Republic of Iraq

Ministry of Higher Education and

Scientific Research

University of Diyala

College of Science

Department of Computer Science

Sentiment Polarity Identification Framework of Tweets

A thesis
Submitted to the Department of Computer Science\ College of Sciences\ University of Diyala as a Partial Fulfillment of the Requirements for the Degree of Master in Computer Science

By

Sanaa Hammad Dhahi

Supervised By

Assist. Prof. Dr. Jumana Waleed Saleh

2020 A.D. **1442 A.H.**

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

﴿ اللَّهُ نُورُ السَّمَاوَاتِ وَالْأَرْضِ مَثَلُ نُورِهِ كَمِشْكَاةٍ فِيهَا مِصْبَاحٌ
الْمِصْبَاحُ فِي زُجَاجَةٍ الزُّجَاجَةُ كَأَنَّهَا كَوْكَبٌ دُرِّيٌّ يُوقَدُ مِنْ شَجَرَةٍ
مُبَارَكَةٍ زَيْتُونَةٍ لَا شَرْقِيَّةٍ وَلَا غَرْبِيَّةٍ يَكَادُ زَيْتُهَا يُضِيءُ وَلَوْ لَمْ
تَمْسَسْهُ نَارٌ نُورٌ عَلَى نُورٍ يَهْدِي اللَّهُ لِنُورِهِ مَنْ يَشَاءُ وَيَضْرِبُ اللَّهُ
الْأَمْثَالَ لِلنَّاسِ وَاللَّهُ بِكُلِّ شَيْءٍ عَلِيمٌ ﴾

صدق الله العظيم

سورة النّور\ آية35

Supervisors' Certification

We certify that this thesis entitled" ***Sentiment Polarity Identification Framework of Tweets"*** was prepared by ***"Sanaa Hammad Dhahi"*** under our supervisions at the University of Diyala Faculty of Science Department of Computer Science, as a partial fulfillment of the requirements needed to award the degree of Master of Science in Computer Science.

Supervisor :

Signature:

Name: *Assist. Prof. Dr. Jumana Waleed Saleh*

Date: / / 2020

Approved by University of Diyala Faculty of Science Department of Computer Science.

Signature:

Name: Assist. Prof. Dr. Taha M. Hassan

Date: / / 2020

Head Computer Science Department

Dedication

I would like to dedicate this work to:

To my candle that light my life

My Mother and Father.

To My husband Ahmed.

For his unlimited love, his supported,

His patience and Encouragement for me,

About everything in my life.

He is my first and last success.

To My children's Adam and Ayham.

To My Brothers and Sister.

To All My Friends.

Acknowledgements

All my thanks first of all are addressed to Almighty ***Allah***, who has guided my steps towards the path of knowledge and without His help and blessing; this thesis would not have progressed or have seen the light.

My sincere appreciation is expressed to my supervisor ***Asst. Prof. Dr. Jumana Waleed Saleh*** for providing me with ideas, inspiration and continuous support me during the period of my study.

I am extremely grateful to all members of Computer Science Department of Diyala University for their general support.

Finally, I would never have been able to finish my thesis without the help from ***friends***, and support from ***my family*** and ***husband***.

Thank you all!

Sana'a Hammad

Linguistic Certification

This is to certify that this thesis entitled ***“Sentiment Polarity Identification Framework of Tweets "***, prepared by ***“Sanaa Hammad Dhahi”***at the University of Diyala / Department of Computer Science, is reviewed linguistically. Its language was amended to meet the style of the English language.

Signature:

Name:

Date: /   / 2020

Scientific Amendment

*I certify that the thesis entitled **“Sentiment Polarity Identification Framework of Tweets”** presented by **“Sanaa Hammad Dhahi”** has been evaluated scientifically; therefore, it is suitable for debate by examining committee.*

Signature:

Name :

Date : / / **2020**

Abstract

In recent years, Twitter becomes a source of extracting information and knowledge for both individuals and organizations, where opinions and ideas of the users are sharing and exchanging in the form of texts called tweets, about everything that concerns people's daily lives. Therefore, sentiment analysis concerns analyzing people's feelings and classification of these opinions into negative or positive.

