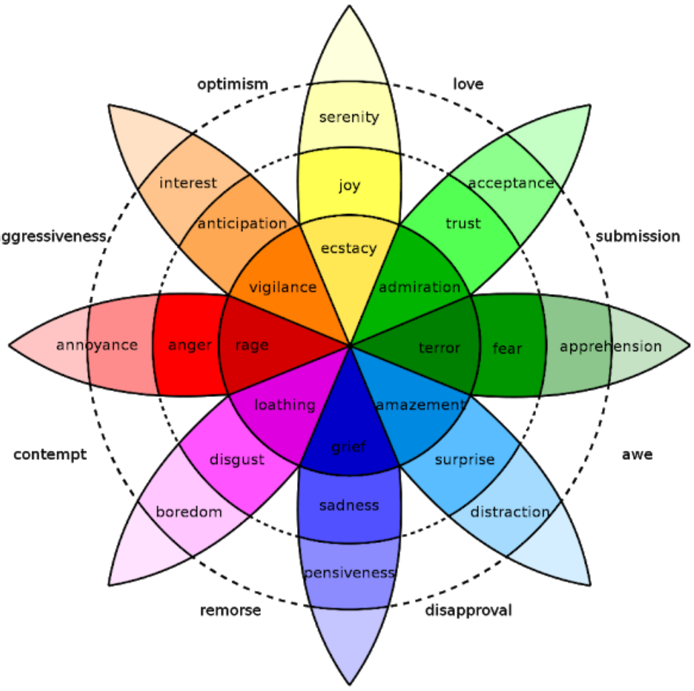


Figure 2.3: Plutchik’s Wheel of Emotions

Bollen et al. [58] used a well established psychometric instrument, the Profile of Mood States (POMS) to classify the sentiment of the tweets. POMS which was introduced by McNair, Loor, and Droppleman [59], measures 6 types of moods namely, Tension, Depression, Anger, Vigour, Fatigue, and Confusion. Traditionally, POMS is not a model used for textual analysis. Rather it is a standard validated psychological test used in research to assess mood states. The questionnaire is composed of 65 adjectives and users are asked to indicate on a 5 point scale how well each of these 65 emotions describe their current emotion. Then the ratings are transformed into the 6-dimensional mood vector. Authors used these standard adjectives to categorise tweets for sentiment analysis in their study.

However, it was observed that there is a lack of literature in the area of classifying beyond the polarity. Nevertheless, among the existing types of sentiment classification ontologies, there is no unique ontology that works for all domains, due the different nature of the domains. Due to the complexity of emotions and large number of emotions, theorists have introduced several emotion models. Hence, we cannot state that one is better than the other. Despite the type emotion model selected, suitable methods of collections, representation, organizations and mapping needs to be decided. Most of the experiments are manually annotated based on specific emotion model or only verbs and nouns are considered as opinion bearing words [55]. Most of these types of classification models have addressed the emotion aspect of the sentiment. Whereas the objective of this research is to address the attitude aspect of the sentiment.

2.7 Applications of ABC Model of Attitudes

Any existing literature on applying ABC Model of attitudes to analyse the sentiment of text couldn’t be found. But ABC Model of Attitudes has been widely applied in the domain of Consumer Psychology and Socialology. In this section, studies that have adapted the ABC attitude model in these domains will be discussed.

This model of attitudes is often used in the domain of consumer psychology to study the buying behaviors, intentions and attitudes of customers towards different products. Sandhe and Joshi [60] used the ABC model to study the consumers' attitude towards organic food products. They identified variables for each attitude component. A few of such identified variables are as follows:

Affective-> Appealing, Generally Good, Preferred over Non-Organic Food Products

Behavioral-> Definitely Buy, Recommend Others, Increase Spending

Cognitive-> Healthy Option, Safe to Consume, Value for Money

Chen and Chen [61] used ABC model of attitudes to study how online and offline behavior processes affect each other in Click-and-mortar business. They regarded these attitudes as a sequence order of formation. In their research context the steps involving the use of Click-and-mortar channels are, learning about the benefits (cognitive), satisfaction with the service (affective) and loyalty of the customers towards the service (behaviour). This study assumed that the system quality and information quality provided by the website determines the cognitive aspect of the customers. The quality of the system versus the customer’s expectation determines the affective attitude. Then these will result in the customer’s behavioural attitude where the customer may or may not develop a strong commitment to repurchase and to avoid alternative offers. Asiegbu et al. [62], in their study reflects that attitudes are primary causes of buying behaviours. They showed that the customer buying behavior can be influenced by altering any of three attitude components and discussed different strategies to alter each of these attitude components. They further discussed the importance of understanding attitudes using the ABC model for the marketers in order to understand why consumers buy or do not buy their products and thereby adapting suitable marketing strategies. In another study, ABC model of attitudes was employed to explore the attitudes of the public toward water-saving equipment, where they studied which attitude type mainly influence the acceptance of the public toward using water-saving equipment to understand what aspects should be given priority when designing those equipment [63].

Zhu and Xu [64] conducted a study in the domain of social psychology to investigate the Hainan (a city in China) residents' attitudes towards migratory groups who come for vacation, pension and work purposes. The ABC model of attitudes was employed to study the attitude of residents. In this context, the affective component was defined by whether the residents liked or disliked the arrival of migratory groups, the behavioural component was defined by how welcoming the resident were towards the migrants and cognitive component was defined by the residents' evaluation of quality and impact of the migrants have towards factors such as security, public resources and housing market.

In another study of ambivalence [65], authors used the ABC attitude model to investigate its consequences beyond the polarity of positive and negative. Ambivalence is a state when a person holds mixed feelings towards a objects or topics such as abortion, eating meat and drugs. In this study authors state that ambivalence can cause affective, cognitive and behavioral consequences.

Most of classification tasks go through the steps of preprocessing, feature extraction, classification and evaluation. In this chapter, the tool and techniques used by the previous studies of similar problems in sentiment analysis to implement these steps were discussed. How those techniques have influenced the final solutions was closely studied. Additionally, in this chapter different models of sentiments classifications used by researchers was studied. It was observed that many studies have been conducted to understand the human emotions but less focus has been given to understand the human attitude. Even though ABC attitude model was not applied in a sentiment analysis task before, this attitude model has been widely applied in other domains such as consumer psychology and sociology. A few of such studies were also discussed in this chapter.

3. RESEARCH METHODOLOGY

The main objective of this study is to investigate sentiment classification beyond mere polarity. For this, the author has selected the well established Tricomponent Attitude Model (ABC model) introduced in consumer psychology. At the end of this research using the knowledge acquired from previous studies, the final output shall be a model capable of correctly classifying the social media short messages into Affective, Behavioural or Cognitive classes based on a selected set of tools. This chapter will discuss the data collection methods, implementation and justification of the technologies used to achieve the research objective.

3.1 Solution Architecture

The Figure 2.1 studied in Chapter 2, depicts the general process flow of solving a classification task. The same model was employed in this research. Before implementing the solution, the first and foremost task is obtaining a dataset to be used in building and evaluating the model. The data source selected for this experiment is Twitter. The extracted dataset was manually annotated by several annotators. Exploratory data analysis (EDA) was conducted in order to get a sound understanding of the dataset at hand and to plan out the rest of the implementation techniques. Several feature extracting techniques such as bag-of-words term frequency, bigram term frequency, TF-IDF, POS tagged term presence and Word2Vec were used. The performance of each of the preprocessing techniques and feature extraction methods were evaluated against the classification algorithms such as Logistic Regression, SVM, Multinomial Naive Bayes, Ensemble classifiers and Deep learning models that were employed in this study.

3.2 Data Collection

Since there wasn’t any existing dataset that is suitable for this research, it was decided to come up with a new dataset. For this research Twitter social media platform was selected as the data source. It was selected due to the following reasons:

- It is a popular social media platform used to express personal opinions and views about products, organizations and brands.
- Character limit of 280 makes processing easier and gives a more uniform length of content across the data set.
- A global platform with 262 million active users and 6,000 tweets every second giving access to engage with opinions of a diverse user base.[1]
- Availability of developer APIs and other open source libraries that allow to extract tweets easily and freely with minimum limitations.
- Reports show that roughly 40% users have admitted that they would make a purchasing decision based on an influencer's tweet. This shows the importance of analyzing the sentiments expressed in tweets.[1]

The attitude object selected for this experiment is "iPhone". The iPhone was selected considering its popularity as a brand and also because people express their positive and negative experiences on such products on Twitter. They may "tweet" about the product to open discussions, to get help, to inform about features or defects, or to merely express their sentiment towards the product. The SA model is expected to classify tweets into 3 classes namely- Affective, Behavioural and Cognitive. After studying the data sets used in the previous research it was decided to obtain approximately 1000 tweets for each class. The duration which was selected for data collection is between 2/12/2019- 15/12/2019 which is over a time period of 2 weeks. The reason for selecting this week is because it is the most recent week as of the day of data collection and it has been some time since the latest iPhone has been released therefore it allows users to express their emotions and opinions based on actual experience on the new releases.

[1] https://learn.g2.com/twitter-statistics

3.2.1 Data Extraction

There are several methods such as APIs, libraries and scrapers to extract tweets. Most of the APIs are made available by Twitter whereas libraries and scrapers are free open source tools. For this research following tools were assessed.

- Twitter API
- Tweepy
- TwitterScraper

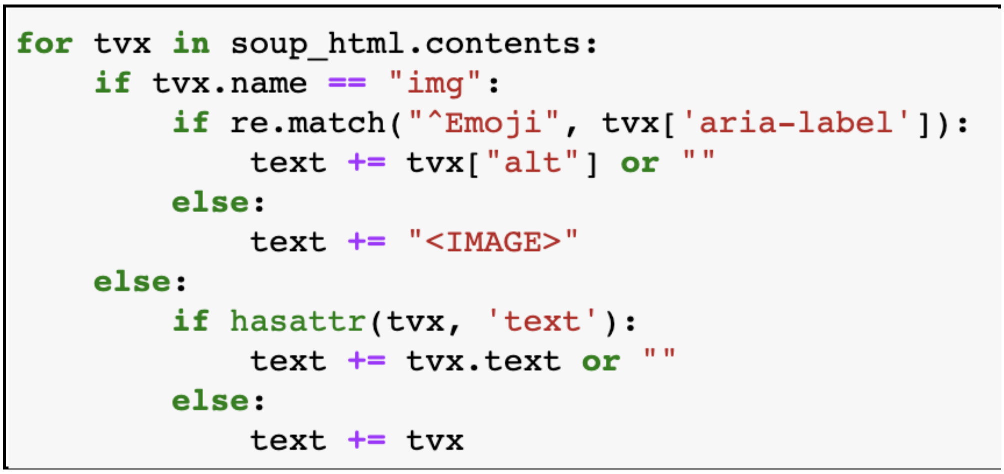
Twitter has made available two APIs that are suitable to extract tweets for a given query term. They are, Search API[2] and Historical PowerTrack API[3]. Historical PowerTrack API is a paid enterprise level API, therefore it wasn't considered for this research. But the search API provides several options such as the standard, enterprise and premium, out of which standard version is free. It allows the users to extract a specific number of tweets for a given query term. Out of several libraries available to extract tweets, Tweepy[4] was one of the most sought after tools. Tweepy is a Python interface that makes interacting with the Twitter API easier. It allows one to access entire twitter RESTful API methods. Both of these methods require setting up a developer account and an app to get authentication tokens. The disadvantage of these methods is the limitations of using the standard free version. Some of such limitations include the 7-day limit where no tweets can be extracted for a date older than one week and the limitation on the number of API calls for a given time.

On the other hand, TwitterScraper[5] developed by Ahmet Taspinar is a simple web scraper that is built specifically for extracting tweets. It overcomes most of the limitations of using the Twitter search API such as extracting data older than one week. TwitterScraper allows extracting tweets posted during any period of duration specified

[2] https://developer.twitter.com/en/docs/tweets/search/api-reference/get-search-tweets
[3] https://developer.twitter.com/en/docs/tweets/batch-historical/api-reference/historical-powertrack
[4] https://tweepy.org
[5] https://github.com/taspinar/twitterscraper

by the user without any authentication tokens. The disadvantage of using a web scraper is that they may not be supported for the long term if the target website changes its HTML code. But since data collection for this experiment is a one time thing this limitation could be ignored. After comparing the three tools for extracting tweets, TwitterScraper was selected considering its ease of use.

But it was observed that the extracted text did not contain any emojis. Emojis are rendered as images in the Twitter page therefore they were inside image tags which were not extracted during the process. Emojis have proved to provide important information on sentiments in the previous research. Hence the scraping code was modified as shown in Figure 3.1 to cater this requirement. In the Twitter webpage, emoji images have an "aria-label" attribute which starts with "Emoji" and the "alt" attribute has the actual emoji Unicode character.



```
for tvx in soup_html.contents:
    if tvx.name == "img":
        if re.match("^Emoji", tvx['aria-label']):
            text += tvx["alt"] or ""
        else:
            text += "<IMAGE>"
    else:
        if hasattr(tvx, 'text'):
            text += tvx.text or ""
        else:
            text += tvx
```

Figure 3.1: Code Snippet to Include Emojis in Text

As shown in the Figure 3.1, in this method, tweet body HTML markup is parsed and all image tags which are emoji images (identified through the aria-label match) are scanned. Then the image Unicode character is extracted to get the emoji name. Other images are tagged with "<IMAGE>" placeholder text and text value are extracted for all other elements.

