

Menoufia University
Faculty of Computers and Information
Department of Computer Science



Sentiment Analysis Enhancement of Micro-blogging Data System

Thesis Submitted in Accordance with Partial Fulfillment of Requirements of
Menoufia University for the Degree of Master in Computers and Information

In
[Computer Science]

By
Raghda Sherif Khaled Elnadrey
B.SC in Computer Science

Supervised By

Prof./Dr. Ashraf B. El-Sisi
ashraf el sisi
Professor of Computer Science
Faculty of Computers and Information,
Menoufia University
[Computer Science]

Dr. Walid Said
Walid Said
Lecturer of Computer Science
Faculty of Computers and Information,
Menoufia University
[Computer Science]

2022

AUTHOR BIBLIOGRAPHY

Name: Raghda Sherif Khalid Elnadrey

Occupation: Instructor of Computer Science

Occupation Place: E-learning, Data Camp company

Date of Birth: 22 Jun 1989

Educational Degrees: Very Good

Educational Institution: Faculty of Computers and Information

Date of Educational: 17 October 2020

Registration Date: JUN 2022

ACKNOWLEDGEMENT

I give my profound thanks to **Allah** for the opportunity and the strength to accomplish this thesis.

I want to thank my supervisors, **Prof. Ashraf El-Sisi** and **Dr Walid Saied**, for their help and support during my work in creating a very inspiring research environment. They have been helpful with background information and have continually encouraged me and helped me with comments to complete my work. They always had time to discuss new ideas and give feedback to tackle research's challenging problems.

I want to thank my family, who were constant sources of rising and supporting my spirit. Especially thanks go out to my father, my mother, my husband, my brother, and my sisters for support and encouragement.

Finally, I would like to thank my faculty members and staff, department, and colleagues who gave me uncountable support and encouragement.

ABSTRACT

Sentiment analysis (SA) is a process based on natural language processing and methodology for estimating the positive or negative emotions expressed by a passage of text. Sentiment analysis is often used to categorize customer feedback messages, product reviews, social media comments, or public social news. Analyzing the feelings of texts written in social media depends on special features and on semantic information for sentences that require many methodologies and processes under text analysis. This kind of information is valuable to everyone. Recently, several scholars have shown interest in using features for text and emoji classification tasks. Therefore, we have started our first steps in analyzing definitive texts and Emojis to classify micro-blogs.

Our approach has steps and results based on experience, implementation, testing, and comparison with other results with the same characteristics and two different datasets. In these approaches, we will apply how to make sense of the sentiment expressed in comments on the Microblogging web(e.g. Twitter). Our approaches are: 1) Design a scalable, efficient, and full-step analytic system. 2) Implementation of the system in five main paths (data collection, preprocessing, feature extraction and classification of its basic features, measurement and evaluation of the results performance). 3) Analyzing emojis as essential features of social media content, especially Twitter. 4) Training our models to analyze texts and also analyze emojis together in more than one model and extract the results of this analysis between texts and emojis with different expressiveness. Five basic extraction features are used in our models (Bag of Words (BOWs), TF-IDF (Term Frequency - Inverse Document Frequency), Nigram and Word embedding) to extract sentiment text features and analyze emojis.

We employ machine learning to predict the sentiment of a review based on the words used in the review. We use five classifications and evaluate its performance in a few different ways. Five classifiers (Logistic Regression (LR), Supportive Vector Machine (SVM), Naïve Bayes (NB), Random Forest (RF), and XGBoost) are used to compare the performance of different features selected for sentiment analysis.

Our suggested models have been created as follows: 1) A template for text-only analysis and testing of performance results. 2) A model for analyzing text and emojis and testing performance results. 3) Compare the results between these models and other models that use the same data sets.

The results show high performance using the Word2Vec approach with the XGBoost and Random Forest classification algorithms. Whereas, Naive Bayes is considered a minor performance. We will carry out an end-to-end sentiment analysis task based on how US airline passengers expressed their feelings on Twitter by the end of these approaches.

Keywords: - Sentiment Analysis, Emoji Analysis, Preprocessing, Features Extraction, Classification, Machine learning, NLP.

TABLE OF CONTENTS

AUTHOR BIBLIOGRAPHY	I
ACKNOWLEDGEMENT	II
ABSTRACT	3
TABLE OF CONTENTS	5
LIST OF FIGURES.....	7
LIST OF TABLES.....	8
LIST OF ABBREVIATIONS.....	9
CHAPTER 1 INTRODUCTION	10
1.1. OVERVIEW ON SENTIMENT ANALYSIS	10
1.2. THE PROBLEM FORMULATION	12
1.3. RESEARCH OBJECTIVE.....	12
1.4. RESEARCH CONTRIBUTION	13
1.5. THESIS ORGANIZATION	14
CHAPTER 2 BACKGROUND	15
2.1. MICROBLOGGING DATA	15
2.2. MICROBLOGGING CHARACTERETICS.....	17
2.3. SENTIMENT ANALYSIS.....	18
2.4. SENTIMENT ANALYSIS TECHNIQUES	20
2.4.1 EMOTION DETECTION	20
2.4.2 ASPECT BASED SENTIMENT ANALYSIS	21
2.4.3 BUILDING RESOURCES	22
2.4.4 MULTILINGUAGLE SENTIMENT ANALYSIS	22
2.4.5 EMOJIS ANALYSIS	23
CHAPTER 3 FEATURES EXTRACTION AND CLASSIFICATION	26
3.1. FEATURES EXTRACTION	26
3.1.1 BAG OF WORDS (BOW)	27
3.1.2 TERM FREQUENCY – INVERSE DOCUMENT FREQUENCY (TF-IDF)	30
3.1.3 N-GRAM	32
3.1.4 WORD EMBEDDING (WE)	33
3.1.4.1 DOC 2 VECTOR (D2V)	34
3.1.4.2 WORD 2 VECTOR (W2V)	34
3.2. SENTIMENT CLASSIFICATION APPROACHES	36

3.2.1 XGBOOST.....	38
3.2.2 LOGISITIC REGRESSION (LR)	39
3.2.3 RANDOM FOREST	40
3.2.4 SUPPORT VECTOR MACHINE	42
3.2.5 NAÏVE BAYES.....	44
CHAPTER 4 PROPOSED SENTIMENT ANALYSIS SYSTEM	49
4.1. SENTIMENT ANALYSIS SYSTEM.....	49
4.2. TEXT ANALYSIS MODEL.....	52
4.3. EMOJIS DATA AND LEXICON APPROACHES.....	54
4.4. TEXT AND EMOJIS ANALYSIS MODEL	55
4.5. EVALUATION METRICS.....	56
CHAPTER 5 EXPERIMENTAL RESULTS AND DISCUSSION	59
5.1. EXPERIMENTAL SETUP	59
5.2. DATASETS.....	60
5.2.1 AIRLINE DATASET OVERVIEW	60
5.2.2 EMOJIS DATASET OVERVIEW	60
5.3. EXPERIMENTAL RESULTS AND DICUSSION	61
5.3.1 TEXT ANALYSIS RESULTS	61
5.3.2 EMOJIS ANALYSIS RESULTS	64
5.3.3 EMOJIS AND TEXT RESULTS	65
CHAPTER 6 CONCLUSION AND FUTURE WORK.....	66
6.1. CONCLUSION	66
6.2. FUTURE WORK	67
REFERENCES	68
PUBLICATIONS.....	81

LIST OF FIGURES

Figure 3-1 Flowchart BOWs	29
Figure 3-2 Flowchart TF-IDF.....	31
Figure 3-3 The model of Word2Vector (W2V).....	35
Figure 3-4 Flowchart XGBoost.....	39
Figure 3-5 Flowchart Logistic Regression.....	40
Figure 3-6 SVM Classes.....	43
Figure 3-7 Flowchart SVM.....	44
Figure 3-8 Flowchart Naïve Bayes.....	45
Figure 4-1 Proposed Sentiment Analysis System.....	51
Figure 4-2 Data Visualization using Word Cloud.....	53

LIST OF TABLES

Table 2-1 Same Sentences with different Emojis.....	24
Table 3-1 Feature Methods (Characteristic and Limitations).....	36
Table 3-2 Difference between Automated and Rule-Based.....	37
Table 3-3 Comparison between Classification Methods.....	47
Table 4-1 Convert ASCII Emojis to Emoticons Symbols.....	55
Table 4-2 Public Six Emojis.....	55
Table 5-1 Airline Dataset Sentiment Value	60
Table 5-2 Example of Emojis Analysis	61
Table 5-3 Evaluation Metrics Using BOW Features.....	62
Table 5-4 Evaluation Metrics Using TF-IDF Features.....	62
Table 5-5 Evaluation Metrics using N-Gram Features	62
Table 5-6 Evaluation Metrics using Word2Vec Features.....	63
Table 5-7 Evaluation Metrics using Doc2Vec Features.....	63
Table 5-8 Results of T (Text only), and TE (Text and Emojis) Analysis.....	63
Table 5-9 Resultd of Text analysis and emojis analysis using dataset2	64

LIST OF ABBREVIATIONS

Symbol	Term
BOWs	Bag Of Words
DA	Data analysis
DC	Data Collection
DM	Data Mining
EL	Emoticon Lexicon
FE	Features Extraction
IR	Information Retrieval
LR	Logistic Regression
NBs	Naïve Bayes
OM	Opinion Mining
Pp	Preprocessing
RS	Recommendation System
SA	Sentiment Analysis.
SC	Sentiment Classification.
SL	Sentiment Lexicon
SVM	Support Vector Machine
TC	Twitter Classification
TR	Text Representation
TA	Twitter analytics
TF- IIDF	Term Frequency- Inverse Document Frequency

Chapter 1

INTRODUCTION

1.1. Overview of Sentiment Analysis

Sentiment analysis, also called opinion mining, is understanding an author's opinion about a subject. *"What is the emotion or opinion of the user of the text about the subject discussed?"* Depending on the context, a sentiment analysis system usually has three elements: First is the opinion or emotion. An opinion (also called "polarity") can be positive, neutral or negative. Emotion could be qualitative (like joy, surprise, or anger) or quantitative (like rating a movie on a scale from 1 to 10). The second element is the subject being talked about, such as a book, a movie, or a product. Sometimes one opinion could discuss multiple aspects of the same subject. For example: *"The camera on this phone is great, but its battery life is rather disappointing"*. The third element is the opinion holder, or entity, expressing the opinion.

Sentiment analysis has many practical applications. Social media is not our only source of information, but we also find sentiment on forums, blogs, and the news. Most brands analyze these sources to enrich their understanding of how customers interact with their brand, what they are happy or unhappy about? What matters most to consumers?. Sentiment analysis is thus critical in brand monitoring, customer and product analytics and market research and analysis. Sentiment analysis tasks can be carried out at different levels of granularity. First is document level, when looking at a service's full review. Second is the sentence level, which refers to determining whether the opinions expressed in each sentence are positive, negative, or neutral. The last level of granularity is the aspect level. The aspect refers to expressing opinions about distinct features of a service. Recently, Microblogging data such as social media platforms helped users share their opinions on topics and events. According to the Forbes website, there are 2.5 million bytes of data each day, and Twitter has received

456,000 tweets every minute of the day. Users frequent social media because it provides the amount of freedom to express their opinion while protecting anonymity. Because of these, they cause tons of hate speeches, derogatory or discriminatory content targeting specific groups, and racist comments. For these reasons, it is essential to have an efficient way to predict user sentiments about public social events, services, and products. The target is to classify the text on social media using machine learning algorithms. In medical science, text classification is used to analyze and categorize reports such as hospital records and brief text in tweets. The government used textual sentiment analysis in blogs to find potential self-murder victims and terrorists. Microblogging makes applying techniques such as simple pattern matching, parsing, spelling, and knowledge reasoning using the semantic web is challenging. One of the popular applications for sentiment analysis is to solve the problems of significant airline problems to classify positive, negative, and neutral tweets. Twitter data was scraped from February 2015, and contributors were asked first to classify positive, negative, and neutral tweets, followed by categorizing negative reasons (such as "late flight" or "rude service").

We aim to analyze how travellers mentioned their feelings on Twitter in February 2015. It would be fascinating for airlines to use this free data to provide better service to their customers, and present the sentiment analysis system for performance investigation using Microblogging data. By use several preprocessing and feature extraction techniques to optimize the sentiment analysis. The impact of the application to preprocess and feature extraction approaches on enhancing the sentiment analysis system performance is compared and discussed. The applied feature extraction approaches are Bag of Words (BOW), Term Frequency-Inverse Document Frequency (TF-IDF), N-Gram, Word2Vector, and Doc2Vector.

The applied classification algorithms are XGBoost, Random Forest (RF), Logistic Regression (LR), support vector machine (SVM), and Naive Bayes (NB). Using the airline dataset selected from Twitter, the devolved sentiment analysis system has been evaluated Twitter.com as a popular Microblogging website.

Each tweet is 140 characters. Tweets are frequently used to express a tweet's emotion on a particular subject. Some firms poll Twitter for analyzing sentiment on a particular topic. The challenge is gathering all such relevant data and detecting and summarizing the overall sentiment on a topic. Sentiment Analysis is a natural language processing methodology for qualifying how positive or negative the emotion is expressed by segmental text and emojis. The sentiment of a text or emojis depends on the syntactic information of the sentences and assigns the texts and emojis sentiment.

1.2. The Problem Formulation

The problem formulation is:

- Analyze and design a scalable, efficient and fully functional Twitter data sentiment model to predict training data using Microblogging data.
- Solve the problem of the polarity of sentiment analysis depending on (Text and emojis) by using long lists of rules (Features and classifiers) approaches.
- The problem in sentiment analysis is classifying the polarity of a text at all documents, sentences, or feature/ aspect levels.
- Whether the expressed idea in a document, a sentence or an entity feature/ aspect is positive, negative, or natural.

1.3. Research Object

The research aim is to analyze data to visualize airline services. This study focuses on a specific sub-field called sentiment analysis using Microblogging data and predictive modeling. Unlike performances, where models are used to understand data, predictive modeling is focused on developing models that make the most accurate predictions and explaining why predictions are made. This study classified twitter data into feedback using machine learning and natural language processing (NLP) with the help of the programming language Python and different machine learning algorithms.

The research objects are:

- Implement an algorithm for automatically classifying a text into positive, negative, or natural.
- Determine the attitude of the mass is positive, negative, or neutral towards the subject of interest.
- Design scalable, efficient and full functional sentiment analysis system models.
- Analyze emojis as unique characteristics mean with social media customers.
- Learn the correlations between words and different emojis.
- Working on five main paths sentiment processes, namely (Data Collecting-Preprocessing data- Feature extraction approaches-Classification approaches-Evaluation metrics and test performance).

1.4. Research Contribution

The research contribution of this work has suggested: Apply several feature extraction techniques to optimize the airline data sentiment analysis. The current approach comprises four steps: preprocessing data, feature extraction, sentiment classification, and experimental results evaluation. Apply five features extraction algorithms like BOW (Bag of Words), TF-IDF (Term Frequency-Inverse Document Frequency), N-Gram, and word embedding (word2vector and doc2vector). Five classification algorithms have been applied to support vector machine (SVM), naïve Bayes (NB), logistic regression (LR), random forest (RF), and XGBoost. Used two datasets with distinct data types to analyze airline datasets. Our results achieved high performance in sentiment analysis with the analysis of text and emojis. Some algorithms have high performance than others in features extraction and classification approaches. Our data preprocessing steps have achieved top results when cleaning unusual data. We compare different sentiment analysis approaches using a dataset of US Airlines from Twitter. Evaluation results stated that, in this case, Word2Vec feature extraction with (XGBoost and random forest) classifiers achieves the highest performance.