In this thesis, an efficient twitter sentiments classification framework has been built to increase the accuracy and decrease the error rate that may be occur in the classification process. A proposed framework consists of three main stages: pre-processing, feature extraction and classification of sentiment stage. In the feature extraction stage a set of (14) features were extracted which includes (13) features statistical were extracted from the tweet itself, and the feature number (14) is a semantic feature was extracted by using Document to Vector technique (Doc2Vec) was computed in order to increase the accuracy of the sentiment classification. In this thesis, two types of a common classifier (Naïve Bayes and Support Vector Machine) were used.

The proposed framework has been tested by using three twitter dataset (Sentiment140, SS-Tweet and STS-Test). The results indicate that the accuracy rate of Naïve Bayes using sentiment140 dataset is 94% and when using SS-Tweet dataset the accuracy rate is 75%, and when using sentiment140 dataset as train and SS-Tweet or STS-Test as test the accuracy rate is 87%,and when Support Vector Machine algorithm is used, the accuracy rate using sentiment140 dataset is 94% and when using SS-Tweet dataset the accuracy rate is 79%, and when using sentiment140 dataset as train and SS-Tweet ,STS-Test as test the accuracy rate is 77% , 84%, respectively.

Table of Contents

List of Abbreviations

Abbreviation	Description
AI	Artificial Intelligence
BOW	Bag of Word
CBOW	Continuous bag of words
CC	Coordinating conjunctions
CM	Confusion Matrix
CSV	comma-separated values
Doc2Vec	Document to vector
ESA	Explicit Semantic Analysis
IC	Information Content
IMDb	Internet Movie Database
IR	Information Retrieval
KNN	K-Nearest Neighbor
LCS	Least Common Subsume
LSA	Latent Semantic Analysis
MAP	Maximum Posteriori
ME	Maximum Entropy
ML	Machine Learning
NB	Naïve Bayes
NGD	Normalized Google Distance
NLP	Natural Language Processing
NLTK	Natural Language Toolkit
NN	Neural Network

Continue of List of Abbreviations

Paragraph2Vec	Paragraph to vector
PMI-IR	Point wise Mutual Information - Information Retrieval
POS	Part Of Speech
PV-DBOW	Distributed Bag of Words version of Paragraph Vector
PV-DM	Distributed Memory Model of Paragraph Vectors
RBF	Radial Basis Function
SA	Sentiment Analysis
SC	Sentiment Classification
SG	Skip-Gram
SGD	Stochastic Gradient Descent
SPTI	Sentiment Polarity of Tweets Identification
SS-Tweet	Sentiment Strength -Tweet
STS	Stanford Twitter Sentiment
SVM	Support Vector Machine
TF-IDF	Term Frequency-Inverse Document Frequency
TM	Text Mining
URL	Uniform Resource Locator

List of Figures

List of Tables

List of Algorithms

Chapter One

General Introduction

Chapter one

General Introduction

1.1 Introduction

Computational linguistics is the science which combines linguistics and artificial intelligence for automatic Natural Language Processing (NLP). In many natural language processing applications such as sentiment analysis (SA) the text similarity measures are used and these measures also use in some domain that is related to text mining. The similarity measure process considered an important task which has high effect in many applications that is dealing with text such as: text summarization, information retrieval, document classification and other applications. The methods used to determine similarity among texts are based on lexical matching is a simple method, it focuses on analyze the share words among texts and detect the degree of similarity among texts based on the words number which appearing in both documents and that match to the lexical style. Lexical methods are easy to implement, but it is poor in reflecting the relationship among words that have a similar meaning, such as words that have the same root or synonymous to each other, co-occurrence words may appear in the longer texts, but it may slightly in short texts or even scarce [1] [2] [3].

There are many applications in natural language processing such as sentiment analysis require specifying the semantic similarity. The concept of semantic similarity can be interpreted as a group of different words which have the similar meaning. Many areas of text mining use the concept of semantic similarity which is an important aspect in natural language processing [4].

1.2 Overview of Social Media

The growth and increase of social media has exploded the publicly accessible text created by users on the internet. This information that created by user can be used to supply insights into people's feelings and also, blogs, online forums and comments on social networking sites like Facebook, Instagram and Twitter can all be considered as a social media. Social media can get millions of people's opinions about a particular topic and it has become an increasing important source of information [5].