It was observed that greater percentage tweets contain non-sentiment tweets such as questions and advertisements. Therefore, the limit was set to 14,000 keeping some buffer for such tweets. The scraper extracted 14,067 tweets within 1 minute of execution. The extracted data set contained the following attributes:

“screen_name”, “username”, “user_id”, “tweet_id”, “tweet_url”, “timestamp”, “timestamp_epochs”, “text”, “text_html”, “links”, “hashtags”, “has_media”, “img_urls”, “video_url”, “likes”, “retweets”, “replies”, “is_replied”, “is_reply_to”, “parent_tweet_id”, “reply_to_users”

The scraper doesn’t extract retweets but it was observed that some duplicate tweets have been extracted during the process. Columns “timestamp_epochs” and “text” were used to filter out duplicate tweets. A dataset of 10,000 tweets was exported into a CSV file after extracting the unique tweets. For this study, only the columns “tweet_id” and “text” were used. Therefore, tweets were later extracted to another file with the selected columns, “tweet_id” and “text” for data annotation.

3.2.2 Data Annotation

Since the supervised machine learning approach was selected for this study, the extracted dataset needed to be labelled. The dataset was broken into 10 CSV files with 1000 tweets each for processing and evaluating convenience. The annotation was done by the author. In order to calculate the inter-rater agreement, to evaluate the dataset, a psychology expert annotated a sample of 1000 tweets from the dataset. Evaluation of inter annotator agreement will be discussed in the Chapter 4. Even though for the current research annotation based on the ABC attitude model was sufficient, tweets were also annotated based on the sentiment polarity that can be used for future improvements. Below are the set of labels used for annotation:

Attitude Labels:

Affective- A
Behavioural- B
Cognitive- C
Unknown/ Neutral/ Conflicting- U
Advertisement- V

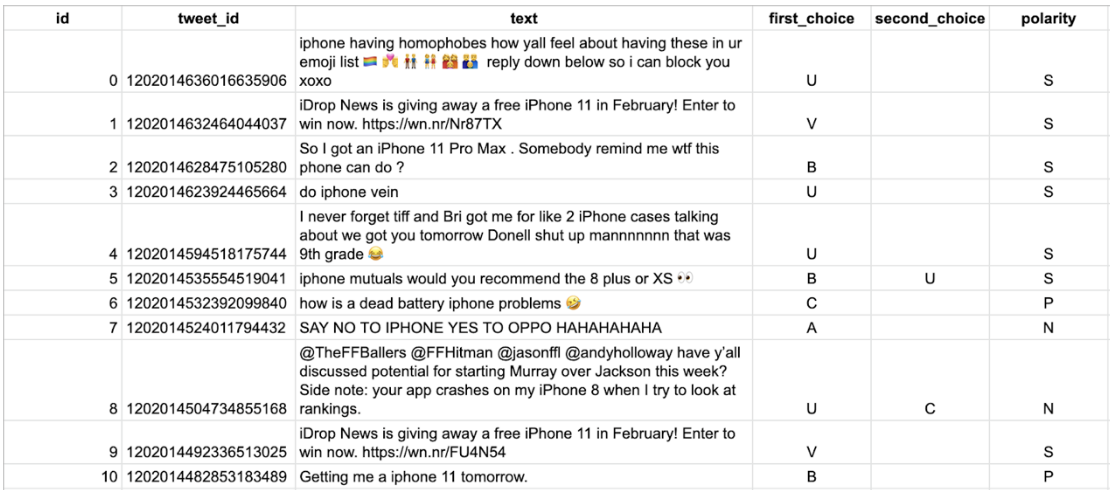
Polarity labels:

Positive- P

Negative- N

Neutral- S

A tweet will belong only to one label. But the annotators were given the option of “second_choice” when they cannot decide between 2 labels. Shown below in Figure 3.2 is a sample of an annotated dataset.



id	tweet_id	text	first_choice	second_choice	polarity
0	1202014636016635906	iphone having homophobes how yall feel about having these in ur emoji list 🏳️‍🌈 💏 👬 👭 💑 👪 reply down below so i can block you xoxo	U		S
1	1202014632464044037	iDrop News is giving away a free iPhone 11 in February! Enter to win now. https://wn.nr/Nr87TX	V		S
2	1202014628475105280	So I got an iPhone 11 Pro Max . Somebody remind me wtf this phone can do ?	B		S
3	1202014623924465664	do iphone vein	U		S
4	1202014594518175744	I never forget tiff and Bri got me for like 2 iPhone cases talking about we got you tomorrow Donell shut up mannnnnnn that was 9th grade 😂	U		S
5	1202014535554519041	iphone mutuals would you recommend the 8 plus or XS 👀	B	U	S
6	1202014532392099840	how is a dead battery iphone problems 🤣	C		P
7	1202014524011794432	SAY NO TO IPHONE YES TO OPPO HAHAHAHAHA	A		N
8	1202014504734855168	@TheFFBallers @FFHitman @jasonffl @andyholloway have y'all discussed potential for starting Murray over Jackson this week? Side note: your app crashes on my iPhone 8 when I try to look at rankings.	U	C	N
9	1202014492336513025	iDrop News is giving away a free iPhone 11 in February! Enter to win now. https://wn.nr/FU4N54	V		S
10	1202014482853183489	Getting me a iphone 11 tomorrow.	B		P

Figure 3.2: Sample of annotated dataset

3.3 Exploratory Data Analysis (EDA)

After the data annotation has been done, EDA was conducted to understand the dataset at hand. It is important to know the nature of data and its distributions for making design decisions further into the study.

First of all, the class distribution was analysed as shown below in Table 3.1.

Table 3.1: Categorization and Distribution of Classes

	Class	Count
Attitudinal	Affective (A)	1103
	Behavioral (B)	1848
	Cognitive (C)	2090
Non-Attitudinal	Neutral/ Conflicting/ Unknown (U)	4148
	Advertisements (V)	811

The dataset can be divided into attitudinal and non-attitudinal groups. Attitudinal classes are the ones with attitudes, i.e. A, B, C. Whereas the non-attitudinal classes are the ones containing neutral tweets (U) and advertisements (A). As shown in the pie chart below in Figure 3.3, attitudinal and non-attitudinal classes split the dataset into almost equal halves.

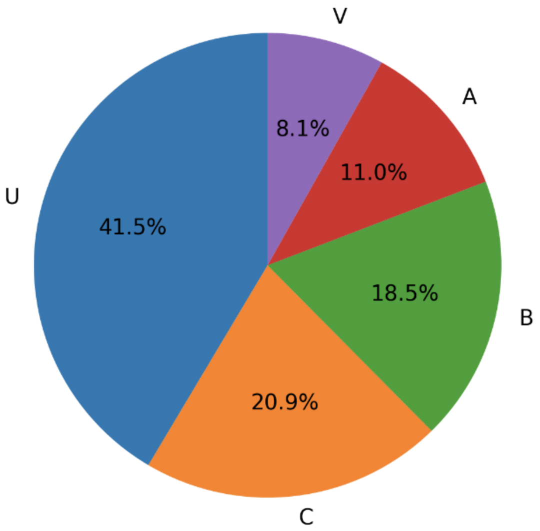


Figure 3.3: Distribution of Classes

For this research since we will be employing the ABC model of attitude, only the attitudinal classes will be used as non-attitudinal types are not a part of this model. Therefore, the rest of the EDA will be conducted only on the attitudinal dataset.

The below graph in Figure 3.4 shows the distribution of the length of the tweets. Twitter has a limited character count. Therefore, it is visible that the maximum length (or word count) for this twitter dataset has been limited to 60. Most of the tweets are of length below 20. The length does not have significant variance. This uniformity of length is one of the reasons for selecting Twitter as the source of data for this experiment. The same experiment was carried out for each class and similar distribution was observed therefore they were not included in the report.

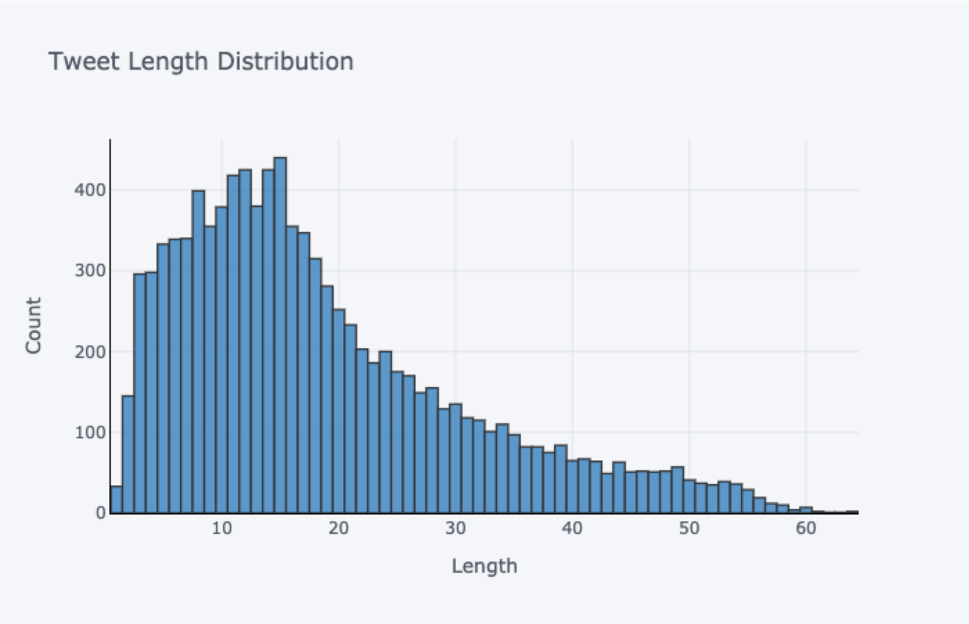


Figure 3.4: Length Distribution of Tweets

The graph in Figure 3.5 shows the frequency distribution of most occurring top 50 words in the dataset. It is evident that most of the frequently occurring words are common words such as stop words and punctuations. This shows the importance of preprocessing the text to remove such noise.

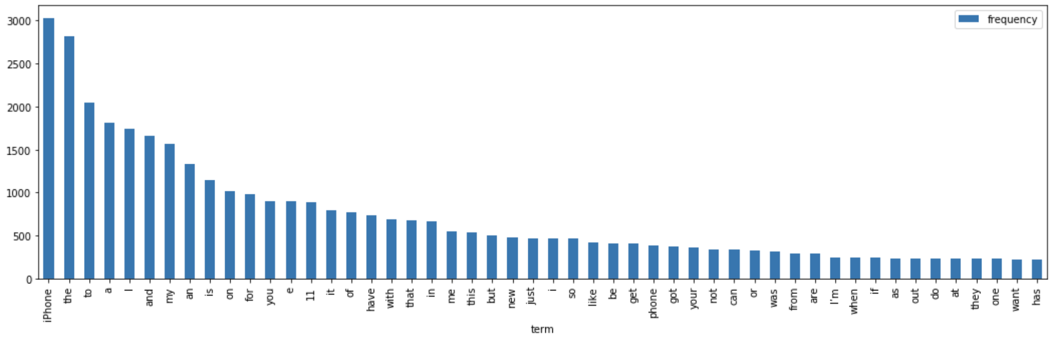


Figure 3.5: Top 50 Most frequent terms before preprocessing

The graph in Figure 3.6 shows the most occurring 50 terms after preprocessing. All the preprocessing techniques that were selected for this experiment have been applied except the spell correction in order to view the terms as they are. All the preprocessing techniques will be discussed in Section 3.4.

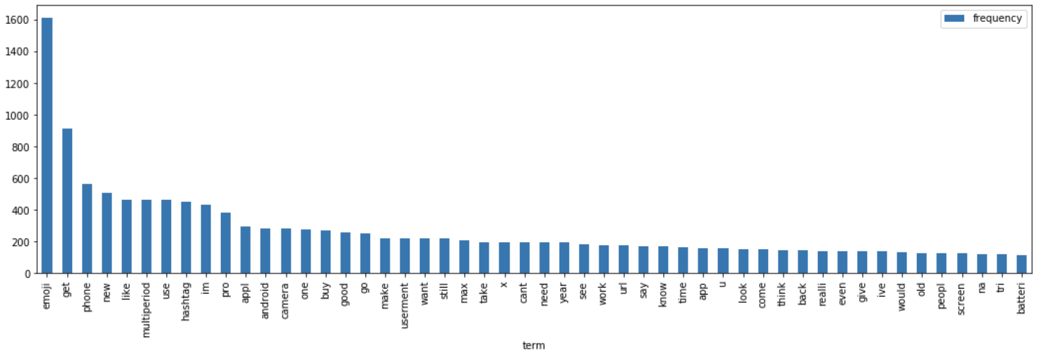


Figure 3.6: Top 50 Most frequent terms after preprocessing

It is interesting to investigate which terms are prominent in each class as these will play a major role in the classification task. Following graphs were generated for the top 50 most occurring terms in each class. Before generating the graphs, Twitter tags such as hashtags, user mentions, images and URLs have been replaced with tags such as “hash_tags”, “user_mentions”, “image”, “url” to cleanse the dataset. The text data has also been converted to lowercase. None of the other preprocessing techniques other than these two have been applied. But these graphs give some insight on the nature of the terms and the prominent words used to express different types of attitudes. It can be observed that the nature of the terms of these graphs are very different to one another.