In case, Doc2Vec and the Naive Bayes classifier achieve the lowest performance. Text and emojis analysis get better accuracy results than analysis of the text only.

1.5. Thesis Organization

The thesis is organized:

- **Chapter 1:** presents the introduction, the background of the problem of developing models, the research objectives as the problem formulation, and the research contributions are given to how to solve the problems and the results.
- **Chapter 2:** introduces the background of work using the concept of the Microblogging data, the Microblogging characteristics, the sentiment analysis techniques (the background of the emotion detection, the background of the aspect-based sentiment analysis, the background of the multilingual sentiment analysis, the background of the building resources, and the background of the transfer learning), and the emojis analysis techniques.
- **Chapter 3:** presents features extraction and classification approaches. The features extraction have used a bag of words (BOW) algorithm, Term Frequency - Inverse Document Frequency (TF-IDF) algorithm, N-gram algorithm, Word Embedding (WE) algorithm. Sentiment Classification approaches as XGBoost algorithm, Logistic Regression (LR) algorithm, Random Forest (RF) algorithm, Support Vector Machine (SVM) algorithm, Naïve Bayes (NB) algorithm.
- **Chapter 4** proposes the sentiment analysis system, using Text data analysis, emojis data analysis and lexicon approach.
- **Chapter 5** presents the preliminary results and discussion. The experimental setup, details of airline datasets, text analysis results, emojis analysis results, text and emojis analysis results. Then, the guidelines for future work in the research area.
- **Chapter 6:** This thesis's conclusion provides guidelines for future work in the research area of estimating the sensor data.

Chapter 2

BACKGROUND

2.1. Microblogging Data

Social media are collective technologies that allow the creation or sharing of information, ideas, activities, and other definition forms via virtual communities and networks. Thus, its usage has increased drastically in the last decade. It helps the development of online social networks by connecting a user's profile with those of other individuals or groups. Social media systems come in various forms and support many genres of interaction. Social media's lifeblood is user-generated content, such as text posts, comments, digital photos and videos, and data generated through online interactions. Therefore, challenges to its definition arise because of the wide variety of stand-alone and built-in services currently available, and there are some standard features of these services. The factors that drive social media engagement can be divided broadly into three groups: those that are related to the post's creator (e.g., the creator's sex, age, number of followers); the post's context (e.g., time, location); and specific features of the content, such as textual content (e.g., words, tags), visual content (e.g., images, videos), and audio content.

Microblogging combines blogging and instant messaging to create brief messages posted and shared with an audience online. Social media defines a fixed platform or ecosystem for users to create a semi-public or public profile, interact with other users where they are connected, and articulate their thoughts and ideas [1]. The most prominent source of raw Microblogging material is undoubtedly Twitter [2].

Twitter (the name is an intentional misspelling of "tweeter", someone who tweets) is one of the basic social networking frequently used to give an opinion on various topics [3, 4].

Besides the short text, a tweet contains a certain amount of metadata: information about the user and its social network induced by the following relation, a fine-grain timestamp, geographical coordinates (if the user activates the option), the language used, images, videos and web links included in the text, and more. A tweet can also be simply re-tweeted, propagating a tweet to one's followers, possibly including an additional comment [5]. Other elements that make Twitter text peculiar are mentions (screen names of other Twitter users) and hash tags, short strings preceded by the symbol # to mark a topic specific to the message. Twitter is a more suitable ground for sentiment analysis than Facebook because of the limited text size of **140** characters and single opinion. In recent years, Microblogging systems such as Twitter, and Tumblr [6], have become basic communication methods for people to share their opinions, discoveries and activities with their friends. One reason users frequent social media as a platform of expression is the amount of freedom and protection of anonymity.

Often, people over-exercise their freedom of speech rights, coupled with the anonymity feature provided by the platform. This generated tons of hate speeches, racist comments, derogatory or discriminant content targeting specific groups and cyber bullying. Social media web such as Facebook and Twitter have long recognized this prevalent problem, but the development of regulations and laws to prevent such incidents is proved slow, compared to the development and growth of social media users. It is an excellent place for companies to show a more human side, speak directly with customers, share content, and offer news. Companies are also instituting internal Microblogs to keep workers connected and help foster more informal communication, particularly in organizations with large or remote workforces. Much work is still needed to make the social media platform more positive. In this sense, Twitter is considered the purest form of thoughts and idea expression social media platform, best for conducting text sentiment analysis research. Several methods are provided to retrieve large quantities of messages, based on searching for keywords, intercepting the continuous stream of messages, or accessing the users' profiles. For instance, use the

streaming API to retrieve a fraction of all the messages containing keywords from a fixed list, in real-time, in order to build a corpus of Italian tweets. At the time of this writing, Twitter counts roughly, which resembles over 350,000 tweets sent per minute, 500 million tweets per day and around 200 billion tweets per year[7].

Sentiment-related information is often encoded in lexical resources, such as effective lists and corpora. Both sentiment and emotion lexicons, and psycholinguistic resources available for English, refer to various affective models [8] and capture different nuances of affect, such as sentiment polarity, emotional categories, and emotional dimensions. Such lexicons are usually lists of words associated with a positive or negative or emotion-related label (or score). Classic and widely used lexicons for sentiment are the General Inquirer, the MPQA [9], Bing Liu's Opinion Lexicon [10], AFINN [11], and a comprehensive library of sentiment and emotion resources developed by the NRC group and available at [12]. Besides low lists of affective words, lexical taxonomies have also been enriched with sentiment and emotion information. Both SentiWordNet [13] and WordNetAffect [14] are extensions of WordNet [15]. All such resources represent a deep and varied lexical knowledge about the effect under different perspectives, and virtually all sentiment analysis systems that detect the polarity of Microblogging incorporate lexical information derived from them.

2.2. Microblogging Characteristics

Microblogging platforms provide a simple way to exchange points of view and feedback with followers. They can spread news quickly and easily, but users can also react to posts in real-time. The reciprocal relationship between users occurs through comment functions, retweets, re-blogs, or forwarding. If the content is solid and exciting, it will be spread through the community with no promotion, potentially even going viral. Microblogging data are known for its distinguishing features, which can be summarized as:

Limited length of content: the length of a Microblog is relatively short (e.g. on Twitter, it is capped at 140 characters).

An Easier Way to Share Essential or Time-Sensitive Information: designed to be straightforward to use. With a simple tweet, Instagram photo, or Tumblr post, we can update everyone on what is going on in our life (or even in the news) at this very moment.

Massive amount of data: it is essential to process this data because of the popularity of Microblogging and the valuable information.

Less Time Spent Developing Content: can post something new that takes as little as a few seconds to write or develop.

Less Time Spent Consuming Individual Pieces of Content: It is worth quickly getting the gist of the post in short to the point format without needing to read or watch something that takes too much time.

Mobile Convenience: With the growing trend toward mobile web browsing, Microblogging goes hand in hand with this newer form of web browsing.

More Direct Way to Interact with Followers: can use Microblogging platforms to encourage and facilitate more interaction by commenting, tweeting quickly, re-blogging, liking and more.

2.3. Sentiment Analysis

Sentiment analysis is detecting positive or negative sentiment in text. The future sales of any product or service depend a lot on the sentiments and perceptions of the previous buyers [16]. Therefore, it is necessary to have an efficient way to predict user sentiments about a product or service. For example, if someone has to buy tickets for a movie, then rather than manually going through all the long reviews, a sentiment classifier can predict the overall sentiment of the movie. Based on positive or negative sentiment, a decision can be taken to buy tickets. The primary sources of data are from the product reviews.

These reviews are essential to the business holders as they can take business decisions according to the analysis results of users' opinions about their products.

Determining the sentiment of posts on social media is of great value for businesses and organizations since people are increasingly using social media to express opinions [17, 18]. For example, a company can use sentiment analysis to perform market research to evaluate how their products are experienced by their users without sending out questionnaires or other ways to their customers [18]. Text classification (TC) can be used in many areas: Most consumer-based companies use sentiment classification to auto-generate reports on customer feedback. It is an integral part of CRM [19]. In medical science, text classification is used to analyze and categorize reports, clinical trials, and hospital records. Text classification models are also used in law on trial data, verdicts, and court transcripts. Text classification (TC) models can also be used for Spam email classification [20]. Also, apply to stock markets [21, 22], news articles [23], or political debates [24].

In recent years, a few research directions can be identified in sentiment analysis literature on the Microblogging web. One of them focuses on identifying additional features, including hash tags, emoticons, intensifiers such as all caps, character repetitions, and sentiment topic features [25].

Also, the standard approaches for determining sentiment on Twitter have focused on whether a tweet is positive or negative, also known as binary (or polar) sentiment analysis [26]. On the other side, multi-class sentiment analysis uses categories or clusters such as excited, happy, bored and angry to understand better the emotions expressed in the text [27, 28] and developed a Twitter Specific lexicon (or TSL for short) for Twitter sentiment analysis by using a supervised learning technique using statistical analysis. Their model was tested on **3440** manually collated tweets on Justin Bieber's Twitter account [29].

2.4. Sentiment Analysis Techniques

The idea is a transitional concept that reflects an attitude towards an entity. However, the sentiment reflects feeling or emotion while emotion reflects attitude [30]. It is essential to consider the context of the text and the user preferences. Some argued [31] about the basic prototypical emoticons; there are eight basic and prototypical emotions: joy, sadness, anger, fear, trust, disgust, surprise, and anticipation. The algorithms used for sentiment analysis could be split into two main categories: The first is rule or lexicon-based. Such methods commonly have a pre-defined list of words with a valence score. For example, nice could be +2, good +1, terrible -3, .. etc. The algorithm then matches the words from the lexicon to the words in the text and either sums or averages the scores somehow. For example, let us take the sentence, 'Today was a good day.' Each word gets a score, and to get the total valence, we sum the words. Here, we have a positive sentence. A second category is automated systems, which are based on machine learning.

The task is usually modeled as a classification problem where using some historical data with the general sentiment, we need to predict the sentiment of a new piece of text. A machine learning sentiment analysis relies on having labeled historical data, whereas lexicon-based methods rely on having created rules or dictionaries.

Lexicon-based methods fail at specific tasks because the polarity of words might change with the problem, which will not be reflected in a pre-defined dictionary. However, lexicon-based approaches can be quite fast, whereas Machine learning models might take a while to train. Machine learning models can be pretty powerful. Many people find that a hybrid approach works best in many, usually complex scenarios.

2.4.1. Emotion Detection

Sentiment analysis is concerned mainly with specifying positive or negative opinions, but emotions detection is concerned with detecting various emotions from the text. Emotions detection (ED) can be considered a sentiment analysis task.

This type of sentiment analysis aims to detect emotions like happiness, frustration, anger, and sadness. Many emotion detection systems use lexicons (i.e. lists of words and the emotions they convey) or complex machine learning algorithms. Some words that typically express anger, like evil or kill (e.g. your product is so bad or your customer support is killing me), might also express happiness (e.g. this is bad or you are killing it). Emotion detection aims to extract and analyze emotions, while the emotions could be explicit or implicit in the sentences. Emotions detection at a sentence level was proposed by [32]. They proposed a web-based text mining approach for detecting the emotion of an individual event embedded in English sentences. Their approach was based on the probability distribution of everyday mutual actions between the subject and the object of an event. The integrated web-based text mining and semantic role labeling techniques, several reference entity pairs and hand-crafted emotion generation rule to recognize an event emotion detection system. They did not use any large-scale lexical sources or knowledge base. Also, they showed that their approach revealed a satisfactory result for detecting the positive, negative and neutral emotions. Then they proved that the emotion-sensing problem is context-sensitive.

2.4.2. Aspect-based Sentiment Analysis

Aspect-based sentiment analysis is traditionally divided into three or two subtasks: rule-based [33] topic modeling [34]. For example, in this text: "The battery life of this camera is too short", an aspect-based classifier could determine that the sentence expresses a negative opinion about the feature of battery life. Another example commented, "The food was lousy - too sweet or too salty and the portions tiny". We can identify two distinct aspects: food and portions. The sentiment concerning these two aspects is negative. More fine-grained sentiment detection is usually performed in Aspect-Based Sentiment Analysis (ABSA)[33], aiming to identify the aspects of given target entities and the sentiment expressed for each aspect. Many neural models are equipped with an attention mechanism to quantify the contribution of each context word to sentiment prediction [33, 8].

However, such a mechanism suffers from one drawback: only a few frequent words with sentiment polarities are considered for final sentiment decisions, while models ignore abundant infrequent sentiment words. Future directions involving the ABSA task are evaluating cross-lingual or language-agnostic approaches and analyzing irony and figurative language.

2.4.3. Building Resources

Building Resources (BR) aims at creating lexical corpora in which opinion expressions are annotated according to their polarity and sometimes dictionaries. Building resources is not a sentiment analysis task, but it could also help improve sentiment analysis and Emotions Detection. The significant challenges that confronted the work [35] in this category are the ambiguity of words, multilingual, granularity and the differences in opinion expression among textual genres.

2.4.4. Multilingual Sentiment Analysis

The translation is transferring words or text from one language into another. Because of the complexities of language and the specific differences between various languages, translation is rarely a word-for-word transfer from English to Arabic, or vice versa. Translations must deal with contrasts between basic linguistic building blocks such as grammar, syntax, semantics, lexicons, and morphology. Add in all the various literary devices people use, such as idioms, sarcasm and slang. It is easy to see that capturing the intended meaning of a text through translation can be a daunting task. The main goal of businesses is to gain new customers and keep the ones they already have. Therefore, companies thus need to understand the feelings and opinions of their customers, even if they express them in their native language. Although great in theory, this is difficult for most marketers since the manual analysis of each comment and review is time-consuming and a costly task in the long run. It is not a sustainable model.

That is why it is vital to have a native, multilingual approach to applying sentiment analysis to identify and analyze customer emotions across social media, surveys, and customer service tickets. Thankfully, companies need to ensure that the model they choose offers the same high accuracy score while analyzing all the languages they intend to use. Multilingual sentiment analysis can be complex. It involves a lot of preprocessing and resources. Most of these resources are available online (e.g. sentiment lexicons). Alternatively, in paper [36], they detected language in texts automatically with MonkeyLearn's language classifier and then trained a custom sentiment analysis model to classify texts in the language of your choice. However, with businesses trying to know their consumers, we want to understand the actual intended meaning of their feelings regarding prices, product features, customer service, and quality. When it comes to analyzing the Voice of the Customer, accuracy is everything. Multilingual sentiment analysis allows the extraction of brand insights from customer feedback in the native language without translation. It is indeed one of the essential features of sentiment analysis tools.

2.5. Emojis Analysis



Emojis are picture characters originating in Japan in the late 1990. Emojis have emerged as a cultural form of typographic habits, corporate strategies, copyright claims, online chat rooms, and technical standards disputes. More often than not, semantic classification tasks treat Emojis as noise and remove them from the dataset in the preprocessing stage [37]. In contrast to emoticons of text analysis, Emojis are actual pictures and can convey a broader range of emotions because of their nature. A much more comprehensive range of concepts and ideas can be visualized, such as weather, food and events. For example, the rugby ball Emojis (🏉) can be inserted to show that the text is about rugby. However, because of the increasing number of Emojis, Emojis are rich in social, cultural, and economic significance. Thus, the level of ambiguity created by cultural differences and cross-device implementations has become a problem in sentiment analysis [38].