On the other side, peoples are more ready and glad to share things about their life’s, their experiences and thoughts with the entire world via social media. People share their events by expressing their opinions and clarifying their comments on things that happen in society. The way for people to share their knowledge and sentiments with community through social media pushes companies to gather extra information about their companies and products and know the extent of their reputation among people and thus make important decisions to continue their business effectively [6].

The increased use of social media has made the SA take an important role in discovering people's opinions through written languages and focusing on detecting the polarity of sentiments if they are (positive, negative or neutral) towards a specific topic. For example, a political party may wish in determining whether or not people support their political process [7].

1.3 Overview of Semantic Similarity

Mainly, the texts can be similar in two lexical and semantic methods. Lexical method uses the idea of traditional matching to calculate the distance among text documents, similarity increases when two text documents contain the same character’s sequence, this method always fail to find true similarity degree while semantic similarity method refer to texts are similar if they contain similar meaning in both, used in same context [8].

The nature of semantic similarity is to simulate the ability of individuals into comparing the texts. Semantic similarity is the measure of the distance across texts or a group of words and the determination of distance depends on the similarity on its meaning or semantic content [9].

Semantic similarity is a method that widely used in the language understanding, it measures how two texts (X, Y) are similar based on the meaning of them. Many types of semantic measures have been suggested to compute the semantic similarity, which range from semantic network-based metrics and distributional similarity metrics models, which are depend on learning from large text sets. Generally, semantic similarity methods can be categorized into two groups: knowledge based methods using lexical databases (manually created) and corpus based methods (using statistical methods) [2].

1.4 Related Works

Many researchers have been done to deal with sentiment analysis, to deal with the problem associated with natural language processing some of researchers use semantic similarity concept in SA which can improve the results of the tweets classification and others employ different techniques to classification positive and negative opinion from text. Here is a review of a number of these works.

- **A. Barhan and A. Shakhomirov (2012) [10]:** They proposed up a model which can extract from Twitter data the sentiment polarity of tweets. The features extracted were words containing emotional symbols and n-gram. The results show that the Support Vector Machine (SVM) performance is better than the Naïve Bayes (NB). SVM in combination with unigram feature extraction is the best performing method, which obtaining a precision of 81% and a recall of 74%.
- **P. Bellot et al. (2013) [11]**: They proposed to use many features for sentiment analysis in micro-blogging such as unigram, domain specific, DBpedia, WordNet

and Sentiwordnet features are using with SemEval 2013 dataset. The experimental result showed that add the above features able to improve the F-measure accuracy 2% with Support vector machine and 4% with Naïve Bayes.

- **G. Gautam and D. Yadav (2014) [12]:** They presented a semantic WordNet synonym analysis approach for SA in twitter dataset. This method depends on examining the semantic synonym similarity between training datasets and words in the testing, when it is found this similarity, it will be replacing the words in the testing dataset with their synonyms in the training dataset. The experimental result showed that the Naive Bayes Classifier (NB) obtained the accuracy 88% which is the best result as compared with other classifiers such as Maximum Entropy (ME) and support vector machine(SVM).
- **D.Zhang et al. (2015) [13]:** They focused on semantic features among words instead of lexical features and two tools are used to classify the Chinese comments texts are Word2Vec and SVM ^{perf}. The results of the proposed method to classification sentiment reached to 90% accuracy.
- **A.Tripathy et al.(2016) [14]:** They attempted to classify the reviews of movies using several classification algorithms like Naive Bayes(NB), Maximum Entropy(ME), Stochastic Gradient Descent (SGD**)** and Support Vector Machine (SVM) then using n-gram feature with these algorithms which are applied on dataset of the IMDB. They noticed that results obtained by applied the above classifiers are 86%, 88%, 85% and 88% respectively.
- **K. Kavitha and Ch. Suneetha (2017) [15]:** They focused on simplifying the rapid detection of sentimental contents. The K-Nearest Neighbor (KNN) and NB classifier are used for sentiment classification of the movie reviews. The experimental results show the NB yielded results 83% and KNN result 72%.