The graph in Figure 3.7 depicts the most occurring words in the Affective attitude class. As discussed in the previous section, the affective class contains varying types of text as this type of attitude is often expressed in association with another object entity. This observation is clearly visible in this distribution graph. It seems that the attitude expressed towards “iPhone” is often associated with other attitude objects such as “person” or “android”. Because generally people express the affective attitudes towards an entity in contrast to their counterparts, which is Android in this scenario. Similarly, humans tend to associate positive or negative attitudes towards the entity with the attitude they already have towards the people who are associated with that entity. Nevertheless, among these words it can also be observed that there are a few strong emotion-expressing words such as “love”, “like” and “hate”. Furthermore, this also shows that repeated punctuation marks (denoted by “multiperiod”) are strong indicators of affective attitude.

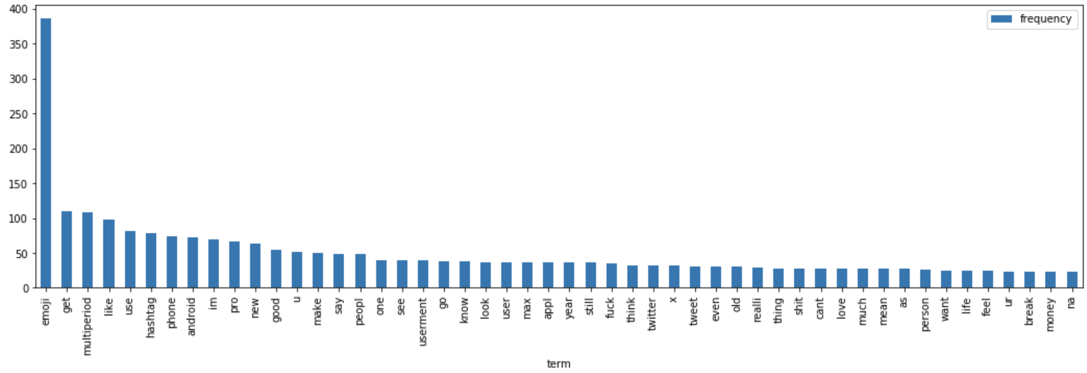


Figure 3.7: Top 50 most occurring words in Affective class

Unlike in the distribution graph for Affective class, the graph in Figure 3.8 for the Behavioural class shows more verbs such as “got”, “need”, “want” and “buy”. This observation is expected for the Behavioural class as this attitude type consists of a person's tendencies to behave in a particular way toward an object or his/her intentions with regard to it.

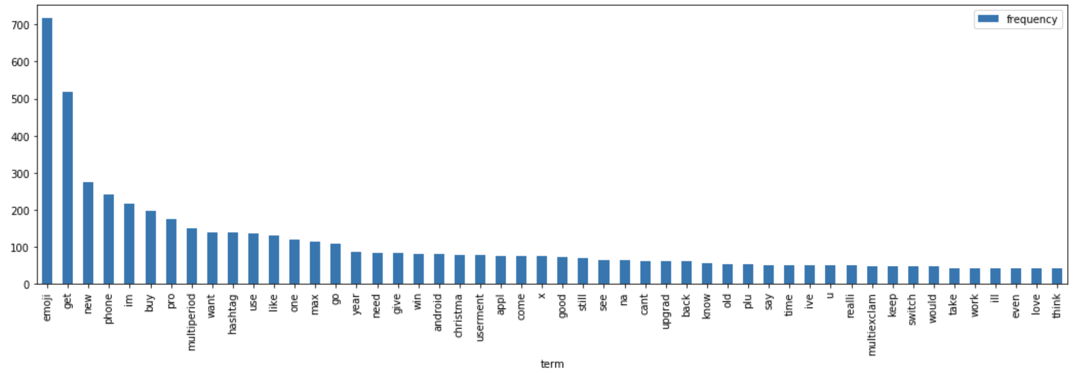


Figure 3.8: Top 50 most occurring words in Behavioural class

As discussed in Section 1.2.1, Cognitive attitudes are based on knowledge or beliefs. Hence it is mostly targeted towards certain attributes or specifications of the attitude object. This characteristic is well depicted in the graph below (Figure 3.9). This graph contains terms associated with the iPhone's specifications such as "camera", "image", "screen" and "battery". These are types of terms that one will use to express the learned attitudes towards our attitude object.

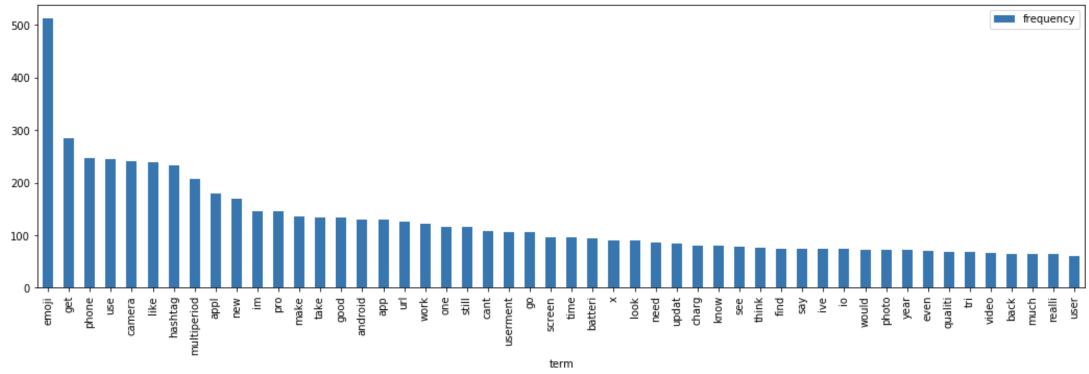


Figure 3.9: Top 50 most occurring words in Cognitive class

3.4 Implementation of Preprocessing

Preprocessing in social media sentiment analysis plays a crucial role. Various preprocessing techniques used in previous literature and their impact on the results were compared and contrasted in the Literature Review section. For the current experiment, several preprocessing techniques have been selected by considering their behaviour in the previous research. Figure 3.10 depicts the list of preprocessing techniques in their order of execution. The objective of trying out many techniques is to separately evaluate the impact of these techniques for the classification model for this given problem.

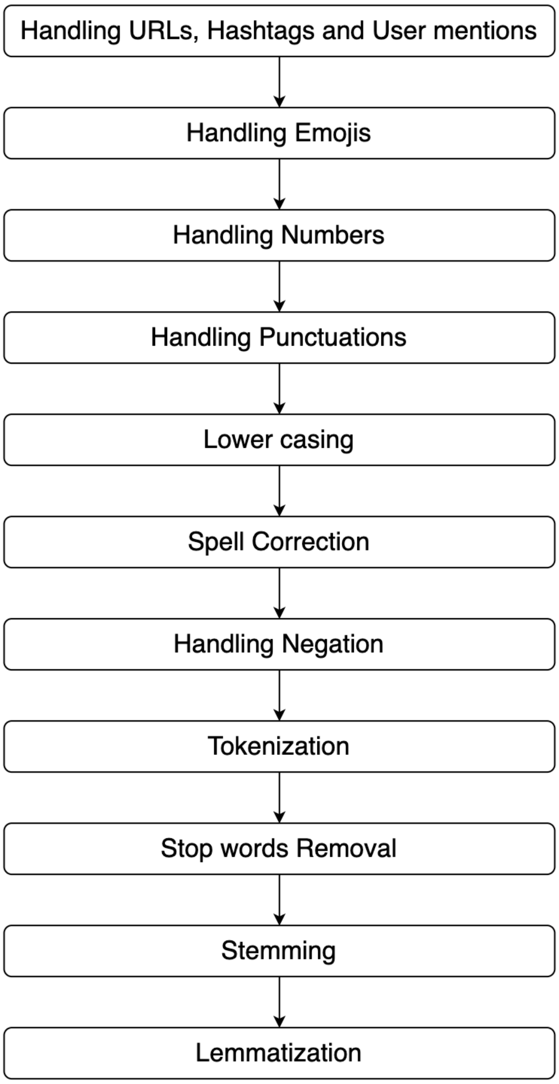


Figure 3.10: Process flow- Preprocessing

Tweets often contain various attributes such as “Retweet”, “Twitter for iPhone”, URLs and user mentions (@someusername). These attributes are a part of the text and might give information on the expressed sentiment. Therefore, such attributes cannot be completely disregarded. Some previous studies have used tags to indicate these attributes instead of removing them [12], [15]. Same technique was employed in this experiment as shown in Table 3.2.

Table 3.2: Example- Handling Twitter tags

Original Text	**Processed Text**
“Sure wish I had that iPhone 11 Max 👀 https://twitter.com/breyochamays/status /1202002608254332929 ...”	*“Sure wish I had that iPhone 11 Max 👀 _<url> ...”*
“@Apple how about a free iPhone 11”	*<user_mention> how about a free iPhone 11*
“They call it evolution of cameras but i say its nuthin but cell division. #iPhone pic.twitter.com/Wv95px9UvA”	*“They call it evolution of cameras but i say its nuthin but cell division. <hashtag><image>”*

Simple regular expressions were used to filter out these attributes as they have recurring patterns. The most common Twitter attributes identified from the dataset are, Images, URLs and user mentions and they have been handled using relevant tags as depicted in the above table. The data set doesn’t contain any Retweets hence that term wasn’t seen in the dataset. It was also observed that attributes such as “Twitter for iPhone” or “Twitter for Android” have not been extracted by the tool. However, there were instances where users have explicitly included those attributes in their text to express sentiments, such occurrences were not replaced or removed.

Emojis are pictographs of faces, objects and symbols (“😄”) whereas emoticons are punctuation marks arranged in a certain way to display an emotion (“:D”). Nevertheless, both of these characters are strong indicators of emotions and are widely used in informal social media context. With the use of smartphones, emoticons have been replaced with emojis. Therefore, in this study we've only focused on handling emojis which was found to be more commonly present in the dataset than emoticons.

There are several ways to handle emojis as described in the literature. The simplest technique is to keep the emojis as it is in the text. The emojis can also be converted to text using the Python Emoji[6] module. This translates an emoji into its text equivalent (e.g.: "😄" will be translated to "grinning_face_with_smiling_eyes")

But this does not help to generalize the text. Another method is to add a tag such as <emoji> in place of the emoji. This generalizes the emojis and causes less sparsity as there are numerous emoji characters. A more advanced technique is to categorize emojis into several classes [10]. This method generalizes the emojis while preserving its sentiment.

E.g.:

Happy-> 😃😇😊🤩😍

Sad-> 😞😢😭☹️

Angry-> 😡😠🤬😫

Neutral-> 😐😶👁🏁

As a suitable dictionary of emojis couldn't be found, in this study, the <emoji> was used in place of the emoji. Unicode characters range of the emojis were used to detect the emojis in this implementation. Following Table 3.3 shows the processed text using this method.

Table 3.3: Example- Handling Emojis

Original Text	**Processed Text**
"got the new iphone 🤪"	*"got the new iphone <emoji>"*
"I miss my iPhone 😩😩😩"	*"I miss my iPhone <emoji><emoji><emoji>"*

[6] https://pypi.org/project/emoji/

Lowercasing the text is a common preprocessing technique used in most of the text classification problems [10]. But in some previous studies it was argued that texts in uppercase express strong sentiments. Therefore, it was suggested to add some indication of all capitalized text i.e.:<all_caps> before the word when handling the casing [12]. But in this data set such texts were not frequent therefore a significant impact was not expected from the casing of the word. Hence a simple lowercasing technique was used by calling the Python's inbuilt "string.lower" method. Table 3.4 below shows the processed text using this method.

Table 3.4: Example- Lowercasing

Original Text	**Processed Text**
"This is why I need an iPhone ASAP"	*"this is why i need an iphone asap"*
"I JUST FOUND OUT MY GOOGLE PIXEL HAS A BUILT IN WIRELESS CHARGER ITS OVER FOR U IPHONE"	*"i just found out my google pixel has a built in wireless charger its over for u iphone"*
"Ago @tim_cook you are KILLING ME broo!! this is why I have not and will not update past this Iphone 6s smh..."	*"ago @tim_cook you are killing me broo!! this is why i have not and will not update past this iphone 6s smh..."*

In social media context, various short words are used and therefore a basic spell corrector might not be able to catch them. Therefore, for this technique, 4 spell corrector algorithms were tested, they are namely, Pyspellchecker[7], Symspellpy[8] and Ekphrasis[9]. Each of these libraries use different techniques to implement spell correction. Pyspellchecker based on Peter Norvig explanation[10] on writing a spell corrector.

[7] https://github.com/barrust/pyspellchecker
[8] https://github.com/mammothb/symspellpy
[9] https://github.com/cbaziotis/ekphrasis
[10] http://norvig.com/spell-correct.html

Norvig' method uses a distance calculation called Levenshtein Distance to find permutations within an edit distance of 2 from the original term. Words appearing more frequently in this permutation list are considered more likely to be the correct word for the input word. Symspellpy uses the The Symmetric Delete spelling correction algorithm. This is said to be 6 times faster than a standard spell corrector. Even though this uses the same distance function, this algorithm reduces the complexity of edit candidate generation and dictionary lookup. The reason for considering Ekphrasis is that it is optimized for text from social networks, such as Twitter or Facebook. This was developed as part of the text processing pipeline for a study submitted for SemEval-2017 Task 4 (English), Sentiment Analysis in Twitter [66]. In order to select one of these algorithms for the implementation, a sample text was used to test their performance and results are shown in Table 3.5.