However, because of their ubiquity and variety, Emojis contains semantic information. Emojis, considered for sentiment analysis and sarcasm detection, demonstrate that using the semantic information they carry is beneficial [39, 40]. The sentiment is usually considered to have three polarities, i.e., positive, negative, and neutral [41]. Many approaches, such as surveys [42], biometric measurements [43], and text analysis [44], have been developed to detect the sentiment or emotion of users. Emojis have been used as a type of distant supervision using predefined emotion classes based on psychological models [45], binary (positive/negative) classes [3], or a set of single Emojis [46]. However, such predefined Emojis classes rarely account for the culturally diverse use of Emojis [28]. Research [28] on the interpretation and prediction of Emojis has developed in a similar spirit to other research in an NLP, with similar representation learning-based methods. In recent years, much research worked to choose and find the Emojis sentiment about tweets. Few types of research extract sentiment found by utilizing Emojis. They work to design many text mining methods to analyze Emojis. Researchers usually incorporate feature extraction methodologies with machine learning techniques in building Emoticons analysis frameworks to achieve effective classification performance. The significant benefits of Emojis are operating in social responses to customers and using them as keyboard characters to represent facial expressions and to send the writer's character. There is an example of positive feeling using Emojis ("The car dealership agent conveys all the positive feeling words like thanks"), uses the thumbs up (👍) and smiley Emojis (😊) to intensify the feeling, most of the users use at least a smiley or two. Therefore, users use Emojis in social, educational, medical, and other topics on the Microblogging web. Many users show a shocking lack of class or civility on social media of Microblogging data. Social media makes them feel anonymous, and they forget their words are public. Sometimes Emojis are considered a simple tradition of facial expression [47]. In paper [48], both modes of data were analyzed by combined and separately with both machine learning and deep learning algorithms for finding sentiments from Twitter-based

airline data using several features such as TF-IDF, Bag of words, N-gram, and emoticon lexicons. T

he research found that DL algorithms work better than ML. On the other side, [49] conducted sentiment analysis by taking Emojis illustrations and using emoticons sentiment vector. They put the Emojis illustrations into an alternative model neural network on a Chinese Microblog dataset. **Table 2-1** presents an example of the same sentence meaning different sentiments when using different Emojis.

Table 2-1 Same Sentences with different Emojis

Sentence	Emojis	Description	Sentiment
"Ok, I will go there tomorrow."		Happy	Positive
"Ok, I will go there tomorrow."		Boring	Negative

Chapter 3

FEATURES EXTRACTION AND CLASSIFICATION APPROACHES

3.1. Features Extraction (FE)

There are often too many factors (features) in real-world machine learning problems based on the final prediction. The higher the number of features, the harder it gets to visualize the training set and then work on it. Where the dimensionality reduction algorithms come into play. Feature extraction is a process of dimensionality reduction by which we reduce an initial set of raw data to more manageable groups for processing. It is a process where we auto-extract those features in our data that contribute most to the prediction variable or output we are interested in. In the feature extraction step, we parse and extract different feature types, i.e. words and word combinations, then Stemming. Stemming and lemmatization are text normalization (or sometimes called Word Normalization) techniques in natural language processing to prepare text, words, and documents for further processing [22]. Having irrelevant features in our data can decrease the accuracy of many models, especially linear algorithms like linear and logistic regression. In feature extraction, we first define the feature type (e.g., words or word combinations) that best reflects the tweet's content and second parse all tweets to extract their features.

Before modeling data, there are four benefits to performing feature extraction:

Reduce the over-fitting: Less redundant data means less opportunity to decide based on noise from the data set.

Improve Accuracy: Less misleading data means modeling accuracy improves.

Reduce the training time: fewer data means algorithms train faster. Reduce resources: reduce the number of resources losing no critical information.

The ability of the machine learning process highly depends on its features, so it is essential to choose the purpose of feature extraction [50, 51]. With the development of social networking, the amount of data on the web is growing exponentially. There are over **115** million active Twitter users every month, and they publish millions of tweets every day and the valuable information in this data. The challenges to managing these data have attracted many researchers' interest in designing systems for Microblogging. Many feature extraction methods exist for this data and are widely used. All these methods aim to remove redundant and irrelevant features so that classification of new instances will be more accurate. Therefore, their advantages and disadvantages are outlined to provide a clearer idea of when to use each of them to save computational time and resources. The data features used to train the machine learning models greatly influence their performance. Irrelevant or partially relevant features can negatively affect model performance. We will discover automatic feature extraction techniques that can prepare our system. We will review some of the most popular machine learning methods. After completing this chapter, we will know feature extraction, principles component analysis, feature importance, algorithms like a bag of words (BOW), Term Frequency -Inverse Document Frequency (TF-IDF), N-Gram, and Word embedding. Classification approaches like XGBoost, Logistic regression, Random Forest, Support vector machine, and naïve Bayes. Last, we compare algorithms with their limitations and their characteristics.

3.1.1 Bag of Words (BOWs)

The first step in performing a sentiment analysis task: transforming text data to numeric form. A machine learning model cannot work with the text data directly, but we create from the data with numeric features. Starts with a primary and crude but often quite helpful method, called bag-of-words (BOW). A bag-of-words approach describes the occurrence or frequency of words within a document or a collection of documents (called a corpus). It comes down to building a vocabulary of all the words occurring in the document and keeping track of their frequencies.

A Bag-Of-Words (BOW) model, or BOW for short, is a way of extracting features from the text for modeling, such as with machine learning algorithms. BOW is the most used technique for natural language processing. It is a technique for NLP that extracts the words (features) used in a sentence, document, website, etc. and classifies them by frequency of use. This technique in this study applies to text processing [50]. In unsupervised learning, one tries to uncover hidden regularities (clusters). In the Microblogging filtering task, some features could be the bag of words or the subject line analysis. The input of this technique is some tweets, and the output is the sentiment scores for each word. Each tweet is a bag of words. Assume that the order of words has no significance (the word “homemade” has the same probability as “made home”). Bag of words representation (or vector space representation) [52] is the primary method of information retrieval researchers to represent text corpus, which is a straightforward approach to convert unstructured text to structured data based on a word by word and neglects the grammar. A bag of words (BOW) represents a text that describes the occurrence of words within a document. It involves two things: a vocabulary of known words and a measure of the presence of known words. This algorithm has some goals, and it cannot only evaluate sentiment scores. **Figure 3-1** shows the flowchart of the bag of words algorithm.

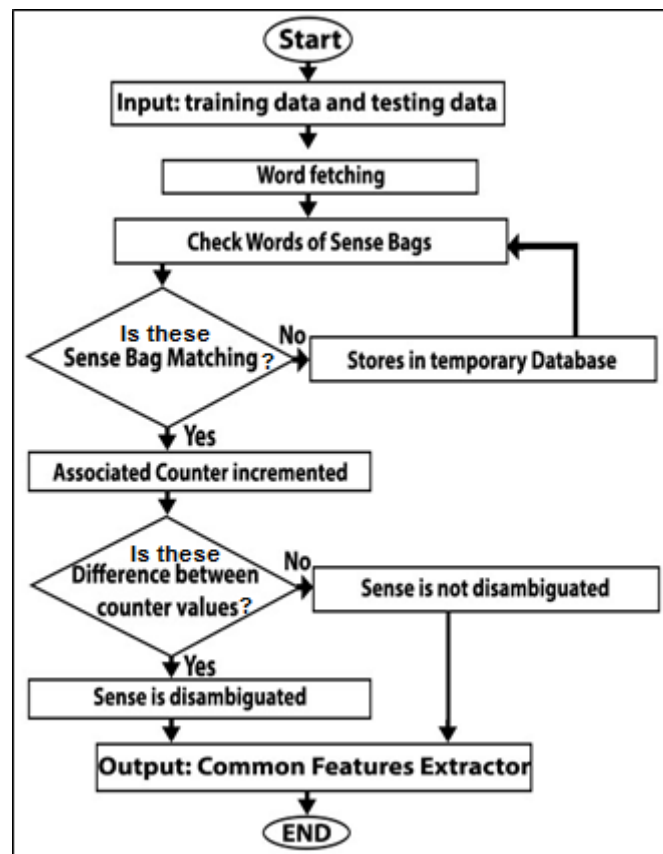


Figure 3-1 Flowchart BOWs

Flowchart BOWs with the input training and testing data and the output standard feature extraction. After fetching words from the dataset, we check words of sense bags, check the difference between counter values and extract the dataset's features. The Bag-Of-Words model is the most widely used technique for sentiment analysis; it has two significant weaknesses: using a manual evaluation for a lexicon in determining the evaluation of words and analyzing sentiments with low accuracy because of neglecting the language grammar affects the words and ignores semantics of the words. So, feature extraction is one of the essential parts of this complete process. The approach is straightforward and flexible and can extract features from documents.

3.1.2. Term Frequency - Inverse Document Frequency (TF-IDF)

Term frequency–Inverse document frequency model or TF-IDF for short is a numerical statistic intended to reflect how important a word is to a document in a corpus. That calculates the inverse likelihood of finding a word in a document. Term frequency (TF) represents a repeated number of times a term or word in a document, and inverse document frequency (IDF) represents the inverse frequency of a document [53]. The methods for word frequency (TF) and inverse tweet frequency (IDF) are as follows [54] as in equation (1):

$$tf_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}} \quad (1)$$

Here, in the numerator, n is the number of times the term “ i ” appears in the document “ j ”. Thus, each document and term would have its TF value. Where n_{ij} is the number of appearances of the word t_i in doc d_j , and $\sum_k n_{kj}$ is the sum of the appearances of all words in the doc d_j [54, 55] as in equation(2).

$$Idf_i = \log \frac{|D|}{|\{j: t_i \in d_j\}|} \quad (2)$$

IDF is a measure of how important a term is, and computing just the **TF** alone does not understand the importance of words: Where, $|D|$ represents the number of the dataset in the framework, $|\{j: t_i \in d_j\}|$ is the tweet number containing the word t_p . If the word is not in the framework, that will cause the dividend to be zero. So use, $1 + |\{j: t_i \in d_j\}|$. TF-IDF approach is to rescale the frequency of words by how often they appear in all documents so that the scores for frequent words like "the" that are also frequent across all documents are penalized.

Represent the word's score in any document as per the following equation (3), compute the TF-IDF score for each word in the corpus. Words with a higher score are more important, and those with a lower score are less important as in equation(3):

$$TFIDF(word, doc) = TF(word, doc) * IDF(word) \quad (3)$$

TF-IDF algorithm calculates the term frequency and orders the word by term frequency (TF) from the training and testing data. Then, calculate the n-words weight in each tweet, build vectors for tweets, and extract the dataset's features.

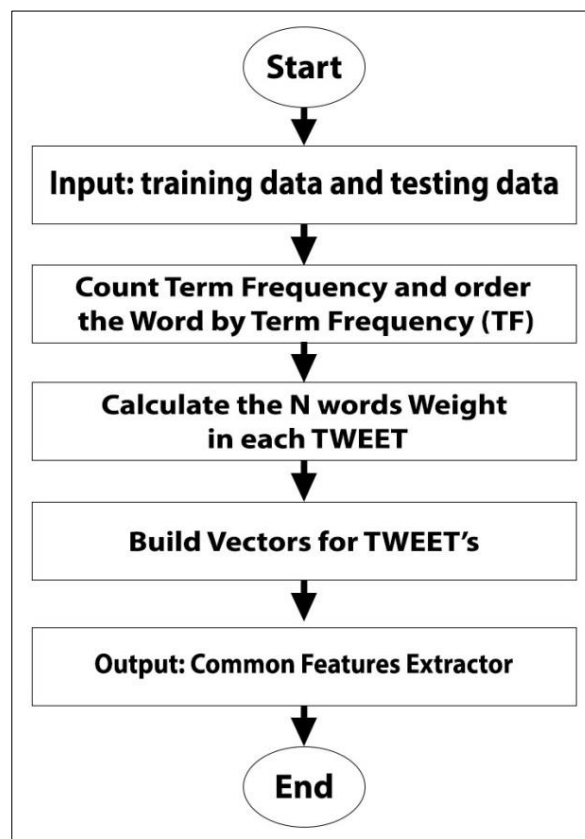


Figure 3-2 Flowchart TF-IDF

Figure 3-2 shows the TF-IDF flowchart, starting with the input training and testing data and ending with the standard features. It counts term frequency, orders the word by term frequency, calculates the N-words weight in each Tweet, and builds tweets vectors.

3.1.3. N-Gram

Imagine you have a sentence such as 'I am happy, not sad' and another one 'I am sad, not happy'. They will have the same representation as a BOW, even though the meanings are inverted. Here, putting NOT in front of the word (also called negation) changes the whole meaning and shows why context is important. There is a way to capture the context when using a BOW by, for example, considering pairs or triples of tokens that appear next to each other. Let us define some terms. Single tokens used so far and are also called 'unigrams'. A bigram is pair of tokens, trigrams are triples of tokens, and a sequence of n-tokens is called 'n-grams.'

It is easy to implement n-grams with the CountVectorizer method. To specify the n-grams, we use the N-gram range parameter. The N-gram range is a tuple where the first parameter is the minimum length and the second parameter is the maximum length of tokens. For instance, N-gram range = (1, 1) means use only unigrams (1, 2), use unigrams and bigrams, etc. It is difficult to determine the optimal sequence for the problem. If we use a more extended token sequence, this will cause more features. In principle, the number of bigrams could be the number of unigrams squared; trigrams, the number of unigrams to the power of 3 and so forth. Having longer sequences results in more precise machine learning models, increasing the risk of over-fitting. An approach to finding the optimal sequence length would be to try different lengths in a grid search and see which results in the best model. It is difficult to determine what the optimal sequence for the problem is. When using a more extended token sequence, this will cause more features. In principle, the number of bigrams could be the number of unigrams squared; trigrams, the number of unigrams to the power of 3 and so forth. Having longer sequences results in more precise machine learning models, increasing the risk of over-fitting. An approach to finding the optimal sequence length would be to try different lengths in a grid search and see which results in the best model. N-gram is a contiguous sequence of (n) items from a text sample. It is a sequence of N words, letters, or syllables [56]. These items include letters, words, and phonetics upon application [57].

N-grams of texts are extensively used in text mining and natural language processing tasks. They are a set of co-occurring words within a given window, and when computing the n-grams, you typically move one word forward. For Microblogging N-grams, splitting a tweet into a substring of fixed length. It is also called, 'unigram', 'bigram', and, 'trigram'. A unigram is a single word, a bigram is a phrase made of two single words, and a trigram is a phrase made of three single words. A tweet document can be transformed into many kinds of unique features. For example, the sentence "How are you" has three unigrams, "How", "are" and "you", two bigrams "How are", "are you", and one trigram, "How are you" [58]. The size of N is 1, 2, and 3. 3-Gram was slightly weaker than 2-Gram since 3-Gram suffers from a high number of combinations, causing a rapid decrease in actual frequencies per feature. Thus, 2-Gram was used in our experiment. This study uses a bigram with a fixed length (1, 2). If $X = \text{Num of words in a sentence } K$, the number of n-grams for sentence K would be [58] as in equation (4):

$$N_{gramsK} = X - (N - 1) \quad (4)$$

N-grams are used for developing features for supervised Machine Learning models such as SVMs, Random forests, and Naive Bayes. The idea is to use tokens such as bigrams in the feature space instead of just unigrams.

3.1.4. Word Embedding (WE)

Word embedding model, or (WE) for short, is a word representation that allows words with similar meanings to have an equal representation. Natural language processing (NLP) used word embedding to represent words for text analysis. Getting word embedding using a set of language modeling and feature learning techniques where words from the vocabulary are mapped to vectors of real numbers. It involves the mathematical embedding from space with many dimensions per word to a continuous vector space with a much lower dimension. Map each word one vector and learn the vector values in a resemble neural network with a similar meaning word to have an equal representation [59].