- **O.Araque et al.(2018) [16]:** They proposed an approach of using lexicons of sentiment , that is depend on measure the semantic similarity between lexicon of vocabularies and words in text .A proposal approach consists of a SA which use embedding based representations as well lexicon based semantic similarity as features. The results of the proposed method to classification sentiment reached to 89% accuracy.
- **A.Oussous et al.(2019)[17]:** They proposed an approach to detect the best model for classification the polarity. They showed that unigram is the best for classification the polarity also, they noticed removing the stop words decrease the performance of the classifiers that are used for classifying the sentiments. The experiment results prove that combining between more than one classifier gives the best results with accuracy:86% and presion:89%.

Table (1.1): Related works summarizations

NO.	Year	Author	Data sets	Feature set	Technique	Accuracy
1	2012	A. Barhan and A. Shakhomirov	Twitter messages	n-grams and emoticons	SVM, NB	recall: 74% precision : 81%
2	2013	P. Bellot et al.	SemEval 2013	unigram, Domain specific, DBpedia, WordNet and Senti-features	SVM, NB	adding features has improved the F-measure accuracy 2% with SVM and 4% with NB
3	2014	G. Gautam and D. Yadav	product reviews based on twitter data.	unigram and POS	NB,SVM and ME	88%
4	2015	D. Zhang et al.	Chinese comments on clothing products	Lexicon based and POS	SVM^{perf}	90%

5	2016	A.Tripathy et al.	IMDb	n-gram	NB, ME, SVM and SGD	NB:86% ME:88% SVM:88% SGD:85%
6	2017	K. Kavitha and Ch. Suneetha	movie reviews	POS tagging, unigram, bigram	K-NN, NB	NB: 83% K-NN: 72%
7	2018	O. Araque et. al.	Twitter related: sentiment140 SemEval2014, Vader and STS Gold. Movie reviews: IMDb,PL04 and PL05	Sentiment lexicon and Word embeddings	semantic similarity and lexical metrics	89%
8	2019	A. Oussous et al.	40k Arabic tweets	N-gram	SVM,NB and ME	accuracy:86% presion:89%

1.5 Problem Statement

Literature survey shows that the most studies of the sentiment analysis in twitter is depended on traditional(lexical)features such as part of speech, negation, hashtag, etc., to identify the sentiment polarity. Therefore, the main problem of this thesis is to build a sentiment polarity identification framework using semantic similarity features, in order to help the analysts of data in a huge company to making them able to deal with the general opinions and measure it accurately. For this reason, the analysts of texts (tweets) needs an efficient technique gives an analysis accurately to help them taking the accurate decision about any topic.

1.6 Aim of Thesis

The aim of this thesis is to design and implement a sentiment polarity identification framework of tweets able to accurately classification tweets into positive and negative in twitter by using Naive Bayes and Support Vector Machine algorithms, in order to obtain high accuracy to help the opinion's analysts to prevent the errors while identifying and classifying the sentiment from different datasets and the user also can make direct decisions about any movie, product, service, etc. Without the need for individual reviews, through a combination features provided by the proposed framework developed to achieve this purpose.

1.7 Thesis Outline

This thesis is structured around five chapters, including chapter one, it contains the following chapters:

Chapter 2: Theoretical Background

Presents semantic similarity methods, word embedding, tweet preprocessing, sentiment analysis applications and levels, sentiment analysis classification and accuracy measuring.

Chapter 3: **The Proposed Framework**

Presents the detail of the proposed framework and explains the practical stages of this framework.

Chapter 4: **Experiments and Results**

Includes the experimental results obtained from applying the proposed framework, the evaluation of classifiers techniques on the dataset.

Chapter 5: **Conclusions and Suggestions for Future Works**

Presents conclusions, discussions and suggestions for future works.

Chapter Two

Theoretical Background

Chapter Two
Theoretical Background

2.1 Introduction

The increasing development of humans live and the content generated by them in many websites, social media and online applications such as Twitter, Amazon, etc., has increased the size of the opinion information obtainable from these applications which is available free to any internet user. These opinions can impact directly in many domains, such as commercial transactions, politics, economics and trade [18] [19]. Therefore, concern to opinion mining (sentiment analysis) methods and techniques that are playing an important role in extracting and analyzing people's opinions that are generated automatically has been rise [20].