Table 3.5: Evaluation of Different Spell Correctors

Original list of strings	*"lol", "lmao" "hellooo", "dissappoint", "iphone", "yu"*
Pyspellchecker	*"lol", "mao", "hello", "disappoint", "phone", "yu"*
Symspellpy	*"low", "lao", "hello", "disappoint", "iphone", "you"*
Ekphrasis	*"lol", "lmao", "hello", "disappoint", "iphone", "yu"*

It was observed that Symspellpy is not ideal for social media context. Even though Ekphrasis is the most ideal for this context, it couldn't be optimized for the large dataset and therefore faced issues during the runtime. Considering these limitations, Pyspellchecker was selected for this study.

Individual numbers don't usually carry any sentiment [14]. Therefore, most of the studies have removed numbers from the text. But in this dataset, numerical values were often used to describe prices and technical specifications of the product. Hence, instead of removing numbers, in this study they have been replaced with tags such as <dollar_amount>, <percentage> and <number> as shown in the Table 3.6 below [13]. This will retain some level of information while generalizing the text. Regular expressions were used to identify these different types of numbers. This method is capable of identifying comma-separated as well as period-separated numbers.

Table 3.6: Example- Handling Numbers

Original Text	Processed Text
“Back in the day by this time your iPhone would’ve shut off by now due to cold weather with a 50% battery”	*“Back in the day by this time your iPhone would’ve shut off by now due to cold weather with a <percentage> battery”*
“About to switch to iPhone again. Using Visible, a Verizon MNVO , $40 all in for unlimited everything.”	*“About to switch to iPhone again. Using Visible, a Verizon MNVO , <dollar_amount> all in for unlimited everything.”*
“Got the new IPhone 11 Pro Max 🤩”	*“Got the new IPhone <number> Pro Max 🤩”*

In this research, it was observed that repeated punctuation marks often express strong emotions (e.g.: “???” and “!!!”). Therefore, in order to preserve such informative features, non repetitive punctuation marks were eliminated and repetitive punctuation marks were replaced with tags <multi_question> or <multi_exclamation> using regular expressions as shown in Table 3.7. This method was adapted from Effrosynidis et al. [12].

Table 3.7: Example- Handling Punctuations

Original Text	Processed Text
“This new battery on the new iPhone is bomb!!!”	*“This new battery on the new iPhone is bomb <multi_exclamation>*
“Four cameras? Really??? I just bought this iPhone Pro Max 11 I’m not buying another phone till 2021 😂there might be a 15c out by then”	*“Four cameras Really <multi_question> I just bought this iPhone Pro Max 11 I’m not buying another phone till 2021 😂there might be a 15c out by then”*

As discussed in the Literature Review section, words such as “won’t”, “can’t” and “don’t” negate their following word. They all try to express a similar context therefore instead of various negations, they were all substituted with <not> [10]. For this technique, Python’s “replace” method was used. A list of identified negation words in English language was defined and if an input word exists in that list, it will be replaced with <not> as shown in the Table 3.8.

Table 3.8: Example- Handling Negation

Original Text	**Processed Text**
“Guess they won’t be getting the new IPhone.”	*“Guess they <not> be getting the new IPhone.”*
“I refuse to go past the iPhone 6s. I don’t want a dongle”	*“I refuse to go past the iPhone 6s. I <not> want a dongle”*

Tokenization is breaking down a text into tokens/ words. These tokenized words are then used to remove stop words. Stop words are commonly used words which often don’t have sentiment values in the context. Therefore, it is argued it is safe to eliminate such words so that the model can focus on more important words [18].

Finally stemming and lemmatization was conducted on the data set to bring the word tokens into their base form. For stemming PorterStemmer was used and WordNetLemmatizer was used for lemmatization.

The order in which they will be executed was carefully organized considering the inter-dependencies. For example, if the punctuation removal was done before handling URLs, URLs will no longer be identifiable using regular expressions. An in depth evaluation of the impact of each of these techniques on the model accuracy with regard to classification algorithms will be discussed in the Chapter 4.

3.5 Implementation of Feature Extraction

The preprocessed tweets need to be transformed into feature vectors that are accepted by machine learning algorithms. There are various techniques to extract features from textual data and a few of them were analysed and discussed in the literature review. Out of those techniques, following feature extraction techniques were experimented in this research,

- N-gram (N=1 and N=2)
- POS tags
- TF-IDF
- Word Embedding (Word2Vec and FastText)

Bag of words (BOW) also known as N-gram (N=1) is one of the most primitive feature extraction methods that is easy to implement. Therefore, it was the first feature extraction technique that was carried out. The concept of BOW is to denote the frequency of term presence after each term in the dataset is converted into a feature. Accordingly, using the Scikit-Learn module "CountVectorizer", around 5000 terms were identified as features. This technique does not take position or the grammatical structure into account. Bigram or N-gram (N=2) is an extension of BOW that takes the position into account by taking each word pair as a single feature. The same "CountVectorizer" module was used to extract bigrams with the additional parameter ngram_range=(2, 2). Using this technique around 33,000 features were picked out.

In POS instead of merely considering the term it also includes the POS tag of the term as a feature. By applying this technique for the current dataset, a feature set of around 8,000 features were compiled.

As the above techniques only consider the frequency, TF-IDF was selected as another feature extraction method for this study. "TfidfVectorizer" module in the Scikit Learn's feature_extraction package was used to implement this. The same number of features as the BOW was extracted but the difference here is the weighting scheme.

Word2Vec is one of the recent developments in the NLP world that has caught a lot of attention over the years due to its proven success as a feature extraction technique in many areas. There are 2 methods that the Word2Vec technique uses to construct embeddings, namely, Skip Gram and Common Bag Of Words (CBOW). The CBOW method takes the context of each word as the input and then attempts to predict the word corresponding to its context. On the other hand, Skip-gram tries to predict the immediate neighbours of a given word. It looks one word ahead and one behind, or two ahead and two behind or any count that is defined. Skip-gram and CBOW results were compared and decided to use Skip-gram for this research as it yielded better results than CBOW for the current dataset. For this experiment the Gensim[11] implementation of Word2Vec was employed in constructing word embeddings. In the implementation, a tweet is represented by the average of the word embedding vectors of the words in the tweet. Feature size selected for this method is 200.

In order to generate a more robust Word2Vec model a large corpus is needed. Even though this dataset doesn't have a big enough corpus, luckily, a generated Word2Vec model can be reused across different domains. Hence there are numerous pre-trained Word2Vec models that can be easily adopted to this solution. Glove Twitter pre-trained embeddings [67] and Word2Vec Google News[12] pre-trained embeddings were used to build models for this experiment.

FastText[13] developed by Facebook is the latest breakthrough in the area of word embedding. It is a library for more efficient learning of word representations and sentence classification. This overcomes the Word2Vec model's inability to deal with out of corpus words by treating each word as a composition of N-grams. In order to implement this in Python, Gensim library was used.

In addition to these textual features, some metadata such as length of the text, number of verbs and timestamp were also tried out as features. But these features did not show a significant impact on the results in this experiment, therefore was not included in the final solution.

[11] https://pypi.org/project/gensim/

[12] https://code.google.com/archive/p/word2vec/

[13] https://github.com/facebookresearch/fastText

3.6 Implementation of Feature Selection

Features extracted using the above mentioned techniques except word embedding have very high dimensions. Therefore, a feature selection method was employed to filter out only the most influential features. For this, the Chi-Squared score was calculated for each feature and the ones with the best score was selected. Scikit-Learn's Chi-Squared implementation in its "feature_selection" module was used to implement this.

3.7 Implementation of Classification

Once after the dataset has been processed in such a way that the classification algorithms accept as input, the next step is to design and develop the classifiers. For this study, only the supervised learning approach was selected considering its performance in the previous studies. But when selecting the classification algorithms, it was decided to select a variety of algorithms of different categories of supervised learning approach. Listed below in Table 3.9 is the selected algorithms and the categories they belong to. As most of these classifiers and their underlying concepts have already been discussed in Chapter 2, in this section only the justification for selecting each of these algorithms for this study will be discussed. Comparison of the results of each of these classifiers will be discussed in detail in the Chapter 4.

Table 3.9: Selected Supervised Learning Classifiers

Category	Classification Algorithm
Probabilistic Classifiers	Multinomial Naive Bayes
Linear Classifiers	Logistic Regression
	SVM
Decision Tree Based Classifiers	Random Forest (also an ensemble classifier)
Ensemble Classifiers	Stacking
	Random Subspace
Neural Networks	Simple Neural Network
	Convolutional Neural Network
	Recurrent Neural Network

3.7.1 Multinomial Naive Bayes (MNB)

Naive Bayes is a primitive algorithm but is commonly applied for solving a variety of classification problems considering its simplicity and interpretability. Multinomial Naive Bayes is an extension of Naive Bayes that is used to solve multiclass classification problems. There are a few properties of MNB that seemed to be advantageous for this experiment.

- Fast and highly scalable.
- A simple and interpretable algorithm to explore the problem at hand which makes it a good starting point algorithm.
- Makes probabilistic predictions using the basic concept of Naive Bayes that many are familiar with.
- It is most suitable for classification with discrete features (e.g., word counts for text classification).
- Works well with both integers (BOW) as well as fractional counts (TF-IDF).

Scikit-Learn's "MultinomialNB" class under the "naive_bayes" module was used to implement this classifier.

3.7.2 Logistic Regression

Regression algorithms are used to predict continuous variables. But Logistic Regression is a special type of regression that is specifically designed to predict discrete output variables. This is also a simple algorithm but it performs well in comparison to other complex algorithms. Its below listed properties are what makes it a good fit for the problem at hand.

- Provides great interpretability of the model.
- It is easily customizable and optimizable using its numerous parameters.
- It trains very efficiently therefore does not require too many computational resources.
- It is often considered as a good baseline model that can be used to compare and measure the performance of more complex models.

- It can handle both dense and sparse input and the variables don’t need to be normally distributed.

Scikit-Learn’s “LogisticRegression” class under the “linear_model” module was used to implement this classifier.

3.7.3 Support Vector Machines (SVM)

SVM has earned its popularity in the area of NLP as it works well with both structured and semi-structured data such as text and images. Considering its promising performance in the previous literature, SVM classifier was experimented in this research and it confirmed previous findings. SVM has worked well in many state-of-the-art text classification tasks due to its unique properties.

- It has the ability to handle high dimensional spaces effectively.
- It is easily customizable and optimizable using its numerous parameters.
- SVM models have generalization in practice. Therefore, the risk of overfitting is low in SVM.
- It uses the kernel trick, so a sound knowledge about the problem can be acquired by engineering the kernel.

Scikit-Learn’s “SVC” class under the “svm” module was used to implement this classifier.

3.7.4 Random Forest

Even though Random Forest is another ensemble learning method, this was selected as it is from the class of decision trees. This is a popular algorithm which is recognized for its high interpretability. But this algorithm is not as popular in NLP literature as SVM or Logistic Regression. Nevertheless, this was selected for this study to experiment how a decision tree based classifier would perform when applied for this classification problem. Following are some of the properties of Random Forest that were deemed to be advantageous for this problem.

- Decision trees in RF do not require normalization or scaling of data.
- Generate understandable rules, hence the high interpretability.
- Linearity between independent and dependent variables is not a constraint for the DF model predictions.
- Getting the majority vote of several decision trees as this is an ensemble method

Scikit-Learn's "RandomForestClass" class under the "ensemble" module was used to implement this classifier.

3.7.5 Ensemble Classifiers

Even though ensemble classifiers are not very popular in the paradigm of text classification, as discussed in Chapter 2, some studies that have incorporated ensemble classifiers have shown promising results. Ensemble classifiers bring together the best properties of all the base learners. Additionally,

- Reduces overfitting. This is important for this experiment because it does not have a very large dataset.
- It is easily customizable and optimizable using its numerous parameters that belong to each base learner. This creates a large space for parameters that can be experimented with.

Out of the several types of ensemble models discussed in the Section 2.4.3, Stacking and Random Subspace methods were selected for this study. Stacking was selected based on its performance in the previous studies. Random subspace was selected considering the large featureset.

Scikit-Learn provides an implementation of stacking in its "StackingClassifier" class under the "ensemble" module. Random Subspace can be implemented using the same module by importing the "BaggingClassifier" class and setting its parameter "bootstrap_features" to true to test the models on different feature sets. Best performing models in this experiment were selected as the base models for the ensemble classifiers.

3.7.6 Simple Neural Network

Before moving to any deep learning models, a simple Neural Network (NN) was tested as the baseline model to compare the Deep Neural Networks with. This model consists of an input layer, 2 hidden dense layers and an output layer. The activation function used in all 3 layers is “softmax”. The optimizer selected here is “RMSprop”. Even though the “Adam” optimizer is known to perform well in most problems, for this particular experiment it was observed that the RMSprop optimizer performed better than the Adam optimizer. Keras library with Tensorflow backend was used to implement this simple NN. Even though NN often needs a substantial amount of data to get good accuracy and often takes more training time that other algorithms, the advantages of using NN for sentiment analysis is as follow,

- They are easily scalable to fit most problems as they have many parameters and architectures that can be played around with.
- NN models are useful when there is less domain knowledge and less knowledge on features to be used.
- NN has proven to be more effective on problems that are not linearly separable than other classification algorithms.