Word embedding methods learn a real-valued vector representation for a pre-defined fixed-sized vocabulary from a text corpus. The learning process is joint with the neural network model on some task, such as document classification, or is an unsupervised process using document statistics. This section reviews two techniques for learning a word embedding feature extraction for text (Word2Vec) and (Doc2Vec).

3.1.4.1. Doc 2 Vector (D2V)

For short, Doc 2 vector models (D2V) are an unsupervised learning approach to generate vectors for sentences, paragraphs, or documents. The input (tweets) is varied, while the output is fixed-length vectors. First, pass the training data to build vocabulary and request the training phase to compute word vectors. Then, encode it by providing training testing data, and pass vectors to a sentiment analysis classifier.

3.1.4.2. Word 2 Vector (W2V)

Word 2 vector model, or W2V for short, is a shallow text representation model with the fundamental principle of learning low dimensional vectors from bagging [57, 54]. This work bases the Word2vec model on a word appearing around the word in the tweet. It gives an individual weight to each word. Features of the word2vec model represent a vector representing a specific word with a particular list of numbers. Use the average weight of the Word2vec model to target learning word relationships of the tweet by using a neural network model from a large corpus. Word2vec can discover similar words for a partial sentence or suggest other words. Chose the vectors in mathematical function to show the semantic similarity level between the words described by those vectors [57], as in **Figure 3-3**.

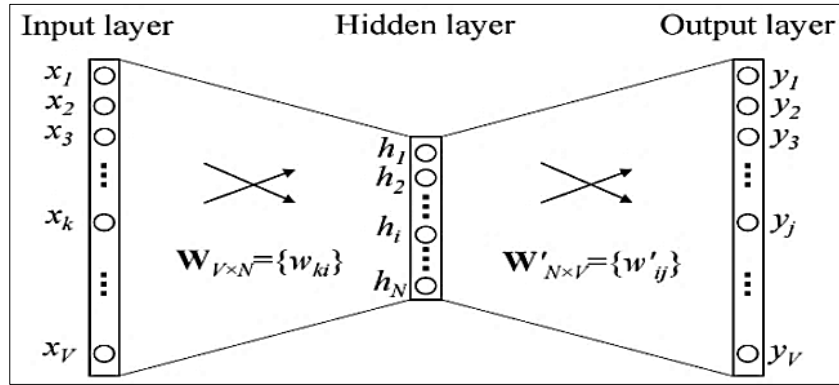


Figure 3-3 the model of Word2Vec [60]

Adapted the model from a neural network with a single hidden layer, and the embedding is the weight of the hidden layer in the neural network, as shown in **Figure3-3** [60]. Word2Vec gives a unique weight to each word based on the words appearing around the word. The features extracted from the Word2Vec algorithm comprise vector space, typically of several hundred dimensions, with each unique word in the corpus of text being assigned a corresponding vector in the space. Word vectors in Word2Vec are positioned on the vector space such that the words that share everyday contexts in the corpus are in proximity to one another in the space [57].

The model maps the document to as a feature vector [54] as in equation(5):

$$V(d) = (t_1, W_1(d); \dots; t_n, W_n(d)) \quad (5)$$

In this formula, t_i ($i=1, 2, \dots, n$) is a list of mutually exclusive terms, $W_i(d)$, which define as the frequency $tf_i(d)$ where t_i appears in d . In contrast to noun phrases, this feature type is not limited to certain parts of speech. However, it may also contain verbs and adverbs as long as the feature selection attests to high explanatory power, as presented in **Table 3 -1**, the difference between features extraction algorithms with characteristics and limitations.

Table 3-1 Feature Methods (Characteristic and Limitations)

Feature Methods	Characteristic	Limitations
Bag of Words	Very simple to understand, implement, and offer much flexibility for customization on your specific text data. Great success in prediction problems like language modeling and document classification.	Vocabulary: requires careful design in order to manage the size. Sparsely: Sparse representations are harder to model for computational reasons (space and time complexity) and information reasons, where the challenge is for the models to harness so little information in such a sizeable representational space. Meaning: Discarding word order ignores the context and meaning of words in the document (semantics).
TF-IDF	Easy to compute. Easy to Compute the similarity between 2 documents using it. Primary metric to extract the most descriptive terms in a document. Common words do not affect the results due to IDF.	It does not capture a position in the text (syntactic). It does not capture meaning in the text (Semantics).
N-Gram	Many NLP applications benefit from N-gram models, including part-of-speech tagging, natural language generation, word similarity, sentiment extraction and predictive text input.	N-gram models poorly capture longer-distance context. It has been shown that after 6-grams, performance gains are limited. Need to ensure that the training corpus looks similar to the test corpus.
Word Embedding	The vector space model is the most widely used text representation model.	It is prone to disasters of dimensions. It cannot describe the relationship between words very well.

3.2. Sentiment Classification Approaches

The Sentiment classification classifies a target unit in a document to a positive (favourable) or negative (unfavourable) class. There are two main approaches in sentiment analysis, i.e. Supervised learning and Unsupervised Learning Approach. An unsupervised learning approach is used to classify the sentiment when having training data, and it can solve the problem of domain dependency and the need to reduce the training data. There are three primary classification levels [61]:

Document-level: classifies an opinion document as expressing a positive or negative opinion or sentiment.

Sentence-level: classifies sentiment expressed in each sentence. If the sentence is subjective, it classifies it in positive or negative opinions.

Aspect-level: classifies the sentiment regarding the specific aspects of entities. Users can give different opinions about different aspects of the same entity.

We describe three approaches for performing sentiment extraction:

The subjective lexicon approach is a list of words assigned a score that shows its nature in terms of positive, negative or aim. Base lexicon-based methods on the insight that we can get the polarity of a piece of text on the ground of the polarity of the words which compose it.

N-gram modelling approach: uni-gram, bi-gram, tri-gram, or combination for the sentiments classification.

Machine learning approach: performs the semi and supervised learning by extracting the features from the text and learning the model.

Training and testing data that users usually are product reviews. Sentiment classification is essentially a text classification problem. Traditional text classification mainly classifies different topics, e.g., politics, sciences, and sports. In such classifications, topic-related words are the key features. However, sentiment or opinion words that show positive or negative opinions are more critical in sentiment classification, e.g., excellent, unique, horrible, worst. We will present classification algorithms like XGBoost, Logistic regression, Random forest, support vector machine, and naïve bases, flowcharts and functions. **Table 3-2** shows the difference between Automated/Machine learning and Rule/lexicon-based.

Table 3-2 Difference between Automated and Rule-Based

Automated/ Machine learning	Limitations
Rely on having labeled historical data.	Rely on manually crafted valence scores.
It might take a while to train.	Different words might have different polarities in different contexts.
The latest machine learning model can be powerful.	It can be quite fast.

3.2.1. XGBoost

XGBoost is the machine learning algorithm under the gradient boosting framework. It is an optimized distributed gradient boosting library designed to be highly efficient, flexible and portable. XGBoost provides a parallel tree boosting (also known as GBDT, GBM) that solves many data science problems quickly and accurately. Data scientists use the XGBoost to provide parallel tree boosting and achieve state-of-the-art results on many machine learning challenges. This algorithm has many different names, such as gradient boosting, multiple additive regression trees, stochastic gradient boosting or gradient boosting machines. Boosting is an ensemble technique where we added new models to correct the errors made by existing models. We add sequential models until we can make no further improvements. A famous example is an AdaBoost algorithm [62] that weights data points hard to predict. Gradient boosting is an approach where new models are created that predict the residuals or errors of prior models and then added together to make the final prediction. It is called gradient boosting because it uses a gradient descent algorithm to minimize the loss when adding new models. This approach supports both regression and classification predictive modeling problems. XGBoost algorithm is used to classify the airline dataset [63, 58] using five feature extractions. **Figure 3-4** shows the flowchart of the XGBoost classification algorithm.

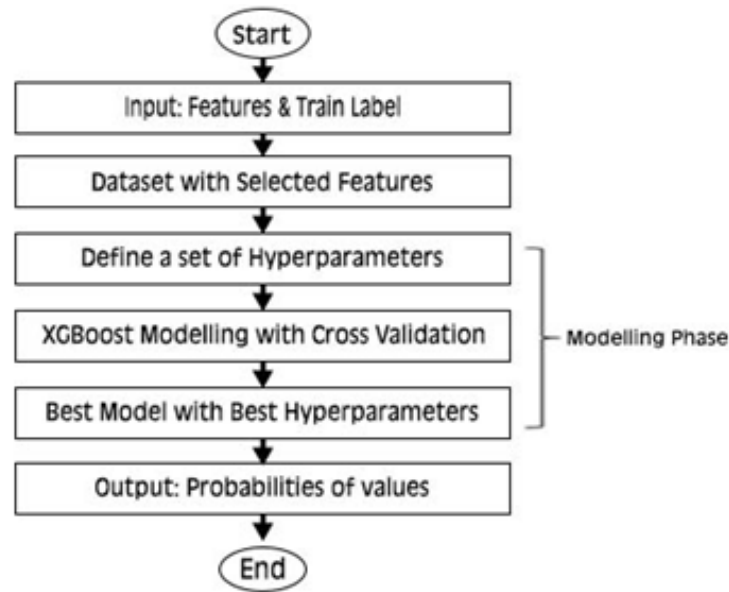


Figure 3-4 Flowchart XGBoost

We read extracted features and trained label data, then defined a set of hyperparameters and used the XGBoost approach to cross-validation. The output of this step is the probability values of testing labels. The two reasons to use XGBoost are also the two goals of our system [62]: **Execution Speed**: XGBoost is fast. Fast when compared to other implementations of gradient boosting. **Model Performance**: XGBoost dominates structured or tabular datasets on classification and regression predictive modeling problems. The evidence is that it is the go-to algorithm for competition winners on the Kaggle competitive data science platform [64].

3.2.2. Logistic Regression (LR)

A logistic regression model, or LR for short, is a classification model in which the response variable is categorical. It is an algorithm that comes from statistics and is used for supervised classification problems. Logistic regression is a proper analytic technique. Regression is a statistical process for evaluating the relationships among variables; the output often predicts any outcome (test labels). Logistic regression is also called maximum entropy.

It assumes a Gaussian distribution for the numeric input variables and models binary classification problems. **Figure 3-5** shows a flowchart of a logistic regression classification algorithm.

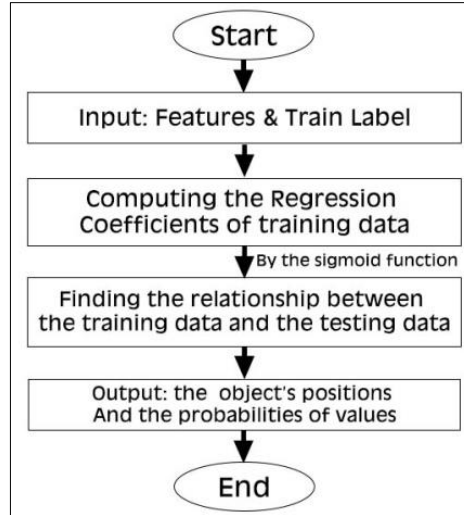


Figure 3-5 Flowchart Logistic Regression

Binary logistic regression is a type of regression analysis where the dependent variable is a dummy variable: coded 0 (negative) or 1 (positive). The most common logistic regression model is a binary outcome that can take two values, such as true/false, yes/no, and so on. The vertical axis stands for the probability of classification, and the horizontal axis is the value of \mathbf{x} . It assumes that the distribution of $y | \mathbf{x}$. The formula of LR is as follows [65] as in equation(6):

$$F(\mathbf{x}) = \frac{1}{1 + e^{-(B_0 + B_1 \mathbf{x})}} \quad (6)$$

Here $B_0 + B_1 \mathbf{x}$ is similar to the linear model $y = ax + b$. The logistic function applies a sigmoid function to restrict the value from a large scale to within the range 0 -1.

3.2.3. Random Forest (RF)

A random forest is a supervised machine learning algorithm constructed from decision tree algorithms. This algorithm is applied in various industries, such as banking and e-commerce, to predict behavior and outcomes. It is used to solve regression and classification problems.

It uses ensemble learning, a technique that combines many classifiers, to solve complex problems. The "forest" it builds is an ensemble of decision trees, usually trained with the "bagging" method. The general idea of the bagging method is that a combination of learning models increases the overall result. RF is used to solve regression and classification problems. Random forest uses ensemble learning, a technique that combines many classifiers, to solve complex problems. It comprises many decision trees. This algorithm runs efficiently on large data sets and uses features randomness and bagging when building each tree. It tries to create a forest of trees to predict more accurately than any individual tree has below: for a set of tweets t_1, t_2, \dots, t_n , and their sentiment labels s_1, s_2, \dots, s_n . Gets selects a random sample (Tb, Sb) with replacement. The output of this step will be the object's positions and the probabilities of values of test labels. All classification trees $f(b)$ were trained using a random sample (Tb, Sb) where b ranges from $1, \dots, n$. Finally, we are taking a majority vote of predictions of these B-trees [58].

Classification in random forests employs an ensemble method to attain the outcome. They feed the training data to train various decision trees. This dataset comprises observations and features selected randomly during the splitting of nodes. A rain forest system relies on various decision trees. Every decision tree comprises decision nodes, leaf nodes, and a root node. The leaf node of each tree is the final output produced by that specific decision tree. The selection of the final output follows the majority-voting system. Here, the output chosen by most decision trees becomes the final output of the rain forest system. Decision trees are the building blocks of a random forest algorithm. A decision tree is a decision support technique that forms a tree-like structure. An overview of decision trees will help us understand how random forest algorithms work. Combining hundreds of decision tree models, the random forest reduces the variance and bias, which is hard to achieve because of the bias-variance threshold. The significant difference between the decision tree algorithm and the random forest algorithm is establishing root nodes, and we did segregate nodes randomly in the latter.

The random forest employs the bagging method to generate the required prediction. Random forest trains decision tree classifiers (in parallel) on various sub-samples of the dataset (also referred to as bootstrapping) and various sub-samples of the features. Random forest is an ensemble classifier based on bootstrap followed by aggregation (jointly referred to as bagging) [67]. Samples of the training dataset are taken with replacement, but we construct the trees to reduce the correlation between individual classifiers. Specifically, rather than greedily choosing the best split point in the construction of each tree, only a random subset of features is considered for each split. Random Forest enhances classification accuracy and is very good for large datasets [66]. The eventual results were that the RF algorithm established the outcome based on the predictions of the decision trees. It predicts by taking the average or mean of the output from various trees. Increasing the number of trees increases the precision of the outcome. A random forest eradicates the limitations of a decision tree algorithm. It reduces the over-fitting of datasets and increases precision.

3.2.4. Support Vector Machine (SVM)

Support vector Machine or SVM or short (SVM) is a supervised machine learning algorithm. We based SVM on the concept of decision planes that define decision boundaries. We can use it for both classification and regression challenges [20]. In the SVM algorithm, we plot each data item as a point in n-dimensional space (where n is several features we have), with the value of each feature being the value of a particular coordinate. Then, we perform classification by finding the hyper-plane that differentiates the two classes very well, as in **Figure 3-6**.