SA focuses about the classification of opinions or attitudes expressed in texts generated by human. Text can be classified into several categories the most common types being positive category and negative category. SA can be classified into three levels: Document level, as in a movie review which aim is to determine whether an entire document carry positive, negative, or neutral opinion. Sentence level , aim to determine the polarity of each sentence separately assuming that each sentence has only one opinion about one entity and Aspect level which performing more realistic analysis than document and sentence levels based on the supposition that opinion consists of feelings which can be related with a word or small set of words[20].The main indicator of the polarity of sentiment is the words that carry a specific sentiment in the sentence such as love and wonderful can be considered positive words whereas hate and bad can express negative opinion. There are many semantic similarity methods can be used in sentiment analysis. These methods are explained in the next section.

2.2. Semantic Similarity Methods

The concept of semantic similarity has been described previously in section (1.3). In this chapter, we will explain methods of semantic similarity.

2.2.1 Knowledge Based Methods

These types of methods deriving information from semantic networks to determine the degree of similarity between words [22]. In these methods, the input should be a pair of words and whole of these measures return a value of semantic similarity between a pair of word. One of the most popular semantic networks which is used in these measures is WordNet which is produced by "Princeton University", as research project consist of a great lexical database of the English vocabulary and was designed in the form of a graph semantic dictionary containing (words, brief definition and synonyms of the word) [22]. WordNet regulated through a various kinds of semantic relationships into synonyms set called (synset), which is the smallest unit, and every synset consist through a set of words that share a single meaning (synonyms) [23]. The degree of the similarity between any two synsets is determine through computing the space between them in WordNet [24]. It can be concluded that the concepts in higher level are more general. Therefore, the similarities between the concepts of the down level must be more similar than the concepts of higher where it is more specific. Figure (2.1) shows example about the WordNet structure, where vehicle is more general than bicycle while a vehicle is more specified than conveyance [22]. Several measures are suggested for measure the semantic similarity, there are four kinds of these measures: **feature based measures, information content measures, path based measures and hybrid measures** [22].

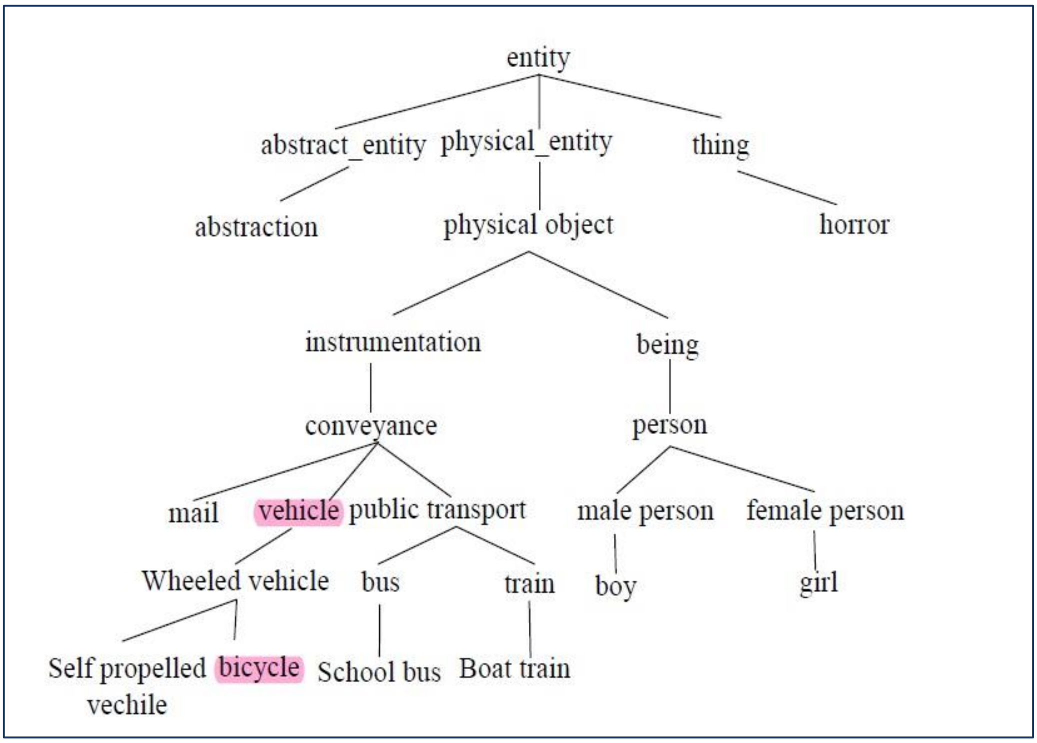


Figure (2.1): WordNet structure [22]