3.7.7 Convolutional Neural Network (CNN)

Deep Learning is one of the recent breakthroughs in the area of machine learning. CNN is a feed forward Deep neural network which is often used in image classification. But researchers such as Santos and Gatti [40] tried out CNN in their sentiment analysis study of short texts and yielded good results. In addition to the properties of NN, CNN has some additional advantages such as,

- Even though they are computationally expensive, they perform better than NN in most cases.
- CNNs makes use of spatial locality. This is done by enforcing a local connectivity pattern between neurons of adjacent layers. This makes the network to first come up with representations of smaller parts of the input. Then use that to assemble larger parts.

The Keras library with Tensorflow backend was used to implement this model. For this experiment a special feature set was compiled using the Keras's Tockeniser method. This tokenizes the text into a vector representation. CNN accepts the data in a form of matrix with a fixed size. So, the tokenized text needs to be brought to the same length. When deciding the size, either the max or average length of a tweet can be selected. After experimenting with various values, 60 was the size selected for the matrix, which is closer to the maximum length of tweets. So, in order to bring all the tweets to length of 60, the shorter tweets were padded with zeros and longer ones were pruned using Keras's "pad_sequence" method.

Shown below in Figure 3.11 is the architecture of the CNN model used for this experiment. Adding more layers such as Dropout layers and Dense layers were tried out. But they did not show any increase in the accuracy. The same loss function and the optimizer as the NN model was used. But different activation functions were used in the layers. Relu was used in the Convolutional layer, Softmax in the hidden layer and Sigmoid in the output layer.

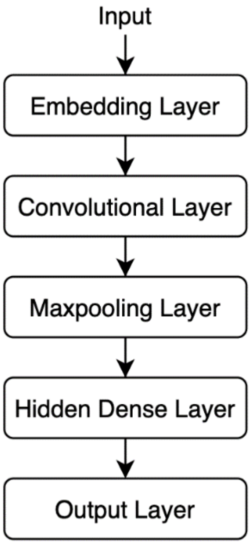


Figure 3.11: CNN architecture

3.7.8 Recurrent Neural Network (RNN)

RNN is another Deep Learning model where connections between nodes form a directed graph. RNN-LSTM (Long-short Term Memory) is a variation of RNN. The concept of LSTM is that it allows to preserve the error that can be backpropagated through time and layers in RNN. This is done using a gated cell. In this gated cell, learned information can be stored, written, or read like data in a computer's memory. The cell is responsible for making decisions such as what to store, and when to allow reads, writes and erasures, using gates that open and close. LSTMs are good at handling sequential data. As opposed to CNN which makes use of spatial locality, RNN-LSTM has a temporal dimension which takes time and sequence into account. This simply means that this allows considering input with previous input. These properties make it ideal for sequential data, hence the remarkable performance in NLP.

The same word embedding technique used in the CNN implementation was used here. Keras with Tensorflow backend was used to build the RNN-LSTM model. The below depicted architecture in Figure 3.12 was implemented.

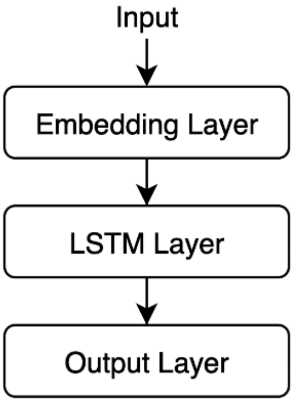


Figure 3.12: RNN-LSTM architecture

This is not a very sophisticated model architecture. The LSTM layer is the significant layer here. There are two methods to add a Dropout regularizer to avoid overfitting in this model. Firstly, adding a Dropout layer was tried out but it decreased the accuracy. Instead, the dropout option was added in the LSTM layer which was deemed more suitable for RNN-LSTM models. Same loss and optimizer types were used and the Softmax activation function was used.

3.8 Handling Imbalanced Class Problem

As depicted in the Figure 3.3 in the EDA (Section 3.3), the class distribution is not equal. Even though there is no significant difference between Behavioural and Cognitive classes. The Affective class is almost half the size of Cognitive class. Hence Cognitive and Affective classes can be introduced as the majority classes and Affective can be introduced as the minority class.

Differences in distribution of classes largely affect the accuracy of a model. During the initial phases of implementation and evaluation, it was observed that the Affective class had the lowest F1 score, thus affecting the overall F1 score even if the other two classes have good F1 scores. So, in order to improve the model accuracy, it was important to tackle this class imbalance problem.

3.8.1 Re-sampling

Re-sampling is one of the most common ways of handling the class imbalance. The dataset can be either undersampled or oversampled. Initially the undersampling technique was carried out. Both majority classes were undersampled to the size of the Affective class. This method randomly removes samples from the majority classes. Two over sampling techniques were tried out, namely, Random Over Sampling (ROS) and Synthetic Minority Oversampling (SMOTE).

Imbalanced-Learn[14] library was used to resample. Its “RandomUnderSampler” under the “under_sampling” module was used to implement the undersampling and, “RandomOverSampler” and “SMOTE” under the “over_sampling” was used to implement oversampling.

[14] https://pypi.org/project/imbalanced-learn/

3.8.2 Class Weights

In this method, instead of resampling, the model is made aware of the class imbalance by incorporating weights into the cost function. A higher weight is given for the minority class and lower weights for the majority classes. The weights are usually determined based on the fractions obtained from the class distribution. These are called balance weights. But weights can also be determined using an optimization method such as Grid search. For this experiment the balance weights were used. In most models including Scikit-learn and Keras, weights can be assigned using the “weights” parameter when the model is being built. Here, the user can provide a list of weights or use the “balance” keyword to let the model calculate the balanced weights.

3.8.3 Hyper Parameter Optimization

Models have numerous parameters. The effect of these parameters depends on the feature set class distribution and the sample set. Selecting the optimum parameters for a model is crucial for its performance as they are the determinants of the model accuracy. Usually these are manually selected and gradually adjusted based on the model accuracy. But there are existing techniques to select the optimum parameters. Random search, Grid Search and Bayesian optimization are a few such algorithms. In order to further improve the accuracy after trying out class weights, the hyper parameter optimization was used. For this Hyperopt[15] library was used. The search space for the parameters can be provided by the user. Hyperopt runs several trials with different parameters from this space until the highest accuracy is reached. How the parameters are selected for the trials are decided based on the selected algorithm. Hyperopt encompasses 3 algorithms that we can select from, namely, Random Search, Tree of Parzen Estimators (TPE), Adaptive TPE. The default algorithm TPE of Hyperopt was used for this experiment. The experiment showed a significant improvement of the accuracy proving the importance of parameter tuning.

[15] https://github.com/hyperopt/hyperopt

3.9 Evaluation Metrics

Several evaluation metrics that were used in previous studies to evaluate the classification models were discussed in Chapter 2. It was observed that F1 score is the most common and accepted metric to calculate the model performance across all classes. Hence for this research F1 score was used as the main metric of evaluation. Along with the F1 score, Precision and Recall of some models were also compared.

For this study, the K-Fold cross validation method was used instead of the Holdout method. K-Fold cross validation is a more reliable method of evaluating specially if the data at hand is limited. As K=5 was selected for this experiment, the dataset is split into 5 subsamples and is made sure that each fold is used as a testing set at some point. Scikit-Learn provides an implementation of this method in their "model_selection" module. During each fold, this calculates the F1 score, precision and recall for each class and then they are averaged for each fold. There are several averaging methods namely, macro, micro and weighted. As we are dealing with imbalanced classes, the weighted average was selected as the most suitable method. It includes the distribution of classes in its calculation. For the evaluation of models, the average values returned from the K-Fold evaluation was averaged to calculate the overall scores for that model.

However, during the implementation stage, it was observed that Deep Learning models take a long training period. Therefore, the holdout method was used for evaluating Deep Learning models.

4. SYSTEM EVALUATION

In the above chapter, implementation of various techniques of data extraction, preprocessing, feature extraction, feature selection and classification algorithms were discussed. The performance of the solution depends on each of these individual aspects. In this chapter, the evaluation of these techniques and their contribution to the solution will be discussed.

4.1 Inter-Rater Agreement

After the data annotation has been done, Cohen's Kappa measure was used to measure the inter-rater agreement. Equation 8 gives the formula to calculate the Cohen's Kappa score. Inter-rater agreement was calculated for 2 annotators based on a sample dataset of size 1,000. The sample size of 1,000 out of the complete dataset of size 10,000 was selected with a confidence interval of 95% and leaving 2.94% as margin of error.

Both the annotators possess domain knowledge in the area of psychology. Annotator 1 is currently reading for her PhD. Annotator 2 has completed her Certificate in Business and Organizational Psychology.

The Kappa score was calculated using the "cohen_kappa_score" module available in the Python Scikit-learn library under the "metrics" package. Following Table 4.1 presents the breakdown of the calculation.

Table 4.1: Inter Rater Agreement

		Annotator 2				
		Affective	Behavioural	Cognitive	Neutral	Advertisement
Annotator 1	Affective	45	**18**	**26**	**19**	0
	Behavioural	3	184	12	6	2
	Cognitive	1	3	215	5	0
	Neutral	1	4	7	309	0
	Advertisement	0	2	2	6	130

$$kappa(\kappa) = \frac{P_o - P_e}{1 - P_e}$$

Equation 4.1: Cohen's Kappa Score

= (0.883-0.23699)/(1-0.23699)

= 0.85

According to the interpretation of Cohen's Kappa coefficient, 0.85 is a very good strength of agreement.

In the above Table 4.1, the cells highlighted in the diagonal line are the counts agreed by both the annotators. The highest agreement is for the Neutral class whereas the lowest agreement is for the Affective class. The Affective class is a rather heterogeneous class. Its heterogeneous nature also caused low accuracy during the model evaluation which will be further discussed in the next chapter. Nevertheless, following are a few conflicting scenarios observed during the annotation.

1. Tweets may contain more than one attitude. In most cases it is Affective and one other attitude type. In such cases, the annotators have given either one of the attitudes which they believe to be prominent.

   E.g.: "*Battery life on the iPhone 11 Max Pro is amazing. I'm liking this phone more everyday.*".
   This tweet contains an experienced Cognitive attitude towards the Battery life of the iPhone. Also the user expresses a growing Affective attitude towards the product. The Annotator 1 has labeled this as Affective whereas the Annotator 2 has labeled this as Cognitive.

2. Unlike other classes, the tweets of the class Affective, the attitude is expressed implicitly with regard to social status (as given in the example below), political/ social standpoints, memories or experiences.

E.g.: “*IPHONE 6 Now*😂😂😂

Which type of phone are you using ??

Me: you know Drogba, that famous footballer... he is a millionaire

My Guy: Yes... I know Didier Drogba

Me: we have the same phone 📱 😂😂💦💦”

These implicitly expressed attitudes can easily be neglected as non-attitude tweets. Hence the Annotator 1 has labeled this as Affective whereas the Annotator 2 has labeled this as Neutral.

3. Some attitudes are not directly aimed at our attitude object, iPhone.

E.g.: “*totally agree. the top comments under this NYT article reflect the same thing. turn off the tablet, xbox, tv, iphone.. get a book to read, start from the parents. yet, politicians and school officials conveniently blame this on "social injustice" and punish those who work hard*”

Even though an Affective attitude is expressed here, it is not directly on the iPhone. Hence one annotator has recognized this as Affective whereas the other annotator has labeled it as Neutral.

4.2 Baseline Experiment

An implementation that can be used as a baseline model for this research problem couldn’t be found. Therefore, a Random classifier was built to measure the baseline performance, that is the success rate expected to achieve by simply guessing. The Stratified method which generates predictions by respecting the training set’s class distribution was used to implement this dummy classifier. This Random classifier will be used to compare the performance of Machine Learning Classifiers.

The Effects of techniques such as preprocessing, feature selection and resampling cannot be tested against a random model, as obviously the random guessing will not be impacted by these techniques. Therefore, in order to make the effect of various techniques comparable 2 baseline models were implemented. SVM and Logistic regression algorithms were selected as the baseline models considering the fact that in most studies these two models have been able to yield the best results.

In this experiment the BOW which is the most primitive feature extraction method has been selected and no text preprocessing has been done. Rest of the evaluations in the following sections will be compared against these models while keeping their parameters constant across the various techniques that will be evaluated. Results of the baseline models are presented in Table 4.2.

Table 4.2: Results of Baseline Experiment

Classifier	**Feature Set**	**Precision %**	**Recall %**	**F1 Score %**
Random Classifier	BOW	36.03	35.90	35.90
Logistic Regression	BOW	70.57	70.44	70.44
SVM	BOW	69.70	69.33	69.40

Hyper-parameter used in baseline Logistic Regression Model: C (Strength)= 0.59, Solver= ‘liblinear’, Maximum Iterations= 500, Tolerance= 4.7e-05

Hyper-parameter values were used in baseline SVM Model: C (Strength)= 0.16, Gamma= 5.9, Kernel= “linear”

4.3 Effect of Preprocessing Techniques

In the Chapter 3, implementation of 11 preprocessing techniques were discussed. This section will discuss how each of those techniques contributed to the performance of the model. Initially the effect of individual techniques was experimented and then how the results vary when different types of techniques are combined were evaluated.