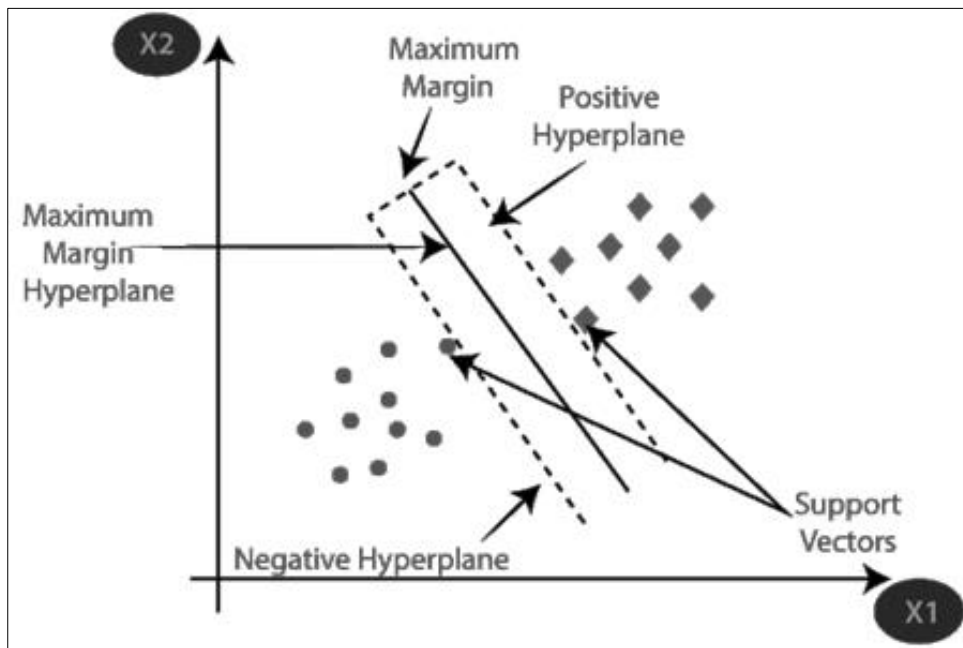


Figure 3-6 SVM classes

The SVM modeling algorithm finds an optimal hyperplane with the maximal margin to separate two classes [20]. Support Vector Machines (or SVM) seek a line that best separates two classes. It classifies new samples or vectors by specifying where on the hyperplane based on the lowest risk principle structural as **Figure3.6**. The basics of Support Vector Machines and how it works are best understood with a simple example. Let us imagine we have two tags: green and blue, and our data has two features: x and y . We want a classifier that, given a pair of (x,y) coordinates, outputs if it is blue (square shape) or green (circle shape). We plot our already labeled training airline data. This algorithm creates a hyperplane separating the positive and negative samples during training. The kernel transforms data not linearly separable in dimensional space to a higher dimension, where it is linearly separable. **Figure 3.7** shows the flowchart of the SVM classification algorithm. SVM chooses the extreme points/vectors that help in creating the hyperplane. These extreme cases are called support vectors, and hence algorithm is termed a Support Vector Machine.

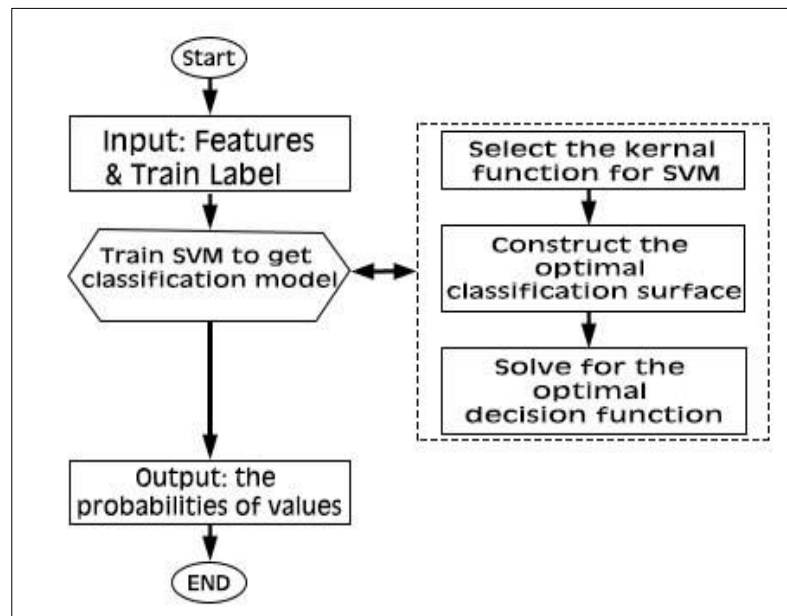


Figure 3-7 Flowchart SVM

Compared to newer algorithms like neural networks, they have two significant advantages: higher speed and better performance with a few samples (in the thousands). This makes the algorithm very suitable for text classification problems, where it is common to have access to a dataset of at most a couple of thousands of tagged samples.

3.2.5. Naïve Bayes (NB)

Naive Bayes' model, or NB for short, is a probabilistic technique for constructing classifiers. Naive Bayes' classifiers are a collection of classification algorithms based on Bayes' theorem. It is not a single algorithm but a family of algorithms where all of them share a common principle, i.e. every pair of classified features is independent of each other. The characteristic assumption of the naïve Bayes classifier is that the value of a particular feature is independent of the value of any other feature, given the class variable. For example, we may consider fruit an apple if it is red, round, and about 3 inches in diameter. Even if these features depend on each other or upon the existence of the other features, these properties independently contribute to the probability that this fruit is an apple, which is why it is known as 'Naive'. This algorithm deals with datasets like tweets.

We have a training data set of airlines and corresponding target variable 'sentiment analysis' (suggesting possibilities of sentimentality). Now, we need to classify whether "Airline companies will be positive or negative score based on tweets of Twitter. Naive Bayes classifiers primarily used in text classification have a higher success rate than other algorithms. As a result, they widely used it in sentiment analysis (in social media analysis, to identify positive and negative customer sentiments). Thus, we could be used it for making predictions in real-time. Along with simplicity, Naive Bayes outperforms even highly sophisticated classification methods. This algorithm can exceed the most potent alternatives with small sample sizes because it is relatively robust, easy to implement, fast, and accurate. The occurrence of one feature in Naïve Bayes does not affect the probability of occurrence of the other feature. Despite the oversimplified assumptions mentioned previously, naïve Bayes classifiers have excellent results in complex real-world situations. An advantage of naïve Bayes is that it only requires a small amount of training data to estimate the parameters necessary for classification and that the classifier can be trained incrementally.

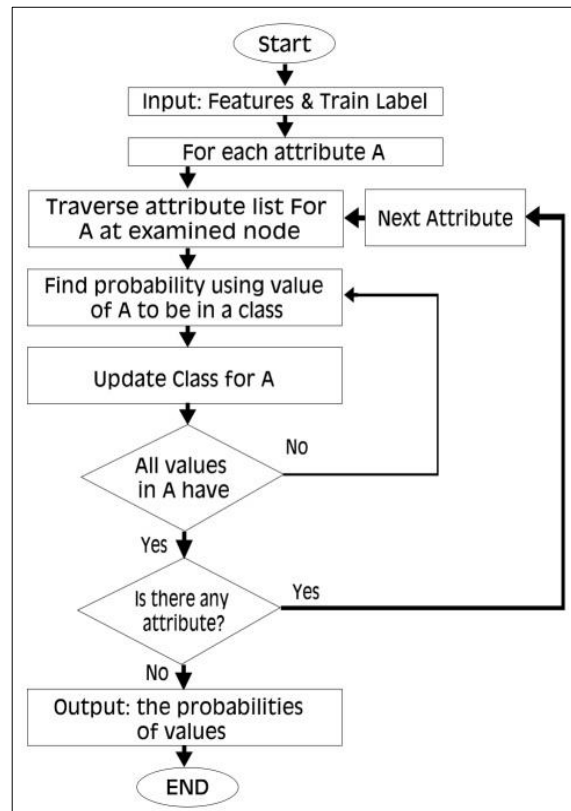


Figure 3-8 Flowchart Naïve Bayes

Figure 3-8 shows a flowchart of the Naive Bayes classification algorithm. Naive Bayes' model is easy to build and valuable for extensive data sets. We derive the Naive Bayes (NB) classifier by first observing that by Bayes' rule [68] as in equation(7):

$$P(c|d) = \frac{P(c)P(d|c)}{P(d)} \quad (7)$$

Where $P(d)$ plays no role in selecting c . To estimate the term $P(d|c)$, Naive Bayes decomposes it by assuming the f_i 's are conditionally independent given d 's class [68] as in equation(8):

$$PNB(c|d) = \frac{P(c) \left(\prod_{i=1}^m P(f_i | c)^{n_i(d)} \right)}{P(d)} \quad (8)$$

Our training method consists of relative-frequency estimation of $P(c)$ and $P(f_i / c)$, using add-one smoothing. Naive Bayes calculates the probability of each class and the conditional probability of each class given each input value. We estimated these probabilities for new data and multiplied them together, assuming that they are all independent (a naïve or straightforward assumption). When working with real-valued data, we assumed a Gaussian distribution to efficiently estimate the probabilities for input variables using the Gaussian Probability Density Function. Gaussian Naive Bayes assume continuous values associated assumes features according to a Gaussian distribution. A Gaussian distribution is also called Normal distribution. When plotted, it gives a bell-shaped curve that is symmetric about the mean of the feature values. The theorem of Naïve Bayes uses the features in the dataset that are mutually independent [4]. Finally, in this section, we compared our classification algorithms approaches in **Table 3-3**. Support vector machines (SVMs) have been highly effective at traditional text categorization outperforming Naive Bayes. They are large-margin, rather than probabilistic, classifiers, in contrast to Naive Bayes and MaxEnt [68].

Recommendation results after Table 3-3 that made a comparison between classifications algorithms with features and limitation, we can suggest that Microblogs data as tweets have achieve high performance when using XGboost, and logistic regression with word 2 vector features. Also after using random forest in the data (Text and emojis) analysis have a high performance. Naïve Bayes achieve a lower

performance in text analysis with Microblog data. Meanwhile using naïve Bayes have a good performance in model of emojis and text. Every datasets have its problems, and the target is how to solve this problem's my building a machine learning models. So every model depending on the natural of dataset, the question is what the problem is, and how to solve it?

Table 3-3 Comparison between classification methods

Feature Methods	Characteristic	Limitations
Naïve Bayes	Based on simple probability principles. They can use it in this type of classification. Simple to implement. Excellent Computational efficiency and classification rate. It predicts accurate results for most of the classification and prediction problems.	It ignores interdependencies that may exist between attributes. Therefore, it is called "Naïve". Computations have to be repeated every time there is a record whose class needs to be discovered. The precision of the algorithm decreases if the amount of data is less. To get excellent results, it requires a large number of records.
LR (Logistic Regression)	Simple to implement. Does not require too many computational resources. It does not require input features to be scaled (pre-processing). It does not require any tuning. A simple algorithm that models a linear relationship between inputs and a continuous numerical output variable. Use cases: Stock price prediction, Predicting housing process, predicting customer lifetime value. Explainable method, interpretable results by its output coefficients, faster to train than other machine learning models.	It cannot solve non-linear problems. Prediction requires that each data point be independent. Attempting to predict outcomes based on a set of independent variables. Assumes linearity between inputs and outputs. Sensitive to outliers. Can underfit with small, high dimensional data.
XGBoost	The model outcomes are weighed based on the previous stage XGBoost can parallel computation ten times faster than Random forest Based on tree model, fast speed, handle, and large-scale dataset. Gradient Boosting algorithm that is efficient & flexible. Can be used for both classification and regression tasks. Use cases: Churn prediction, Claims processing in insurance. Provides accurate results. Captures non linear relationships.	Manually create dummy variable/ label encoding for categorical features before feeding them into the models. Hyper parameter tuning can be complex. Does not perform well on sparse datasets
SVM	High accuracy. Work well even if data is not linearly separable in the base feature space. SVM can model non-linear decision boundaries. Performs similarly to logistic regression when linear separation. Robust against over-fitting problems (especially for text data set because of high-dimensional space). Excellent results for numerical data. SVM has better generalization properties, and it is insensitive to curse to dimensionality.	Speed and size requirements both in training and testing are more. High complexity and extensive memory requirements for classification most times. Lack of transparency results caused by many dimensions (especially for text data). Choosing an efficient kernel function is difficult (susceptible to over-fitting/training issues depending on the kernel). Memory complexity. Expensive; it takes a long time for each run. It is not suitable for handling the dynamic nature of the signal.

Random Forest	Reduced risk of over fitting. Provides flexibility. Easy to determine feature importance. An ensemble learning method that combines the output of multiple decision trees. The use cases are: 1- credit score modeling. 2- Predicting housing prices.	Time-consuming process: slow to process data as they are computing data for each decision tree. Requires more resources: more resources to store more extensive data sets. More complex: The prediction of a single decision tree is easier to interpret when compared to a forest of them. Reduces over fitting. Higher accuracy compared to other models.
----------------------	---	---

Chapter 4

PROPOSED SENTIMENT ANALYSIS SYSTEM

People are usually interested in seeking positive and negative opinions, including likes and dislikes, shared by users for the feature of a particular product or service, which is called sentimentality interaction in the analysis process. Sentiment analysis (SA) is the type of analysis depending on Microblogging or social media data. Another definition of SA is identifying positive and passive opinions, emotions, and estimations, called polarity. Represents polarity either by different classes (e.g., positive, negative, neutral) or on a continuous scale of measurement of sentimentality, ranging from negative to positive. In this section, we will present proposed SA models. A sentiment analysis model has been created with feature extraction, sentiment classification, and sentiment evaluation of airline datasets combined with text and emojis analysis. We proved that the proposed models have top results that rely on improving whole sentiment analysis models of tweets compared to other works. The approach comprises four steps: collecting data, deleting unnecessary data in the process called (preprocessing), feature extraction data, then classifying the output. A dataset fits by adding classifiable status of context as a feature to assist the classification process. We have used two datasets with distinct data types for the process of SA.

4.1. Sentiment Analysis System (Pre-procissing)

Data Preprocessing Data Pre-processing in a sentiment analysis system refers to preparing the input dataset to make it suitable for classifying and training machine

learning systems to predict the testing data. Explore and clean the data are the first step as shown in **Fig.4-2**.

Figure 4-1 shows the structure for our proposed approaches models. The distinct steps involved in our framework are explained in the subsections below. Starting with collecting the datasets from The API Twitter (Tweepy) interface. Collected these data with the hashtag #US_airlines, #airlines_company. Our hash tags use the English language. After collecting 1.6 m tweets, we downsized these data to 10k tweets and deleted unnecessary data into the processes explained in the subsections. Preprocessing steps include the autoionization process, removing stop words, and URL.

The first model that we will explain its details in the subsections includes removing punctuation, removing emojis, and analyzing the text of tweets only as on the right side of **Figure 4-1**. Then, we applied Feature Extraction (FE) algorithms (BOW, TF-IDF, N-gram, W2V, and D2V) and applied sentiment classification (SC) algorithms as (XGBoost, RF, LR, SVM, and NB). Finally, we tested the performance by measuring the accuracy, recall, precision, and F-score.

In the second model, we analyzed the text and emojis together on the left side of **Figure 4-1**, as we will explain in the subsections, depending on sentiment polarity and emojis dataset. We converted ASCII emojis to emoticons symbols and added emoticons for tweets containing the words ("Sad", "Unhappy", "Crying", "Smile", "Happy", and "Love") within six and especially emojis.

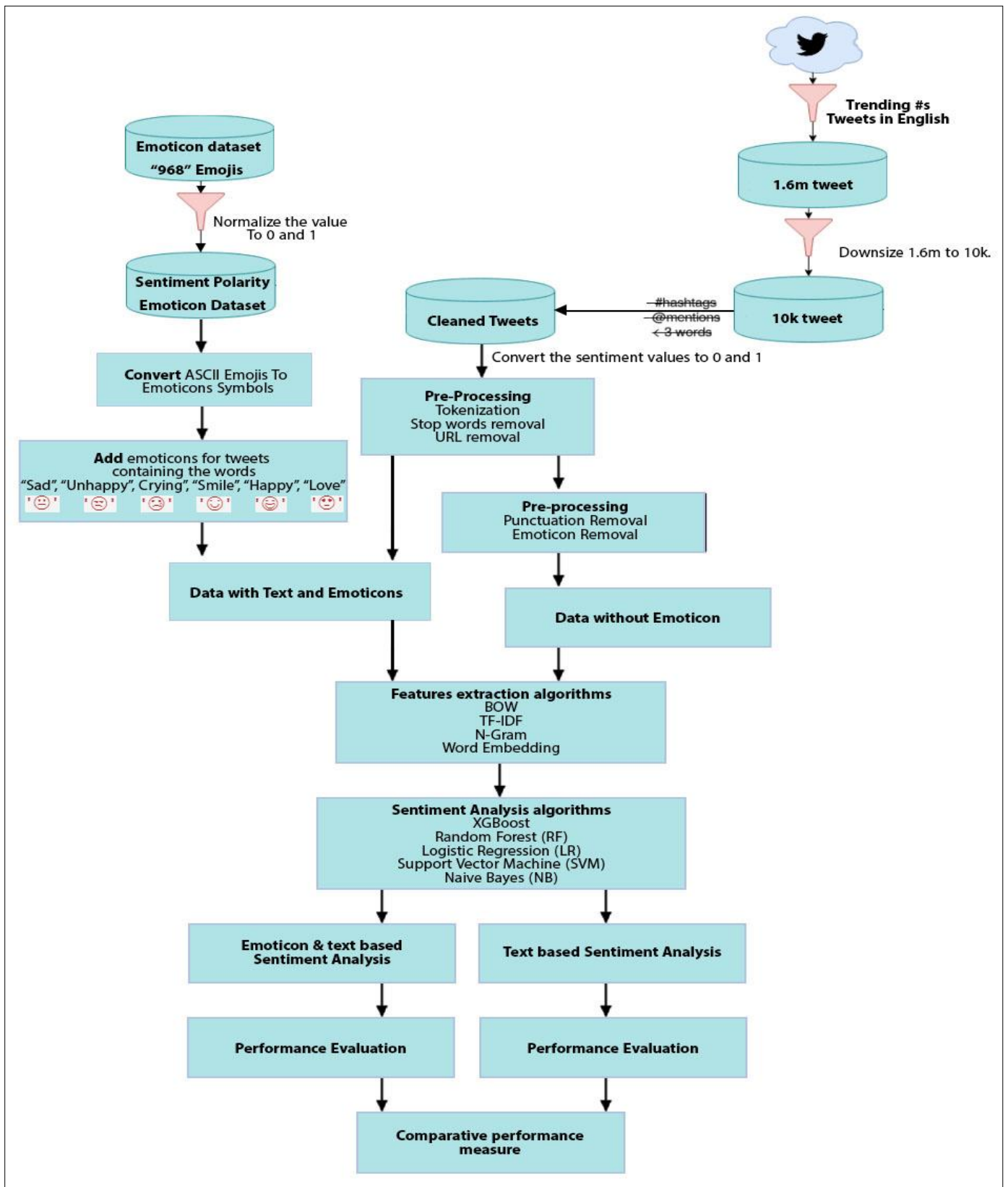


Figure 4-1 Proposed Sentiment Analysis Systems

4.2. Text Analysis Model

Microblogging's data is the data growing or created from Twitter. With the growth up the social media sits, the total amount of data on the web is growing exponentially. For example, there are over **200** million active Twitter users every month, and millions of tweets are published day by day, as in the statistics. Once the problem is defined, the valuable information in these data and the challenges to handle these data has attracted many scientists' interest in designing systems for Microblogging. Because social media provides unlimited sources for different topics, the data collection step plays the most critical role in the Microblogging analytics cycle. There are two ways to collect this data. After creating a Twitter developer account, some tweets were collected from the Twitter API using the python programming language. For example, Airplane Company was included and find out which topics were out. The other way to collect Microblogging data is to use CSV dataset files downloaded before. The most famous data format for machine learning is Comma-Separated Values (CSV) files. There are several considerations when loading our machine learning data and CSV files. **Figure 4-1**, data preprocessing has been made by removed words and tags from the tweets like ('@', 'HTTP', '//', '&', '#'). To implement the second model (text and emojis analysis) on the left side of **Figure 4-1**, We unstructured data through cleaned and converting it to a data matrix. After removing unnecessary data, we will be ready to use it and achieve high performance in the next step. Data visualization algorithms such as the process called (Word Cloud (WC)) clarify the sentiments contained across the dataset. One way to accomplish this task is by understanding the common words by plotting word sizing. Word Cloud is a visualization process wherein the most frequent words appear enormous, and the less frequent words appear in smaller sizes, as in **Figure 4-2**. It is a graphical representation of information and data. It provided a reachable way to see and understand trends, outliers, and patterns in data. A word cloud in **Figure 4-2** is an image composed of words of different sizes and colours. In the most common type of word clouds - and the one we used in this work - the size of the text corresponds to the frequency of the Word.

Algorithms like BOW, TF-IDF, N-gram, W2V, and D2V have been applied, as shown in Figure 4-1, and XGBoost, LR, RF, SVM, and NB to test passed SA and output our relative performance measure.

4.3. Emojis Data and Lexicon Approaches

The Second model was built to analyze the sentiment polarity of the text and Emojis of sentimental posts. Combining Emojis with text in Sentiment Classification (SC) to achieve better performance in prediction sentiment data. The Data Feature (DF) trains the machine learning models to have a massive effect on their performance. Having irrelevant features in data can shrink the accuracy of many models, Particularly linear algorithms like linear and logistic regression. The irrelevant or partly relevant feature can negatively affect model achievement. Two benefits of performing features extraction before modelling data are:

- 1) Several orthographic features, such as emoticons, expressive lengthening, and non-standard punctuation, have become popular in social media services, including Twitter.
- 2) Introducing emojis is a dramatic shift in online writing, potentially replacing this user-defined linguistic avoidance with pre-defined graphical icons.

Our methodology comprises three phases: data creation, processing, and sentiment analysis methods. Online writing lacks the non-verbal cues present in face-to-face communication, which provide additional contextual information about the utterance, such as the speaker's intention or affective state. User mentions, retweets and punctuation have been removed, and tags from the tweets since hyperlinks and tags add little to the sentiment of the past have been removed. In contrast to emoticons created from ASCII character sequences, emojis are represented by Unicode characters and continuously introduce new characters in each new Unicode version.¹ Emoji characters include faces and concepts and ideas such as weather, vehicles and buildings, food and drink, or activities such as running and dancing.

The dataset is listed in the column and listed in by row and assign a sentiment polarity value (1 & 0, positive and negative, respectively). Here we will find each text-based emojis and convert them into utf-8 emoticon symbols, also add emoticons on half of the tweets to increase the sentiment value of the utf-8 emoticons as **Table 4-1**, **Table 4-2**.

Table 4.1 Convert ASCII Emojis to Emoticons symbols


















‘:’), ‘: P’, ‘:D’, ‘: ’, ‘:’(”, ‘:O’, ‘:~’, ‘<3’, ‘: (’, ‘;)’, ‘xD’, ‘:/’, ‘=D’	ASCII Emoji
          	Emoticon symbols

Table 4-2 Public Six Emojis

Sad	Angry	Crying	Smile	Happy	Love
					

4.4. Text and Emojis Analysis Model

Sentiment analysis (SA), also known as Opinion Mining (OM), is a powerful tool that allows us to describe a user's emotional state based on their words. Most sentiment analysis uses text rather than emoticons, but adding emojis will add an extra analysis layer to help classify the messages further. Emoticons or emojis are widely used to express complex or straightforward emotions as icons commonly used on most online websites, forums, and chats such as social media websites. Data for sentiment analysis contain polarity associated with each Word, and these values are used to decide the overall sentiment of a sentence. We train the processed tweet dataset using classification algorithms to set the sentiments. We used our trained model to predict the extracted text from the input tweet. We averaged the sentiment value of the extracted emoticon and tweets to get the final sentiment polarity. This method trains our processed dataset as is, where it contains both emoticons and text. This method sets the sentiment of each emoticon. Dataset might skew the genuine sentiment of the emoticon, where we use a dataset containing both utf-8 emoticons and tweets texts.

The lexicon approach is a typical way to start sentiment analysis; entries are tagged with their a priori prior polarity in this lexicon. We will train it using our chosen classification algorithms. Then, we construct a new sentiment seed lexicon and analyze similarities and differences of sentiment words and emojis. Subsequently, we verify whether users' information can clarify emojis regarding sentiment and determine the sentiment polarity of ambiguous emojis. After these, we use emojis' position, sentiment, and semantic information to get emoji representations. The rich emotional information in emoji usage makes emoji prediction a suitable source task of sentiment and emotion detection. The proposed aim is to determine Emojis sentiment related to text sentiment. Then predict the unknown sentiment data. We collected the dataset with various packages to clean, analyze, and display data. Then, a search tweets function from tweets packages generated a raw user media data frame using the same naming conversation from an Emoji analysis. Used the same name convention from an Emoji analysis, counted the occurrence of Emojis within the data done, and matched Emojis to words that coexist with the same tweet. Our proposed work evaluated the Twitter dataset with a 1.6m tweet dataset with sentiment polarity and emoticon. The tweet dataset downloaded is processed data that converted the emoticons into words and contains sentiment polarity. We processed most of the dataset with sentiment polarity, manipulated, and simulated the tweet data set. Added the emoticons while retaining the sentiment polarity and ensured that the Emojis icons would have the corresponding sentiment polarity. After studying the relationship between textual similarity and Emojis polarity. We analyze the dataset to provide an insight into how the similarity of sentences changes based on the Emojis used. We compare the performance of standard sentence similarity models on our dataset using just Word embedding and Word and Emojis embedding and provide a comprehensive analysis of the results of the experiments.

4.5. Evaluation Metrics

The perfect method that we can use to measure the performance of a machine learning (ML) algorithm is to use different training and testing datasets after taking the original dataset and splitting it into two parts. We trained the algorithm, made predictions and evaluated the predictions against the expected results. Used 80% of the data for training and the remaining 20% for testing. It is ideal for large datasets (millions of records) with robust evidence that both data splits represent the evidence in a problem. The metrics that choose to evaluate our machine learning algorithms are essential. Choice of metrics influences how the performance of machine learning algorithms is measured and compared. They influence how we weigh the importance of unique characteristics in the results and our ultimate choice of which algorithm to select. Standard metrics calculate the system's performance (accuracy, precision, recall, and f-score). Let in a group of reviews with positive (class positive) and negative (class negative) reviews. Where (TP) was classified as positive with class positive, (FP) classified as positive with the class negative, (FN) classified as unfavorable with the class positive, (TN) classified as negative with class negative. **Accuracy:** estimated how many tweets are predicted correctly as belonging to a category of the positive or negative Word of tweets in the corpus [50] as in equation (9).

$$\textbf{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \quad (9)$$

Precision: estimated how many tweets are predicted correctly as belonging to a category (positive or negative). All the predicted tweets (correctly or incorrectly) belonging to the category were presented in the airline dataset as in equation (10).

$$\textbf{Precision} = \frac{TP}{TP+FP} \quad (10)$$

Recall: estimated how many tweets were correctly predicted as belonging to positive or negative sentiment. All the texts should have been predicted as belonging to the category as in equation (11).

$$\mathbf{Precision} = \frac{TP}{TP+FN} \quad (11)$$

F-score: evaluated the accuracy of the proposed system about both the precision and the recall rate values as in equation (12).

$$\mathbf{F-score} = \frac{2*Precision*Rcall}{Precision+Rcall} \quad (12)$$

Chapter 5

EXPERIMENTAL RESULTS AND DISCUSSION

Social media, namely Microblogging, is an essential part of people's daily routines and the business of some. As a result, the system described in this work proposes an approach for evaluating sentiment score at the word and emojis levels with the increased popularity of emojis in textual communication. This research focuses on considering how emojis compete with emoticons to communicate the prediction content of Microblogging data. We review some of the most popular (ML) methods and their applicability to the problem of classification Microblogging data using sentiment analysis classification. The sentiment analysis of brand opinions can communicate to companies the level of satisfaction among consumers. It is carried out using TextBlob analyzer tools and data pre-processing analyzes techniques such as normalizing, word cloud data. Consumer loyalty prediction is performed using Twitter data. The experiment shows the top results recall percentage. Using five features extraction Twitter data and four performance evaluations (accuracy, recall, f-score and precision). By presenting two models for classification data, first for text analysis only and second for text and emojis analysis together by using two Twitter datasets. Data can be essential information by understanding a company's standing within consumers and regarding sentiment scores to infer consumer feedback.

5.1. Experimental Setup

Experiments are implemented on laptops with an Intel ® Core™ i7 –2670M CPU 2.20GHz, 64-bit Windows 7 operating system, and 16 GB RAM. Also, some tools were used, such as Python 3.7, Jupiter lab, Matplotlib, NumPy, pandas, anaconda3 (version 4.5.12), and Scikit learn.

5.2. Data sets

Microblogging data like tweets can explain the genuine sentiment cores for each airline. Two datasets for airlines companies from social media and a dataset for emojis analysis to create text and emojis, analysis models.

5.2.1. Airline Dataset1 (text analysis)

The Airline dataset contains (14640) English language reviews, with positive reviews (**2363**), negative reviews (**9178**), and neutral reviews (**3099**) as presented in **Table 5-1**. Dataset split into two types of data (training and testing data), where **80%** is training data, and **20%** is testing data. Graphically represents the positive and negative mean sentiment scores for airlines. We have discovered the varieties of positive and negative sentiments for the various airlines. Sentiment score is significant for airlines to perceive users' sentiment opinion about their services at a point in time.

Table 5-1 Airline dataset sentiment value

Sentiment Value	Value Counts	Percentage
Negative	9178	62.69%
Neutral	3099	21.17%
Positive	2363	16.14%








Any airline wants to confirm that their customer's positive opinion score is more significant than their negative opinion score. These scores also resource in competitor advantage.

5.2.2. Dataset2 (Text and emojis analysis)

Emojis are used alongside text in social media communication, but their seeable nature allows sentimental representation. Using the emoticon dataset, simulated, replaced, and added utf-8 emoticons to the dataset. After downloading the 1.6 m tweet dataset, we downsize the 1.6m tweet to 10k.

Then, we remove the name tags and links since most would not contribute to the value of the sentiment and convert the sentiment values to 0 and 1. The emoticon dataset has sentiment; the sentiment is formatted by the voting value of the data acquirer. Create a Separate Sentiment Polarity Emoticon Dataset. Emoticons dataset have "968" different emojis now with positive and negative sentiment polarity, "794" positive emojis, "173" negative emojis. The dataset has 10.000 tweets, with 500 positive and 500 negative. These posts or tweets have "46" positive and "28" negative emojis. **Table 5-2** presented examples of emojis in the dataset.

Table 5-2 Example of emojis analysis

Emojis	Sentiment
	1
	0
	1
	0
	1
	0
	1

5.3. Experimental Results and Discussion

5.3.1. Text Analysis Results

The efficiency of the algorithmic rule depends on the accuracy results. The comparison had between the five algorithms with text-only analysis by using dataset1. The following tables show the results and performance using different techniques of text representation and classification algorithms. Results in **Table 5-3**, evaluation metrics using BOW features, show that, in the case Bag of Words features, the system achieves the highest performance using the random forest (RF) technique for sentiment analysis with an accuracy of **93%**. Comparing the results of the classification techniques, Naive Bayes are the lowest performance with an accuracy of **77%**.