The following results in Table 4.3 shows how each of the techniques impacted the results of the baseline model individually. The effect was calculated by subtracting the F1 Scores of baseline models from the F1 scores of models after applying the technique.

Table 4.3: Effect of Preprocessing Techniques

	F1 Score %		**Effect on the Baseline Models**	
Preprocessing Technique	**Logistic Regression**	**SVM**	**Logistic Regression**	**SVM**
Handling Twitter tags	70.06	69.32	-0.38	-0.08
Handling Emojis	70.66	70.01	0.21	0.60
Handling Numbers	70.42	70.11	-0.02	0.71
Handling Punctuations	70.53	69.75	0.09	0.35
Lowercasing	70.32	69.95	-0.12	0.55
Spell correction	70.17	68.72	-0.28	-0.68
Handling Negation	70.22	69.37	-0.22	-0.03
Removing Stop Words	68.11	66.51	-2.33	-2.89
Stemming	70.98	70.32	0.54	0.92
Lemmatization	71.21	70.01	0.77	0.60

Even though it was considered that attributes such as @user, #hashtags and URLs add noise to the data, the above Table 4.3 shows that handling those features had a negative impact on the performance. Initially the tags were replaced as discussed in Chapter 3. Since the impact was negative, then the tags were removed. Even though removing showed better accuracy than replacing, it still had a negative impact on the baseline models.

Emojis are often expected to be strong indicators of emotions in the social media context. Hence instead of removing them, <Emoji> tag was used. Applying this technique shows a slight increment of performance in comparison to the baseline models.

Initially as discussed in the Chapter 3, all the numbers except Dollar amounts and percentage values were removed. But this experiment had a negative effect on the baseline results. So, it was then experimented by removing all numbers including the Dollar amounts and percentage values. As shown in Table 4.3, it had a positive effect on the baseline results of the SVM model. But it had a minor decrement in the result of the Logistic Regression model. Hence it can be observed that even though Dollar values and percentages were expected to be useful features in this context, they do not have any special value and are non-sentiment terms like the rest of the numbers.

Replacing repeated punctuation marks with special tags and removing the rest of all punctuation marks shows an improvement in the results. However, using only this technique without handling the URLs is not suitable as it will create non existing patterns when the punctuations are removed from the URLs.

Lowercasing has shown a positive effect on SVM whereas a negative effect on Logistic Regression model. This difference could be due to the fact that unlike in a formal context in this social media context linguistic rules such as upper casing the first letter of Proper nouns or first word in a sentence is not often given attention. However, SVM seems to have largely benefitted by this technique by identifying the generalization of text that this technique has caused.

It was observed that spell correction had a negative impact on the accuracy of both baseline models. This could be due to the fact that the spell checker used here is not designed for social media context. As observed in the Chapter 3, key terms such as "iphone" were converted to "phone" but not "iPhone". The term "iphone" becomes an insignificant term in this study because it is the search term that is expected to be presented in all tweets. However, when this is translated to "phone", it brings a new value because as observed in the Section 3.3, its frequency of occurrence varies from class to class.

Handling negation seems to have a negative impact on the results. This could be due to the fact that certain auxiliary verbs have value in this context with regards to identifying the variance among classes.

By far, removing stop words shows the highest negative impact. The NLTK Corpus of Stop words contains an almost exhaustive list of stop words in English language. Out of those words, some words may carry an importance in different contexts. Hence the negative impact on results. A context specific stop words list might have given different results. Since such a list couldn't be found, a custom list was curated with few of the common words from the most occurring list obtained from the Section 3.3. Even though the magnitude was reduced by 1%, the impact was still negative.

As seen in most of the previous studies, stemming and lemmatization had the highest positive impact on the accuracy of the model. Intuitively, bringing the words to their base forms generalizes the terms, thereby adding a significance to those words.

All the techniques that had a positive impact on the results were selected and these all were combined together to experiment the overall impact. The results presented in Table 4.4 shows that with the right preprocessing techniques in place, they can contribute positively to the performance of the model in this study.

Table 4.4: Results of Preprocessing Step Vs Baseline Model

	Classifier	**Precision %**	**Recall %**	**F1 Score %**
Baseline Models with unprocessed text as input	*Logistic Regression*	*70.57*	*70.44*	*70.44*
	SVM	*69.70*	*69.33*	*69.40*
Preprocessed text as input	Logistic Regression	71.48	71.41	71.42
	SVM	70.95	70.80	70.83

4.4 Effect of Resampling

The effect of resampling on the baseline models is presented in Table 4.5. This experiment was conducted to improve the accuracy by handling the class imbalance problem. But when compared with the baseline results, it can be observed that all three methods have failed to do so. Undersampling has the highest negative impact on SVM which can be expected because this method causes information loss. However, even though oversampling was expected to have increased performance, it has given the highest negative impact on Logistic Regression. These results in Table 4.5 show that resampling technique gives poor results for this study. The effect was calculated by subtracting the F1 Scores of baseline models from the F1 scores of models after applying resampling.

Table 4.5: Effect of resampling techniques

	F1 Score %		**Effect on the Baseline Model**	
Re-sampling Technique	**Logistic Regression**	**SVM**	**Logistic Regression**	**SVM**
Undersampling	70.02	66.07	-0.42	-3.33
SMOTE	68.97	67.79	-1.48	-1.61
ROS	69.57	69.07	-0.87	-0.33

4.5 Effect of Feature Selection

This experiment was conducted on the baseline models to evaluate the effect of feature selection using the Chi-Squared test. The BOW feature engineering method has extracted around 5000 features. Using the Chi-Squared feature selection method, this feature set was reduced for 3000. The result of this experiment is presented in Table 4.6.

Table 4.6: Effect of Feature Selection

	Classifier	**Precision %**	**Recall %**	**F1 Score %**
Baseline Models with all BOW features	*Logistic Regression*	*70.57*	*70.44*	*70.44*
	SVM	*69.70*	*69.33*	*69.40*
Baseline Models with best BOW features selected by Chi-Squared	Logistic Regression	69.59	70.13	69.67
	SVM	68.89	68.66	68.74

Results show that this has reduced the accuracy of the baseline models. Even though the selected feature size was varied from 1000, 2000 and 4000, similar results were observed. Hence, it was decided not to go ahead with the feature selection step using Chi-Squared in this research.

4.6 Evaluation of Models

In this section the results of each model with regard to different feature sets will be discussed. The optimum parameters for each model-feature set combination was obtained using hyper parameter optimization. The class weights have been set to balanced when training these models.

The below tables, Table 4.7 and Table 4.8 show the results obtained by using the SVM and Logistic Regression models respectively. It can be observed that these models obtained their highest accuracy for the feature set with TF-IDF. This could possibly be due to the fact that different terms are important in different attitude classes. The second highest is by using the BOW feature extraction technique. Models with these feature extraction techniques have surpassed the accuracy values of baseline models. The lowest accuracy values were obtained for the models using the Bigram feature set. This could be due to the fact that the frequency of 2 terms occurring together and important enough to create variance among the data is not so common in this context.

Table 4.7: Results of SVM Model

Feature Set	Precision %	Recall %	F1 Score %
Baseline Random Model with BOW	*36.03*	*35.90*	*35.90*
Baseline Logistic Regression Model with BOW	*70.57*	*70.44*	*70.44*
Baseline SVM Model with BOW	*69.70*	*69.33*	*69.40*
BOW	70.95	68.54	69.48
Bigram	64.99	62.57	63.11
POS	69.34	68.04	68.44
TFIDF	73.04	73.38	72.82

Table 4.8: Results of Logistic Regression Model

Feature Set	**Precision %**	**Recall %**	**F1 Score %**
Baseline Random Model with BOW	*36.03*	*35.90*	*35.90*
Baseline Logistic Regression Model with BOW	*70.57*	*70.44*	*70.44*
Baseline SVM Model with BOW	*69.70*	*69.33*	*69.40*
BOW	71.99	71.71	71.80
Bigram	65.84	65.9	65.74
POS	69.92	69.69	69.76
TFIDF	73.55	73.76	73.55

Following graph in Figure 4.1 shows the feature importance as recognized by the Logistic Regression with TF-IDF feature set. The absolute coefficient values were used to derive this graph. These feature important values complement the findings of EDA in Section 3.3. However, most of the important features belong to the Behavioural and Cognitive classes. Comparatively, the model has not been able to identify many significant features for the Affective class.

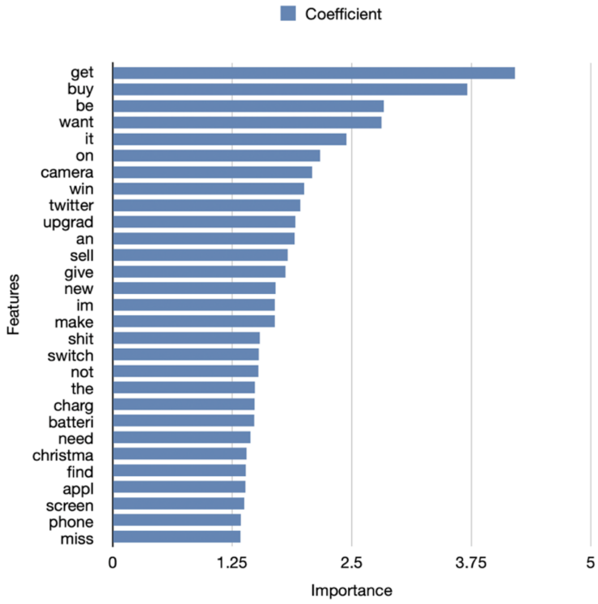


Figure 4.1: Feature Importance Graph (Logistic Regression with TF-IDF features)

The following experiments were conducted to present the effect of hyper-parameter optimization and using balanced class weights on Logistic Regression when dealing with imbalance classes. The different parameters used by the Logistic Regression base model with BOW and Logistic Regression with TF-IDF features have been listed in Table 4.9.

Table 4.9: Baseline and Optimized Hyper Parameter values- Logistic Regression

Parameter	**Values in Baseline Logistic Regression with BOW**	**Optimized Values in Logistic Regression with TF-IDF features**
C (Strength)	0.59	2.07857
Solver	liblinear	liblinear
Maximum Iterations	500	1000
Tolerance	4.7e-05	10.315651000002603e-05

According to the results in Table 4.10, it is evident that models can be optimized with different parameters. In the baseline model, the feature set used was BOW. Results show that those parameters can be further optimized when using a different feature set such as TF-IDF.

Table 4.10: Effect of Parameter Tuning on Logistic Regression with TF-IDF Features

	Precision %	**Recall %**	**F1 Score %**
With the parameters of baseline model	72.32	72.74	72.25
With hyper parameter optimization	73.55	73.76	73.55

Results in Table 4.11 shows how the class weights impact the behaviour of the model Logistic Regression with TF-IDF Features.

Table 4.11: Effect of Balanced Class weights on Logistic Regression with TF-IDF Features

	Precision %	**Recall %**	**F1 Score %**
weights= None	72.42	72.80	71.96
weights= ‘balanced’	73.55	73.76	73.55

In order to further examine this condition, the following confusion matrices were extracted from the fold with highest F1 score. It should be noted that the total counts in the test set will vary as these results are extracted from K-Fold cross validation.

class_weights= None

		Predicted		
		Affective	**Behavioural**	**Cognitive**
Actual	**Affective**	110	60	53
	Behavioural	21	283	60
	Cognitive	27	46	348

Table 4.12: Confusion Matrix Logistic Regression with TF-IDF Features (class_weights= None)

class_weights= ‘balanced’

		Predicted		
		Affective	**Behavioural**	**Cognitive**
Actual	**Affective**	134	39	41
	Behavioural	43	299	46
	Cognitive	44	38	325

Table 4.13: Confusion Matrix Logistic Regression with TF-IDF Features (class_weights= ‘balanced)

Affective class is the minority class in our problem. When there is little information available on a class, a model is expected to produce erroneous predictions for that class. This results in a low F1 score even if the rest of the classes are predicted accurately. This phenomenon is visible in the results shown in Table 4.12 where the highest number of misclassifications are from the Affective class. However, in Table 4.13, even though the misclassifications of other classes have slightly gone up, the overall F1 score has increased. It should be noted that using balanced class weights depends largely on the problem domain. However, instead of using balanced weights, the weights can be assigned differently based on the problem.

Results of the rest of the models have been presented in Table 4.14. Some models have been able to surpass the accuracy values of the baseline SVM and Logistic Regression model while all the models have been able to beat the baseline Random Model. But none of the classifiers have been able to improve the accuracy obtained using the Logistic Regression with TF-IDF features.

Interestingly, unlike Logistic Regression and SVM which had obtained the highest accuracy values using the TF-IDF feature set, Multinomial Naive Bayes Model, Random Forest and Simple Neural Network have obtained the highest accuracy for the BOW feature set.