Table 5-3 Evaluation Metrics Using BOW Features

	Bag Of Words
	Accuracy
XGBoost	0.87
RF	0.93
LR	0.80
SVM	0.80
NB	0.77

With TF-IDF features approach, the developed system achieves the highest performance using the random forest (RF) technique for sentiment analysis with an accuracy of **93%**. Naive Bayes is the worst performance with an accuracy of **75%**.

Table 5-4 Evaluation Metrics Using TF-IDF Features

	TF-IDF
	Accuracy
XGBoost	0.91
RF	0.93
LR	0.80
SVM	0.80
NB	0.75

In **Table 5-5**, in the case of the N-gram features, the system achieves the highest performance using the XGBoost technique sentiment analysis with an accuracy of **94%**. Naive Bayes is the worst performance having an accuracy of **67%**.

Table 5-5 Evaluation metrics using N-Gram features

	N-gram
	Accuracy
XGBoost	0.94
RF	0.94
LR	0.75
SVM	0.75
NB	0.67

In Table 5-6, with the Word2Vec feature, the system achieves the highest performance using XGBoost and RF techniques for sentiment analysis with an accuracy of **95%**. Naive Bayes is the lowest performance with an accuracy of **70%**.

Table 5-6 Evaluation metrics using Word2Vec features

	Word 2 Vec
	Accuracy
XGBoost	0.95
RF	0.95
LR	0.77
SVM	0.78
NB	0.70

In **Table 5.7** with Doc2Vec features, the system achieves the highest performance using the XGBoost technique for sentiment analysis with an accuracy of **81%**. Naive Bayes is the worst performance with an accuracy of **61%**.

Table 5-7 Evaluation metrics using Doc2Vec features

	Doc 2 Vec
	Accuracy
XGBoost	0.81
RF	0.71
LR	0.69
SVM	0.68
NB	0.60

We can see that the XGBoost and random forest models' estimated accuracy was approximately **95%** using the Word2vector algorithm. We notice that, besides specifying the size of the split, we also specify the random seed. Because the data split is random, we want to ensure that the results are reproducible. Specifying the random seed ensures we get the same random numbers each time we run the code and the same split of data. This is important if we want to compare this result to the estimated accuracy of another machine learning algorithm or the same algorithm with a different configuration. We ensure they are trained and tested on precisely the same data.

5.3.2. Comparison between related system and proposed system Results

After compared the results between the related system in [63] and this work. For pre-processing in [63], firstly tokenize and remove stop words, URLs, digits, punctuation, and emoticons. In this paper, making distinct steps, first, system read data and save the dataset in a CSV file and make process called part of speech (POS) tagging, removing stop words dictionary, and numbers. Various forms of the same word were lemmatized by converting them to the keyword using the WordNet Lemmatize module of the Natural Language Toolkit Python library. Then, making the stemming data, visualize data, and normalized the data output. Also, this work uses BOW, TF-IDF, N-gram, Word2Vec, and Doc2Vec, as features extraction and RF, LR, SVM, and NB as classification data. Table.6, shows a comparison between the related system in [63] and the devolved system in this paper. Results in Table 6 show that with Random forest, the best result of accuracy was 93% with BOW and 93% with TF-IDF. With using LR, the best accuracy was 80% with BOW and 80% with TF-IDF. But with using SVM, the best accuracy was 80% with BOW, and 80% with TF-IDF. Also in the case using NB, the best accuracy result was 77% with BOW, and 75% with TF-IDF. In this work, N-gram, Word2Vec, and Doc2Vec feature extractions are used for more performance investigation. In case of RF, the proposed system achieves the best accuracy result (94% with n-gram, 95% with Word2Vec), and the lowest result was 71% with Doc2Vec as shown in Table 5-8.

Table 5-8. Comparison between existing and proposed system

	Related system Accuracy In[63]	Proposed system Accuracy				
		BOW	TF-IDF	N-gram	Word2Vec	Doc2Vec
RF	76%	93%	93%	94%	95%	71%
LR	78%	80%	80%	75%	77%	69%
SVM	78%	80%	80%	75%	78%	68%
NB	52%	77%	75%	67%	70%	60%

5.3.3. Emojis Analysis Results

The tweets contain tags and links which need to be removed. The emoticons were transformed into plain words or words with symbols. ASCII emoticons and word emoticons were converted to UTF-8 emoticons.

Emojis were also randomly added by sentiment to balance the data by manipulating and stimulating the tweet data set and adding the emoticons while retaining the sentiment polarity. This will ensure that the emoji icons have the corresponding sentiment polarity we need. The dataset contains 968 emojis with 'Emoji', 'Negative', 'Neutral', 'Positive', and 'Unicode name'. We find that emojis highly correlated with sentiment and irony provide little semantic content and are also the most popular emojis [78]. With the examination of the percentage degree of different techniques, the accuracy of text reviews content for computing the accuracy of each model by calculating the prediction of the positive and negative proportion given by each technique.

5.3.4. Emojis And Text Results

Table 5-9 presents the accuracy percentage for the two compared models (text only and text with emojis analysis). Text and emojis analysis together get better accuracy results than analysis of the text only, training data 80%, and testing data 20%.

Table 5-9 Result of (Text and Emojis) analysis dataset2 using TF-IDF features

Classification methods	Text and Emojis	Text
LR	0.81	0.78
NB	0.80	0.80
RF	0.78	0.77
SVM	0.80	0.76
XGBoost	0.78	0.74

The emoticons are assigned with their sentiment polarity, where the training of the tweets is separate from the emoticons. We compute the mean sentiment value of both the emojis and texts to analyze the tweets. This method will have a powerful influence on emoticon with a non-changing polarity value.

Chapter 6

CONCLUSION AND FUTURE WORK

6.1. Conclusion

This work has two full models for testing the best performance of preprocessing techniques, features extraction approaches, and classification algorithms. Also, we create a complete model for the analysis of emojis and text. Focus on data with unique characteristics, namely Microblogging data, i.e. Twitter. By presenting two models for classification data, first for text analysis only and second for text and emojis analysis together by using two Twitter datasets. We have observed variations in positive and negative sentiments for the various airlines. An airline would want to ensure that their positive sentiment score is more significant than their negative sentiment scores. These scores also aid in competitor advantage, as an airline can work towards making their positive scores more substantial and adverse scores lesser than their competing airlines. After downloading the 1.6 m tweet dataset, we downsize the 1.6m tweet to 10k. With Bag of words, the system achieves the highest performance using the random forest (RF) technique for sentiment analysis with an accuracy of 93%. Comparing the results of the classification techniques, Naive Bayes are the lowest performance with an accuracy of 77%.

With the TF-IDF features approach, the developed system achieves the highest performance using the random forest (RF) technique for sentiment analysis with an accuracy of 93%. Naive Bayes is the lowest performance with an accuracy of 75%. With the N-gram features approach, the developed system achieves the highest performance using the XGBoost technique sentiment analysis with an accuracy of 94%. Naive Bayes is the lowest performance having an accuracy of 67%. With the Word2Vec features approach, the developed system achieves the highest performance using XGBoost and RF techniques for sentiment analysis with an accuracy of 95%.

Naive Bayes is the lowest performance with an accuracy of 70%. With the Doc2Vec features approach, the developed system achieves the highest performance using the XGBoost technique for sentiment analysis with an accuracy of 81%. Naive Bayes is the lowest performance with an accuracy of 61%. We can see that the XGBoost and random forest models' estimated accuracy was approximately 95% using the Word2vector algorithm. We notice that, besides specifying the size of the split, we also set the random seed. Because the data division is arbitrary, we want to ensure that the results are reproducible. By specifying the random basis, we get the same random numbers each time we run the code and the same split of data. The other model that is using emojis and text for testing sentiment analysis. ASCII emoticons and word emoticons were converted to utf-8 emoticons. Emoticons were also randomly added by sentiment to balance the data by manipulating and stimulating the tweet data set and adding the emoticons while retaining the sentiment polarity. It will ensure that the emoji icons have the corresponding sentiment polarity we need. The dataset contains 968 emojis with 'Emoji', 'Negative', 'Neutral', 'Positive', and 'Unicode name'. We find that emojis highly correlated with sentiment and irony provide little semantic content and are also the most popular emojis. Text and emojis analysis get better accuracy results than analysis of the text only. Evaluation results stated that in this case, Word2Vec feature extraction with (XGBoost and random forest) classifiers achieves the highest performance. And in case, Doc2Vec and the Naive Bayes classifier perform the lowest commission.

6.2. Future Work

In future, we aim at expanding the approach to:

1. Directed toward implementing deep learning techniques for building the sentiment analysis system.
2. Explore people's interpretations of emojis regarding the contexts in which they appear. Another exciting avenue for future work lies in the potential for cultural differences in performing emoji.

REFERENCES

- [1] D. M. Boyd and N. B. Ellison, "Social Network Sites: Definition, History, and Scholarship," *Journal of Computer Mediated Communication* 2007.
- [2] <https://twitter.com>
- [3] B. Xie, I. Vovsha, O. Rambow, and R. Passonneau, "Sentiment Analysis Of Twitter Data, " In *Proceedings Of The Workshop On Languages In Social Media*, Columbia University, New York, 2011.
- [4] S. Maheshwari, Is. Shukla, and D. Kumari, "Twitter Opinion Mining Using Sentiment Analysis," *World Scientific News, An International Scientific Journal*, 2019.
- [5] C. Strapparava and A. Valitutti, "Wordnet-Affect: An Affective Extension Of Wordnet," In: *Proceedings Of The International Conference On Language Resources And Evaluation (LREC'04)*, 2004.
- [6] <https://www.tumblr.com>
- [7] <https://www.internetlivestats.com/twitter-statistics>
- [8] Basile and Pierpaolo et al., "Sentiment Analysis Of Microblogging Data," In *Book: Encyclopedia Of Social Network Analysis And Mining*, 2018.
- [9] T. Wilson, J. Wiebe, and P. Hoffmann, "Recognizing Contextual Polarity In Phrase-Level Sentiment Analysis," In: *Proceedings Of The Conference On Human Language Technology And Empirical Methods In Natural Language Processing*, 2005.
- [10] M. Hu and B. Liu, "Mining And Summarizing Customer Reviews," In: *Proceedings Of The Tenth ACM SIGKDD International Conference On Knowledge Discovery And Data Mining*, 2004.
- [11] <http://www2.imm.dtu.dk/pubdb/P.php?6010>
- [12] <http://saifmohammad.com/webpages/lexicons.html>
- [13] A. Esuli, and F. Sebastiani, "Sentiwordnet: A Publicly Available Lexical Resource For Opinion Mining," In: *Proc. Of LREC*, 2006.

-
- [14] C. Strapparava and A. Valitutti, "Wordnet-Affect: An Affective Extension Of Wordnet," In: Proceedings Of The International Conference On Language Resources And Evaluation (LREC'04), 2004.
 - [15] C. Fellbaum, "Wordnet: An Electronic Lexical Database," Bradford Books, 1998.
 - [16] Lackermair, Georg, D. Kailer, and K. Kanmaz, "Importance of Online Product Reviews from a Consumer's Perspective," Advances in Economics and Business, 2013.
 - [17] M. Ogneva, "How Companies Can Use Sentiment Analysis To Improve Their Business," In: Retrieved August 30, 2010.
 - [18] A. Wright, "Mining the Web for Feelings, Not Facts," In: The New York Times 24, 2009.
 - [19] Mali, Manisha, Atique, and Mohammad, "Applications of Text Classification Using Text Mining," International Journal of Engineering Trends and Technology, 2014.
 - [20] Awad, W. Abou, and S. M. Elseuofi, "Machine Learning Methods for Spam E-Mail Classification," International Journal of Computer Science & Information Technology, IJCSIT, 2011.
 - [21] Y. Liang-Chih, W. Jheng-Long, C. Pei-Chann, and C. Hsuan-Shou, "Using A Contextual Entropy Model To Expand Emotion Words And Their Intensity For The Sentiment Classification Of Stock Market News," Knowl-Based SYST, 2013.
 - [22] M. Hagenau, M. Liebmann, and D. Neumann, "Automated News Reading: Stock Price Prediction Based On Financial News Using Context-Capturing Features," Decis Supp SYST, 2013.
 - [23] T. Xu, P. Qinke, and C. Yinzhao, "Identifying the Semantic Orientation of Terms Using S-Hal for Sentiment Analysis," Knowlbased SYST, 2012.
 - [24] M. Isa, and V. Piek, "A Lexicon Model For Deep Sentiment Analysis and Opinion Mining Applications," Decis Support SYST, 2012.
 - [25] J. Moore, E. Kouloumpis, and T. Wilson, "Twitter Sentiment Analysis: The Good, The Bad And The Omg!," In Fifth International AI Conference on Weblogs and Social Media, 2011.
 - [26] B. Pang and L. Lillian, "Opinion Mining and Sentiment Analysis," Comput. Linguist, 2009.
-

-
- [27] Y. Yamamoto, T. Kumamoto, and A. Nadamoto, "Role of Emoticons for Multidimensional Sentiment Analysis of Twitter," In: Proceedings of the 16th International Conference on Information Integration and Web-Based Applications & Services, ACM, 2014.
 - [28] M. Ghiassi, J. Skinner, and D. Zimbra, "Twitter Brand Sentiment Analysis: A Hybrid System Using N-Gram Analysis And Dynamic Artificial Neural Network," In An Expert System With Applications, Science Direct, 2013.
 - [29] <https://twitter.com/Justinbieber>
 - [30] T. Mikalai, and P. Themis, "Survey on Mining Subjective Data on the Web," Data Min Knowl Discov, 2012.
 - [31] R. Plutchik, "A General Psychoevolutionary Theory of Emotion," Emotion Theory Res Exp, 1980.
 - [32] L. Cheng-Yu, L. Shian-Hua, L. Jen-Chang, C. Samuel, and H. Jen-Shin, "Automatic Event-Level Textual Emotion Sensing Using Mutual Action Histogram Between Entities," Expert Syst Appl, 2010.
 - [33] S. Poria, E. Cambria, and A. Gelbukh, "Aspect Extraction For Opinion Mining With A Deep Convolutional Neural Network," Knowledge-Based Systems, 2016.
 - [34] Weichselbraun, S. Gindl, F. Fischer, S. Vakulenko, and A. Scharl, "Aspect-Based Extraction And Analysis Of Affective Knowledge From Social Media Streams," IEEE Intelligent Systems, 2017.
 - [35] M. Andre'S, M. Patricio, and B. Alexandra, "Subjectivity and Sentiment Analysis: An Overview of the Current State of the Area and Envisaged Developments," Decis Support Syst, 2012.
 - [36] <https://monkeylearn.com/>
 - [37] C. P. Kumar and L. D. Babu, "Novel Text Preprocessing Framework for Sentiment Analysis," In Smart Intelligent Computing And Applications, Springer, 2019.
 - [38] H. Miller et al., "Blissfully Happy" Or "Ready To Fight": Varying Interpretations of Emoji" , 2016.
 - [39] B. Felbo, A. Mislove, A. Sagard, I. Rahwan, and S. Lehmann, "Using Millions of Emoji Occurrences To Learn Any-Domain Representations For Detecting Sentiment, Emotion And Sarcasm," ARXIV Preprint Arxiv, 2017.
-