Table 4.14: Results of Multinomial Naive Bayes, Random Forest, Stacking, Random Subspace and Simple Neural Network

Classifier	Feature Set	Precision %	Recall %	F1 Score %
Baseline Random Model	BOW	36.03	35.90	35.90
Baseline Logistic Regression	BOW	70.57	70.44	70.44
Baseline SVM Model	BOW	69.70	69.33	69.40
Multinomial Naive Bayes Model	BOW	70.66	71.08	69.51
	Bigram	67.24	67.80	65.89
	POS	67.39	67.74	67.51
	TFIDF	69.05	69.51	69.16
Random Forest	BOW	68.71	67.72	68.04
	Bigram	63.24	60.07	60.88
	POS	67.49	66.14	66.54
	TFIDF	67.70	67.67	67.59
Random Subspace (SVM as the base learner)	BOW	69.71	66.44	67.22
	Bigram	63.39	59.89	59.99
	POS	68.55	67.01	67.47
	TFIDF	71.95	72.09	70.99
Stacking Model (SVM and Logistic Regression as base learners)	BOW	71.27	71.57	71.33
	Bigram	65.34	66	65.5
	POS	69.94	70.34	70.04
	TFIDF	73.02	73.42	72.99
Simple Neural Network	BOW	70.52	70.07	70.23
	Bigram	65.76	63.30	64.18
	POS	67.52	66.63	66.98
	TFIDF	67.18	68.26	67.54

The best performing models from the above experiments Logistic Regression and SVM were selected as base learners for the ensemble implementation. Using the Random Subspace classifier as shown in Table 4.14, shows lower accuracy values than the standalone SVM model. Similar outcome was observed when the Logistic Regression model was used as the base learner for Random Subspace. Therefore, the results have not been included here. For the Stacking ensemble method, SVM and Logistic regression was used. The highest result obtained using TF-IDF features in this model is higher than SVM and lower than Logistic Regression. Stacking similar classifiers together was also evaluated (e.g.: SVM+SVM). This showed a behaviour similar to the Random Subspace, where the Stacking ensemble model accuracy values were lower than the accuracy values of their standalone base learners.

4.6.1 Word Embeddings

In order to further improve the model, word embedding features were tried out on the best performing models obtained from the above experiments. Results in Table 4.15 show how the Logistic Regression and SVM have performed with Word2Vec and FastText features.

Table 4.15: Results of Using Word Embedding features with SVM and Logistic Regression Classifiers

Classifier	Feature Set	Precision %	Recall %	F1 Score%
Baseline Random Model	*BOW*	*36.03*	*35.90*	*35.90*
Baseline Logistic Regression	*BOW*	*70.57*	*70.44*	*70.44*
Baseline SVM Model	*BOW*	*69.70*	*69.33*	*69.40*
Logistic Regression	Word2Vec	70.67	71.02	70.74
	FastText	64.72	64.35	64.50
SVM	Word2Vec	70.8	69.67	70.08
	FastText	67.34	66.22	66.59

It can be observed that none of the models have been able to surpass the results of the TF-IDF feature sets. However, the SVM model with Word2Vec features has been able to obtain a higher accuracy than SVM with BOW features.

As the Word2Vec feature set couldn't improve the results significantly, the FastText technique was experimented. Using FastText features in the best performing models gave a poor accuracy.

In addition to calculating word embeddings, FastText library also comes with a classifier. Fasttext[16] wrapper can be used to implement the FastText text classifier in Python. The same wrapper also provides a parameter optimization function to come up with an optimum model for a given problem. Table 4.16 shows the results obtained by FastText using this method with holdout validation.

Table 4.16: Results of FastText Text Classification (Multinomial Logistic Regression)

Feature Set	Precision %	Recall %	F1 Score %
Baseline Random Model with BOW	*36.03*	*35.90*	*35.90*
Baseline Logistic Regression Model with BOW	*70.57*	*70.44*	*70.44*
Baseline SVM Model with BOW	*69.70*	*69.33*	*69.40*
FastText	73.62	74.11	73.54

However, this method does not expose the parameters used to build the model. According to FastText documents, the classifier uses Multinomial Logistic Regression. Even though in our experiment, the Logistic Regression with FastText features was hyper-parameter optimized, it couldn’t reach the same performance. But

[16] https://fasttext.cc/

this proves that the Logistic Regression with FastText has the capability to be further optimized with the right resources.

4.6.2. Deep Learning Models

As word embedding features failed to improve the accuracy values of the best performing models, those features were tested on Deep learning models. The following Table 4.17 shows the results of the experiment using Word2Vec and FastText features on Deep Learning models.

Table 4.17: Results of Deep Learning Model CNN and RNN-LSTM using Word Embedding features

Classifier	**Feature Set**	**Precision %**	**Recall %**	**F1 Score%**
Baseline Random Model	*BOW*	*36.03*	*35.90*	*35.90*
Baseline Logistic Regression	*BOW*	*70.57*	*70.44*	*70.44*
Baseline SVM Model	*BOW*	*69.70*	*69.33*	*69.40*
CNN	Word2Vec	74.63	74.73	73.40
	FastText	71.26	70.76	70.90
RNN-LSTM	Word2Vec	73.24	73.34	72.42
	FastText	72.07	71.46	70.62

Deep learning with word embedding features have achieved higher accuracy results than all baseline models. In fact, CNN and RNN-LSTM have reached the same accuracy values as Logistic Regression and SVM with TF-IDF features. A significant improvement of results can be seen using FasteText features with Deep learning models as opposed to the previous experiment using SVM and Logistic Regression. Among the results in Table 4.17, CNN has higher results than RNN-LSTM and it also has surpassed one of the previously best performed models, SVM classifier with TF-

IDF features. This shows that word embedding features when used with Deep learning models performs well for the problem at hand.

Furthermore, the models were evaluated by using pre-trained word vectors of Google News and Glove Twitter vectors (dimension= 25 and 200). The accuracy given by using pre-trained vectors did not show any improvement and they were similar to the values in Table 4.17. Hence the results were not included here.

In the Table 4.18, the Precision, Recall and F1 Score recorded by the best performing models have been presented. For this comparison, the hold out method was used. In order to make an unbiased comparison, same train- test sets have been used. (It should be noted that since the hold out method was used, the scores of Logistic Regression are different to the previous average values of K-Fold cross validation.)

Table 4.18: Classification report Logistic Regression Vs CNN

	Precision %		**Recall %**		**F1 Score %**	
	Logistic Regression	**CNN**	**Logistic Regression**	**CNN**	**Logistic Regression**	**CNN**
A	60.51	71.55	56.19	39.52	58.27	50.92
B	79.77	78.96	75.82	79.40	77.75	79.18
C	77.35	72.49	83.22	87.82	80.18	79.42
Weighted Average	74.72	74.63	74.93	74.73	74.74	73.40

Both the models display very similar results. Looking closely at the weighted averages of Precision and Recall it can be observed that Logistic Regression has slightly higher values. The reason for overall low weighted F1 score of CNN model is its low Recall value recorded for the minority class of Affective. Even though the class weights are balanced in both the models, the Logistic Regression model has handled the class imbalance problem better than the CNN.

4.7 Error Analysis

The following error analysis was done for the Logistic Regression model with TF-IDF features using the holdout method. Table 4.19 contains a few examples of misclassified tweets.

Table 4.19: Error Analysis

Index	**Tweet**	**Actual Class**	**Predicted Class**
1	*“iphone with the nice camera here i come”*	Behavioural	Cognitive
2	*“I got the iPhone 11 and took this stunning photo of Olive and like truly that’s the only reason I bought this phone pic.twitter.com/e4T9nR9big”*	Cognitive	Behavioural
3	*“ok also is it weird to kinda not wanna get rid of my “old” phone like when i woke up this morning i didnt know id have a new one by the end of the day smh i went through 2 bfs with my iphone x it has sentimental value damn its seen me cry a lot and ugly laugh ok ill stop now”*	Affective	Cognitive
4	*“My iphone been acting weird lately”*	Cognitive	Affective
5	*“my teacher has the iPhone X if y’all hear sumn abt a missing phone keep quiet https://twitter.com/haramores/status /1185987142024982528/video/1 ...”*	Behavioural	Affective
6	*“Also there are Android phones that are even more expensive that some iPhone models... But buying an apple product gives you a "privileged status" which is, stupid.”*	Affective	Behavioural

In Tweet 1 and Tweet 2, even though it has originally been annotated as Behavioral and Cognitive respectively, those particular tweets can belong to both the classes Behavioral and Cognitive. But since our solution only labels a tweet into one class, these were considered as misclassifications. As discussed in the EDA (Section 3.3), Affective attitudes toward the attitude object are often associated with another entity. Similarly, in Tweet 3, the attitude is associated with memories of the user. When the Affective attitude towards the product is expressed implicitly, the model might not be able to catch it. In Tweet 4, the word "weird" is a term usually expressing an Affective attitude. But here it has been used to express the performance of the product. Additionally, the tweet doesn't contain any Cognitive words such as product specifications. Tweet 5 implicitly expresses a behaviour of stealing. Even though a human annotator has been able to identify this, the model has failed to recognise implicit expression of tweets. In Tweet 6, even though an Affective attitude is expressed, the word "buy" is a strong expression of Behavioural attitude.

4.8 Summary

In this section the best results obtained by each model has been summarised into Table 4.20.

Table 4.20 Summary of Best Results

Classifier	Precision %	Recall %	F1 Score %
Logistic Regression	73.55	73.76	73.55
CNN	74.63	74.73	73.40
Stacking (with SVM and Logistic Regression)	73.02	73.42	72.99
SVM	73.04	73.38	72.82
RNN-LSTM	73.24	73.34	72.42
Random Subspace (with SVM)	71.95	72.09	70.99
NN	70.52	70.07	70.23
Multinomial Naive Bayes	70.66	71.08	69.51
Random Forest	68.70	67.72	68.04

The table is arranged based on the descending order of F1 scores of the classifiers. Accordingly, Logistic Regression and CNN have given the best results in this study.

Even though a list of preprocessing steps was curated based on their performance on the baseline model, it was observed that some techniques had a different effect on other classification algorithms. Hence, after careful evaluation, the following set of preprocessing steps were selected as the optimum preprocessing techniques for this solution.

1. Remove Twitter tags
2. Remove punctuations
3. Lowercasing
4. Tokenization
5. Stemming
6. Lemmatization

The feature sets and the important parameters used in achieving the above listed classifiers is presented in Table 4.21. The above listed preprocessing techniques have been applied on all these classifiers.

Table 4.21: Hyper-parameters of Best Performing Models

Classifier	Feature set	Hyper-parameters
Logistic Regression	TF-IDF	Maximum iterations= 1000, Solver= Liblinear, C (strength)= 2.08
CNN	Word2Vec	Feature size= 200, Batch size= 10, Epochs=20, with Early stopping (for validation loss and patience=3), Optimizer= RMSprop, Loss= Categorical Crossentropy
Stacking (with SVM and Logistic Regression)	TF-IDF	Final estimator=Logistic Regression
SVM	TF-IDF	C= 30.48, Kernal= rbf
RNN-LSTM	Word2Vec	Feature size= 200, Batch size= 10, Epochs=20, with Early stopping (for validation loss and patience=3), Optimizer= RMSprop, Loss= Categorical Crossentropy
Random Subspace (with SVM)	TF-IDF	Bootstrap=True, Number of Estimators= 500, Maximum Sample=1000
NN	BOW	Batch size= 4, Epochs=10, Optimizer= RMSprop, Loss= Categorical Crossentropy
Multinomial Naive Bayes	BOW	Alpha=1, Fit Prior=False
Random Forest	BOW	Criterion= Entropy, Maximum depth=20, Maximum number of features='sqrt'

5. CONCLUSION

It was observed that in the domain of sentiment analysis less focus has been given for classifying attitudes beyond the binary classification of polarity. In order to solve this identified problem, it was proposed to make use of the ABC model of attitude introduced in consumer psychology.

Since a suitable dataset for this problem doesn't exist, a dataset was created by extracting Tweets and annotating them manually into Affective, Behavioural, Cognitive, Neutral and Advertisement classes. 10,000 such tweets were extracted and only the attitudinal Tweets (i.e. Affective, Behavioural and Cognitive) were used in the classification process. Exploratory data analysis was conducted to better understand the dataset and design the methodology.

In designing the methodology, techniques that had performed well in previous studies were given prominence. Out of such various preprocessing techniques that were evaluated in this study, combination of removing twitter tags, removing punctuations, lowercasing, tokenization, stemming and lemmatization were selected as the best set of preprocessing steps for this research. N-grams (N=1 and N=2), POS, TF-IDF, Word2Vec, FastText feature extracting techniques were experimented for this problem. Among them, TF-IDF and Word2Vec methods recorded the best accuracy.

Even though only supervised machine learning algorithms were considered in this study, a wide range of models were implemented and evaluated. Out of these classification algorithms, the most promising results were recorded for Logistic Regression and CNN.

At the end of this research, the author was able to develop a system from data extraction to Sentiment classification based on the ABC model of attitude. This study showed the effectiveness of using existing technologies to solve the sentiment classification based on ABC attitude model. Additionally, this study has contributed a dataset for the community that can be utilized for future research in the same domain.

5.1 Future Improvements

Following list presents a list of possible future enhancements and improvements to this research problem,

- The biggest limitation faced during this research is the limited dataset. A larger dataset will definitely add more value to this study and improve the classification accuracy.
- Further preprocessing of data using dictionaries dedicated to classify emojis and identify social media terms such as slangs.
- Incorporating sentiment features by calculating a sentiment score for each term using a tool such as LIWC[17].
- It was observed that some tweets have multiple sentiments. Even though this study focused on classifying into a single class, this can be further optimized to multi-label classification tasks with improved annotation.
- With the existing dataset, this can also be improved with up to 3 levels of classification by introducing the polarity of the attitude and classifying as attitudinal or non-attitudinal in addition to the ABC classes.