-
- [40] A. Gprasad, Sanjana, S. M. Bhat, and B. Harish, "Sentiment Analysis for Sarcasm Detection On Streaming Short Text Data," International Conference On Knowledge Engineering And Applications (ICKEA), 2017.
 - [41] Liu and Bing, "Sentiment Analysis and Opinion Mining," Synthesis Lectures on Human Language Technologies, 2012.
 - [42] M. Kuutla, M. V. Mantyla, M. Claes, M. Elovainio, and B. Adams," Using Experience Sampling To Link Software Repositories with Emotions And Work Well-Being," In Proceedings Of The 12th ACM / IEEE International Symposium On Empirical Software Engineering And Measurement, 2018.
 - [43] D. Girardi, F. Lanubile, N. Novielli, L. Quaranta, and A. Serebrenik, "Towards Recognizing The Emotions Of Developers Using Biometrics: The Design Of A Field Study," In Proceedings Of The 4th International Workshop On Emotion Awareness In Software Engineering, 2019.
 - [44] F. Calefato, F. Lanubile, N. Novielli, and L. Quaranta, "Emtk: The Emotion Mining Toolkit," In Proceedings of The 4th International Workshop On Emotion Awareness In Software Engineering, Semotion@Icse, 2019.
 - [45] D. Kaneko, A. Toet, S. Ushihama, A. Brouwer, V. Kallen, and J. B.F. V. Erp, "Emoji Grid: A 2d Pictorial Scale For Cross-Cultural Emotion Assessment Of Negatively And Positively Valenced Food," Food Research International, 2019.
 - [46] G. S. Kumar, P. Dinh, A. Brett, and V. Svetha, "Regularized Nonnegative Shared Subspace Learning," Data Min Knowl Discov, 2012.
 - [47] G. Guibon, M. Ochs, and P. Bellot, "From Emojis to Sentiment Analysis," In: WACAI, 2016.
 - [48] J. Suttles and N. Ide, "Distant Supervision For Emotion Classification With Discrete Binary Values," In International Conference On Intelligent Text Processing And Computational Linguistics, Pages 121–136, Springer, 2013.
 - [49] G. Rao, W. Huang, Z. Feng, and Q. Cong, "LSTM with Sentence Representations For Document-Level Sentiment Classification," Neurocomputing , 2018.
 - [50] D. El-Din, "Enhancement Bag-Of-Words Model For Solving the Challenges of Sentiment Analysis," International Journal of Advanced Computer Science And Applications, 2016.
-

-
- [51] Ramadhan, W. P. Novianty, S. Astri, Setianingsih, and S. Casi. "Sentiment Analysis Using Multinomial Logistic Regression," International Conference On Control, Electronics, Renewable Energy And Communications (ICCREC), IEEE, 2017.
 - [52] Mccarthy and Carroll, "Disambiguating Nouns, Verbs, and Adjectives Using Automatically Acquired Sectional Preferences," Association For Computational Linguistics, 2003.
 - [53] Das, Bijoyan, Chakraborty, and Sarit, "An Improved Text Sentiment Classification Model Using Tf-Idf and The Next Word Negation," ARXIV, 2018.
 - [54] L. CAI-Zhi, S. Yán-Xiu, W. Zhi-Qiang, and Y. Yong-Quan, "Research of Text Classification Based On Improved Tf-Idf Algorithm," International Conference Of Intelligent Robotic And Control Engineering (IRCE), IEEE, 2018.
 - [55] U. Pavalanathan and J. Eisenstein, "Emoticons vs. Emojis on Twitter: A Causal Inference Approach," In: ARXIV, 2015.
 - [56] M. Butler and V. Keselj, "Financial Forecasting Using Character N-Gram Analysis And Readability Scores Of Annual Reports," Advances In AI, 2009.
 - [57] Lim, Y. Qing, L. C. MING, G. K. Hoon, and S. N. Hana, "Text Sentiment Analysis On Twitter To Identify Positive Or Negative Context In Addressing Inept Regulations On Social Media Platform," Symposium On Computer Applications & Industrial Electronics (ISCAIE), IEEE, 2020.
 - [58] Wan and Q. GAO, "An Ensemble Sentiment Classification System of Twitter Data For Airline Service Analysis," International Conference On Data Mining Workshop (ICDMW), 2015.
 - [59] M. Tomas, C. Kaï, C. Greg, and D. Jeffrey, "Efficient Estimation of Word Representations In Vector Space," ARXIY, 2013.
 - [60] D. Karani, "Introduction To Word Embedding And Word2Vec," Towards Data Science, 1September 2018.
 - [61] Alessia and D., et al., "Approaches, Tools And Applications For Sentiment Analysis Implementation," International Journal Of Computer Applications, 2015.
 - [62] Brownlee and Jason, "Master Machine Learning Algorithms: Discover How They Work And Implement Them From Scratch," Machine Learning Mastery, 2016.
-

-
- [63] U. M. Aman, M. S. Maliha, B. S. Ara, and D. N. SAHA, "An Algorithm And Method For Sentiment Analysis Using The Text And Emoticon," *ICT Express*, 2020.
 - [64] <https://Github.Com/Dmlc/Xgboost/Tree/Master/Demo#Machine-Learning-Challenge-Winning-Solutions>
 - [65] R. Buyya, Calheiros, N. Rodrigo, Dastjerdi, and A. Vahid, "Big Data: Principles and Paradigms," Morgan Kaufmann, 2016.
 - [66] S. Rani and N. Gill, "Hybrid Model for Twitter Data Sentiment Analysis Based On the Ensemble of Dictionary-Based Classifier And Stacked Machine Learning Classifiers-Svm, Knn And C5," *Journal Of Theoretical And Applied Information Technology*, 2020.
 - [67] Misra, L. Siddharth, H. Hao, and Jiabo, "Machine Learning For Subsurface Characterization," Gulf Professional Publishing, 2019.
 - [68] B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs Up?: Sentiment Classification Using Machine Learning Techniques," In *Proceedings Of The ACL-02 Conference On Empirical Methods In Natural Language Processing Volume 10*, 2002.
 - [69] S. Bird and E. Loper, "Nltk: The Natural Language Toolkit," In *Proceedings of the ACL Interactive Poster and Demonstration Sessions, Barcelona, Spain, Association For Computational Linguistics*, 2004.
 - [70] F. Huang, S. Zhang, J. Zhang, and G. Yu, "Multimodal Learning For Topic Sentiment Analysis In Microblogging," *Neurocomputing, Learning Multimodal Data*, 2017.
 - [71] T. Hu, H. Guo, H. Sun, Thuy-Vy, T. Nguyen, and J. Luo, "Spice Up Your Chat: The Intentions And Sentiment Effects Of Using Emojis," In *Proceedings Of The 11th International Conference On Web And Social Media*, 2017.
 - [72] X. Lu, W. Ai, X. Liu, Q. Li, N. Wang, G. Huang, and Q. Mei, "Learning From The Ubiquitous Language: An Empirical Analysis Of Emoji Usage Of Smartphone Users," In *Proceedings Of The 2016 Acm International Joint Conference On Pervasive Computing, UBIComp 6*, 2016.
 - [73] Chen, Zhenpeng, et al., "Emoji-Powered Sentiment And Emotion Detection From Software Developers Communication Data," *ACM, Transactions On Software Engineering And Methodology (TOSEM)*, 2021.
-

-
- [74] B. Felbo, A. Mislove, A. Søgaard, I. Rahwan, and S. Lehmann, “Using Millions Of Emoji Occurrences To Learn Any-Domain Representations For Detecting Sentiment, Emotion And Sarcasm,” In Proceedings Of The Conference On Empirical Methods In Natural Language Processing, Pages 1615–1625, Copenhagen, Denmark, 2017.
 - [75] J. Deriu, M. Gonzenbach, F. Uzdilli, A. Lucchi, V. D. Loca, and M. Jaggi, “Swiss Cheese at Semeval: Sentiment Classification Using An Ensemble Of Convolutional Neural Networks With Distant Supervision,” In Proceedings of the 10th International Workshop On Semantic Evaluation, California, Association for Computational Linguistics, 2016.
 - [76] J. Park, Y. M. Baek, and M. Cha, “Cross-Cultural Comparison of Nonverbal Cues In Emoticons On Twitter: Evidence From Big Data Analysis,” Journal Of Communication, 2012.
 - [77] Boy, S. Ruiter, D. Klakow, and Dietrich, “Emoji-Based Transfer Learning For Sentiment Tasks,” ARXIV, 2021.
 - [78] Debnath and Alok et al., “Semantic Textual Similarity of Sentences With Emojis,” In: Companion Proceedings Of The Web Conference, 2020.

PUBLICATIONS

- [1] Raghdah Elnadree, Ashraf El-Sisi, Walid Atwa. “Performance Investigation Of Features Extraction And Classification Approaches For Sentiment Analysis Systems.” IJCI. International Journal Of Computers And Information (2021).

السيرة الذاتية لمقدم الرسالة

الاسم : رغده شريف خالد النادري.

الوظيفة : محاضر فى علوم البيانات لشركه خاصه

جهة العمل : داتا كامب

تاريخ الميلاد : 22 يونيو 1989

العنوان : طنطا- الغربية.

الدرجة العلمية : بكالوريوس حاسبات ومعلومات فى علوم الحاسب

جهة منح الدرجة العلمية : كلية الحاسبات والمعلومات جامعة المنوفية.

تاريخ تسجيل درجة الماجستير : اكتوبر 2020

تاريخ الحصول على درجة الماجستير : يونيو 2022

ملخص الرسالة

تحليل المشاعر هو منهجية قائمة على معالجة اللغة لتقدير مدى إيجابية أو سلبية المشاعر التي يعبر عنها مقطع من النص. غالبًا ما يتم استخدام تحليل المشاعر لتصنيف رسائل ملاحظات العملاء أو مراجعات المنتجات تلقائيًا. تعتمد تحليل مشاعر النصوص المكتوبة في وسائل التواصل الاجتماعي على سمات خاصة وعلى معلومات نحوية ودلالية للجمل تتطلب العديد من المنهجيات والعمليات المندرجة تحت تحليل النصوص. في الآونة الأخيرة ، أبدى العديد من العلماء اهتمامًا باستخدام الميزات لمهام تصنيفات النصوص والرموز التعبيرية. لذلك قد بدانا في تحليل النصوص والرموز التعبيرية لتصنيف المدونات الصغيرة فالمنهج القائم له خطوات ونتائج قائمة على التجربه وعلى التنفيذ وعلى اختبار النتائج ومقارنتها بنتائج لها نفس الخواص وعلى اكثر من قاعده بيانات. الاقتراح هو : 1) تصميم نظام تحليلي قابل للتطوير وفعال وكامل الخطوات. 2) تنفيذ النظام بخمسة مسارات رئيسيه وهي (جمع البيانات ومعالجتها واستخراج سماتها الاساسيه وتصنيفها وقياس اداء النتائج و تقييمها). 3) تحليل الرموز التعبيرية كسمات اساسيه خاصة بمحتوى وسائل التواصل الاجتماعي وخاصة (تويتر). 4) تدريب النموذج على تحليل النصوص وتحليل ايضا الرموز التعبيرية معا في اكثر من نموذج واستخراج نتائج هذا التحليل بين النصوص والرموز التعبيرية وهل يؤثر تحليل النص عندما يكون مرتبط برموز تعبيريه ومدى مصداقيه هذه الرموز في تحليلها عندما تشتمل على اكثر من معني و الارتباط بين الكلمات والرموز التعبيرية المختلفة. يتم استخدام خمس ميزات استخراج اساسيه في نماذجنا وهي (حقيية الكلمات (BOWs) و TF-IDF (تردد المصطلح - تردد المستند العكسي) و Nigram و Word emedding) لاستخراج مميزات النص لتصنيف المشاعر وتحليل الرموز التعبيرية. يتم استخدام خمسة مصنفات وهي (الانحدار اللوجستي (LR) ، وآلة المتجهات الداعمة (SVM) ، و Naïve Bayes (NB) ، والغابة العشوائية (RF) ، و XGBoost) لمقارنة أداء الميزات المختلفة المحددة لتحليل المشاعر. تم إنشاء نماذجنا المقترحة كالآتي: 1) نموذج لتحليل النص فقط واختبار نتائج الأداء. 2) نموذج لتحليل النص والرموز التعبيرية معًا

واختبار نتائج الأداء معاً. 3) مقارنة النتائج بين هذه النماذج والنماذج الأخرى التي تستخدم نفس مجموعات البيانات. تظهر النتائج تحقيق أداء عالٍ عند استخدام نهج Word2Vec مع خوارزميات تصنيف XGBoost و Random Forest. حيث يعتبر Naive Bayes أقل أداء. تتميز النماذج بنتائج أكثر كفاءة ودقة وأفضل أداء. يحصل تحليل المشاعر النصية والرموز التعبيرية معاً على نتائج تقييم أفضل في الدقة من تحليل المشاعر النصية فقط. أداء النماذج المقترحة مشجع ، وأكثر فاعلية من غيرها لتحليل المشاعر.

تم تنظيم الرسالة على النحو التالي:

- الفصل الأول: عرض المقدمة وخلفية مشاكل تطوير النماذج وأهداف البحث وصياغة المشكلة والمساهمات البحثية في كيفية حل المشكلات والنتائج.
- الفصل 2: عرض خلفية العمل باستخدام مفهوم بيانات المدونات الصغيرة ، وخصائص المدونات الصغيرة ، وتقنيات تحليل المشاعر (خلفية اكتشاف المشاعر ، وخلفية تحليل المشاعر القائمة على الجانب ، وخلفية تحليل المشاعر متعدد اللغات ، وخلفية موارد البناء ، وخلفية نقل التعلم) ، وتقنيات تحليل الرموز التعبيرية.
- الفصل 3: عرض السمات واستخراج نهج التصنيف. استخدم استخراج الميزات خوارزمية حقيقية من الكلمات (BOW) ، وخوارزمية تردد المصطلح - خوارزمية تردد المستند العكسي (TF-IDF) ، وخوارزمية N-gram ، وخوارزمية تضمين الكلمات (WE). نهج تصنيف المشاعر مثل خوارزمية XGBoost وخوارزمية الانحدار اللوجستي (LR) وخوارزمية Random Forest (RF) وخوارزمية دعم Vector Machine (SVM) وخوارزمية (Naïve Bayes (NB).

- الفصل 4 : اقترح نظام تحليل المشاعر باستخدام تحليل البيانات النصية وتحليل بيانات الرموز التعبيرية

ومنهج المعجم.

• الفصل 5 : عرض النتائج الأولية والمناقشة. الإعداد التجريبي وتفاصيل مجموعات بيانات شركات الطيران

ونائج تحليل النص ونتائج تحليل الرموز التعبيرية ونتائج تحليل النصوص والرموز التعبيرية وايضا المبادئ

التوجيهية للعمل المستقبلي في مجال البحث.

• الفصل 6 : تقديم الخاتمة والمبادئ التوجيهية للعمل المستقبلي في مجال البحث لتقدير بيانات الاستشعار.



جامعة المنوفية
كلية الحاسبات والمعلومات
قسم علوم الحاسب

تحسين تحليل المشاعر لنظام المدونات الصغيرة

رسالة مقدمة إلى كلية الحاسبات والمعلومات-جامعة المنوفية لاستكمال متطلبات
الحصول على درجة الماجستير في الحاسبات والمعلومات
[علوم الحاسب]

مقدمة من:

رغده شريف خالد النادري

تحت إشراف

وليد سعيد عطوه
استاذ مساعد علوم الحاسب
كلية الحاسبات والمعلومات
جامعة المنوفية
صار
[علوم الحاسب]

ا.د أشرف السيسي
أستاذ علوم الحاسب
كلية الحاسبات والمعلومات
جامعة المنوفية
[علوم الحاسب]