[17] https://iwc.wpengine.com

REFERENCES

[1] A. D'Andrea, F. Ferri, P. Grifoni and T. Guzzo, "Approaches, Tools and Applications for Sentiment Analysis Implementation", *International Journal of Computer Applications*, vol. 125, no. 3, pp. 26-33, 2015.

[2] W. Medhat, A. Hassan and H. Korashy, "Sentiment analysis algorithms and applications: A survey”, Ain Shams Engineering Journal, vol. 5, no. 4, pp. 1093-1113, 2014.

[3] A. Allport, Perspectives on Perception and Action. Bielefeld: Routledge, 1985.

[4] K. Ravi and V. Ravi, "A survey on opinion mining and sentiment analysis: Tasks, approaches and applications", Knowledge-Based Systems, vol. 89, pp. 14-46, 2015.

[5] L. Schiffman and J. Wisenblit, Consumer behavior, 11th ed. Upper Saddle River, New Jersey: Pearson Education, 2014.

[6] I. Asiegbu, D. Powei and C. Iruka, "Consumer Attitude: Some Reflections on Its Concept, Trilogy, Relationship with Consumer Behavior, and Marketing Implications", European Journal of Business and Management, vol. 4, no. 13, 2012.

[7] Eagly, A. H., & Chaiken, S. (1993). The psychology of attitudes. Harcourt Brace Jovanovich College Publishers.

[8] C. Haugtvedt, Handbook of consumer psychology. New York: Psychology Press, 2008.

[9] D. Coon and J. Mitterer, Psychology: A Journey, 5th ed. Cengage Learning, 2013.

[10] G. Angiani, L. Ferrari, F. Magliani, P. Fornacciari, E. Iotti, T. Fontanini, S. Manicardi, "A Comparison between Preprocessing Techniques for Sentiment Analysis in Twitter", Knowledge Discovery on the WEB, vol. 1748, 2016.

[11] U. Fayyad, G. Piatetsky-Shapiro and R. Uthurusamy, "Summary from the KDD-03 panel -- Data Mining: The Next 10 Years", ACM SIGKDD Explorations Newsletter, vol. 5, no. 2, pp. 191-196, 2003.

[12] D. Effrosynidis, S. Symeonidis and A. Arampatzis, "A Comparison of Pre-processing Techniques for Twitter Sentiment Analysis", Research and Advanced Technology for Digital Libraries, vol. 10450, pp. 394-406, 2017.

[13] A. Celikyilmaz, D. Hakkani-Tur and J. Feng, "Probabilistic model-based sentiment analysis of Twitter messages", in 2010 IEEE Spoken Language Technology Workshop, 2010.

[14] N. Altrabsheh, M. Cocea and S. Fallahkhair, "Sentiment Analysis: Towards a Tool for Analysing Real-Time Students Feedback", 2014 IEEE 26th International Conference on Tools with Artificial Intelligence, 2014.

[15] Z. Jianqiang, "Pre-processing Boosting Twitter Sentiment Analysis?", in 2015 IEEE International Conference on Smart City/SocialCom/SustainCom (SmartCity), 2015.

[16] H. Saif, Y. He and H. Alani, "Semantic Sentiment Analysis of Twitter", The Semantic Web – ISWC 2012, vol. 7649, pp. 508-524, 2012.

[17] K. Gao, S. Su and J. Wang, "A sentiment analysis hybrid approach for microblogging and E-commerce corpus", in 7th International Conference on Modelling, Identification and Control (ICMIC), 2015.

[18] G. Forman, "An extensive empirical study of feature selection metrics for text classification", The Journal of Machine Learning Research, vol. 3, pp. 1289-1305, 2003.

[19] M. Porter, "An algorithm for suffix stripping", Program, vol. 14, no. 3, pp. 130-137, 1980.

[20] C. Paice, "Another stemmer", ACM SIGIR Forum, vol. 24, no. 3, pp. 56-61, 1990.

[21] M. Hagenau, M. Liebmann, M. Hedwig and D. Neumann, "Automated News Reading: Stock Price Prediction Based on Financial News Using Context-Specific Features", in 45th Hawaii International Conference on System Sciences, 2012.

[22] H. Gupta and N. Hasteer, "Utility of Corpus based Approach in the Recognition of Opinionated Text", Indian Journal of Science and Technology, vol. 9, no. 44, 2016.

[23] J. Barry, "Sentiment Analysis of Online Reviews Using Bag-of-Words and LSTM Approaches", Artificial Intelligence and Cognitive Science, vol. 2086, pp. 272-274, 2017.

[24] I. Habernal, T. Ptáček and J. Steinberger, "Sentiment Analysis in Czech Social Media Using Supervised Machine Learning", Computational Approaches to Subjectivity, Sentiment and Social Media Analysis, vol. 4, pp. 65–74, 2013.

[25] X. Glorot, A. Bordes and Y. Bengio, "Domain adaptation for large-scale sentiment classification: A deep learning approach", International Conference on Machine Learning, vol. 28, 2011.

[26] T. Mikolov, G. Corrado, K. Chen and J. Dean, "Efficient Estimation of Word Representations in Vector Space", International Conference on Learning Representations, 2013.

[27] J. Pennington, 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.

[28] S. Rosenthal, A. Ritter, P. Nakov and V. Stoyanov, "DAEDALUS at SemEval-2014 Task 9: Comparing Approaches for Sentiment Analysis in Twitter", in Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014), 2014.

[29] J. Zhao, M. Lan and T. Zhu, "ECNU: Expression- and Message-level Sentiment Orientation Classification in Twitter Using Multiple Effective Features", in Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014), 2014.

[30] A. Duric and F. Song, "Feature selection for sentiment analysis based on content and syntax models", Decision Support Systems, vol. 53, no. 4, pp. 704-711, 2012.

[31] M. Asghar, A. Khan, S. Ahmad and F. Kundi, "A Review of Feature Extraction in Sentiment Analysis", Journal of Basic and Applied Research International, vol. 4, pp. 181-186, 2014.

[32] D. Jurafsky and J. Martin, Speech and language processing. India: Dorling Kindersley Pvt, Ltd., 2017.

[33] T. Hardeniya and D. Borikar, "Dictionary Based Approach to Sentiment Analysis - A Review", International Journal of Advanced Engineering, Management and Science (IJAEMS), vol. 2, no. 5, pp. 317-322, 2016.

[34] A. Moreno-Ortiz and J. Fernández-Cruz, "Identifying Polarity in Financial Texts for Sentiment Analysis: A Corpus-based Approach", Procedia - Social and Behavioral Sciences, vol. 198, pp. 330-338, 2015.

[35] D. Tang, F. Wei, B. Qin, T. Liu and M. Zhou, "Coooolll: A Deep Learning System for Twitter Sentiment Classification", in 8th International Workshop on Semantic Evaluation (SemEval 2014), Dublin, Ireland, 2014, pp. 208–212.

[36] K. Gao, S. Su and J. Wang, "A sentiment analysis hybrid approach for microblogging and E-commerce corpus", in 7th International Conference on Modelling, Identification and Control (ICMIC), 2015.

[37] C. Hutto and E. Gilbert, "VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text", in Proceedings of the Eighth International AAAI Conference on Weblogs and Social Media, At Ann Arbor, 2015.

[38] A. Kumar and R. Rani, "Sentiment Analysis Using Neural Network", in 2nd International Conference on Next Generation Computing Technologies, Dehradun, India, 2016.

[39] J. Schmidhuber, "Deep learning in neural networks: An overview", Neural Networks, vol. 61, pp. 85-117, 2015.

[40] C. Santos and M. Gatti, "Deep Convolutional Neural Networks forSentiment Analysis of Short Texts", in International Conference on Computational Linguistics, Dublin, Ireland, 2014, pp. 69-78.

[41] G. Paltoglou and M. Thelwall, "Twitter, MySpace, Digg", ACM Transactions on Intelligent Systems and Technology, vol. 3, no. 4, pp. 1-19, 2012.

[42] G. Wang, J. Sun, J. Ma, K. Xu and J. Gu, "Sentiment classification: The contribution of ensemble learning", Decision Support Systems, vol. 57, pp. 77-93, 2014.

[43] M. Whitehead and L. Yaeger, "Sentiment Mining Using Ensemble Classification Models", Innovations and Advances in Computer Sciences and Engineering, pp. 509-514, 2009.

[44] Y. Su, Y. Zhang, D. Ji, Y. Wang and H. Wu, "Ensemble Learning for Sentiment Classification", Chinese Lexical Semantics, pp. 84-93, 2013.

[45] T. Chalothom and J. Ellman, "Simple Approaches of Sentiment Analysis via Ensemble Learning", Information Science and Applications, pp. 631-639, 2015.

[46] C. Catal and M. Nangir, "A sentiment classification model based on multiple classifiers", Applied Soft Computing, vol. 50, pp. 135-141, 2017.

[47] N. Silva, E. Hruschka and E. Hruschka, "Biocom Usp: Tweet Sentiment Analysis with Adaptive Boosting Ensemble", in Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014), 2014.

[48] N. da Silva, E. Hruschka and E. Hruschka, "Tweet sentiment analysis with classifier ensembles", Decision Support Systems, vol. 66, pp. 170-179, 2014.

[49] T. Nasukawa and J. Yi, "Sentiment analysis: Capturing Favorability Using Natural Language Processing ", in Proceedings of the international conference on Knowledge capture - K-CAP '03, 2003.

[50] M. Sokolova and G. Lapalme, "A systematic analysis of performance measures for classification tasks", Information Processing & Management, vol. 45, no. 4, pp. 427-437, 2009.

[51] A. Muhammad, N. Wiratunga and R. Lothian, "Contextual sentiment analysis for social media genres", Knowledge-Based Systems, vol. 108, pp. 92-101, 2016.

[52] A. Ceron, L. Curini and S. Iacus, "iSA: A fast, scalable and accurate algorithm for sentiment analysis of social media content", Information Sciences, vol. 368, pp. 105-124, 2016.

[53] M. Bouazizi and T. Ohtsuki, "A Pattern-Based Approach for Multi-Class Sentiment Analysis in Twitter", IEEE Access, vol. 5, pp. 20617-20639, 2017.

[54] R. CBalabantaray, M. Mohammad and N. Sharma, "Multi-Class Twitter Emotion Classification: A New Approach", International Journal of Applied Information Systems, vol. 4, no. 1, pp. 48-53, 2012.

[55] A. Mohammed, A. Paschke and D. Monett, "Emotion Level Sentiment Analysis: The Affective Opinion Evaluation", in Proceedings of the 2th Workshop and Challenge on Emotions, Modality, Sentiment Analysis and the Semantic Web, vol. 1613, 2016.

[56] T. Dalgleish and M. Power, Handbook of cognition and emotion. Hoboken, N.J.: Wiley, 2005.

[57] R. Plutchik, "A general psychoevolutionary theory of emotion", Theories of emotion, vol. 1, pp. 3-31, 1980.

[58] J. Bollen, H. Mao and A. Pepe, "Modeling Public Mood and Emotion: Twitter Sentiment and Socio-Economic Phenomena", in Proceedings of the Fifth International AAAI Conference on Weblogs and Social Media, 2011.

[59] D. McNair, M. Loor and L. Droppleman, Profile of Mood States. 1971.

[60] C. Baziotis, N. Pelekis and C. Doulkeridis, "Deep LSTM with Attention for Message-level and Topic-based Sentiment Analysis", Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017), 2017

[61] J. Pennington, 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.

[59] D. McNair, M. Loor and L. Droppleman, Profile of Mood States. 1971.

[60] A. Sandhe and A. Joshi, "Consumers' Attitude towards Organic Food Products in Vadodara – An Exploratory Study", Pacific Business Review Internationa, vol. 10, no. 1, pp. 32-40, 2017.

[61] C. Chen and C. Cheng, "How online and offline behavior processes affect each other: customer behavior in a cyber-enhanced bookstore", Quality & Quantity, vol. 47, no. 5, pp. 2539-2555, 2012.

[62] I. Asiegbu, D. Powei and C. Iruka, "Consumer Attitude: Some Reflections on Its Concept, Trilogy, Relationship with Consumer Behavior, and Marketing Implications", European Journal of Business and Management, vol. 4, no. 13, pp. 38-50, 2012.

[63] K. Chiu, C. Chen, H. Lin, Y. Wu and L. Shih, "A Preliminary Study on Product Design of Emotional Appeal by Canonical Correlation Analysis of Public Attitudes towards Water-Saving Equipment Based on ABC Model", 2019 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM), 2019.

[64] T. Zhu and Y. Xu, "A Survey of Hainan Residents' Attitudes towards Migratory Group-Based on the ABC Model of Attitude", Proceedings of the 6th International Conference on Management Science and Management Innovation (MSMI 2019), vol. 84, 2019. Available: 10.2991/msmi-19.2019.5 [Accessed 20 May 2020].

[65] F. van Harreveld, H. Nohlen and I. Schneider, "The ABC of Ambivalence: Affective, Behavioral, and Cognitive Consequences of Attitudinal Conflict", Advances in Experimental Social Psychology, pp. 285-324, 2015. Available: 10.1016/bs.aesp.2015.01.002 [Accessed 20 May 2020].

[66] C. Baziotis, N. Pelekis and C. Doulkeridis, "Deep LSTM with Attention for Message-level and Topic-based Sentiment Analysis", Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017), 2017

[67] J. Pennington, 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.