A. Path Based Measures

These kinds of metrics have a group of approaches with one prime concept that semantic similarity score can be computed among words through the path length that connect the words position. The path length express the number of edges that separate one concept or word from other word w_1 to w_2 in the WordNet hierarchy[22].

- **The Shortest Path Measure**

This metric takes the path between two words w_1 and $w_2$and return a degree similar to the senses of two words depend on the shortest path [22], as shown in equation (2.1).

$$sim_{path}(w1, w2) = 2 * \mathrm{depth}_{\mathrm{max}} - len(w1, w2) \qquad (2.1)$$

Where:

$- sim_{path}(w1, w2)$: $semantic\ similarity\ between\ word(w1) and\ word(w2).$

$- len(w1, w2)$: the length of the shortest path from w1 to w2 in WordNet.

$- \mathrm{depth}_{\mathrm{max}}$: the max $depth\ of\ the\ WordNet.$

- **Leacock and Chodorow (lch) Measure**

This metric returns a degree of similarity for two word by take $lenght_path(w_1, w_2)$for two words and the maximum depth , the similarity value between $\boldsymbol{w_1}, \boldsymbol{w_2}$ is $[0, \log(2 * depth_{max} + 1)]$[22],and can be computed as in Eq.(2.2).

$$sim_{lch}(w_1, w_2) = -\ \log_{10} \frac{len(w1, w2)}{2 * \mathrm{depth_{max}}} \qquad (2.2)$$

Where:

$-\ len(w1, w2)$: the length of the shortest path from w1 to w2 in WordNet.

- **Wu & Palmer(wup) Measure**

This measure takes into consideration the depth situation of two specific words in the WordNet and Least Common Subsume (LCS) which is refer to most specific common concept [22], as shown in Eq. (2.3).

$$sim_{wup}(w_1, w_2) = \frac{2 * depth(\mathrm{LCS}(w_1, w_2))}{len(w1, w2)\ + 2 * depth(\mathrm{LCS(w1, w2)})} \qquad (2.3)$$

where:

- $\mathbf{LCS}(\mathbf{w1}, \mathbf{w2})$: **Least Common Subsumer of w1 and w2**. For example, given two strings: *S*(*n*)and T(*m*) with length *n*, *m*. the sequence of characters that appear left to right in both strings:

S = albastruand T = alabaster

In this case, the LCS has length 6 and is the string (albstr)

B. Information Content Measures

These kinds of measures determine the similarity score based on information content (IC) to the concepts. The similarity score is more between concepts when more information's are occurred between them. WordNet is supposed to contain a lot of information's for every concept [22].

- **Resnik’s Measure**

Resnik proposed extracting IC for Least Common Subsume of two words $\boldsymbol{w_1}$ and $\boldsymbol{w_2}$ to determine the degree of similarity[2],as shown in Eq.s(2.4)(2.5)(2.6).

$$sim_{\text{Resnik}}(w_1, w_2) = IC(LCS(w1, w2)) \tag{2.4}$$

$$IC(w) = -\log_{10} P(w) \tag{2.5}$$

$$p(w) = \frac{freq(w)}{M} \tag{2.6}$$

Where

− M: total number of words.

− Freq(w): the frequency of word (w).

−P(w): probability of occurring similar word (w) in a large dataset.

- **Lin Measure**

This measure depend on Resnik’s Measure with the normalization operator, sum of information content of two input concepts [2], as shown in Eq.(2.7).

$$sim_{\text{Lin}}(w_1, w_2) = \frac{2 * \mathrm{IC}(\mathrm{LCS}(\mathrm{w}_1, \mathrm{w}_2))}{\mathrm{IC}(\mathrm{w}_1) + \mathrm{IC}(\mathrm{w}_2)} \tag{2.7}$$

- **Jiang and Conrath Measure**

They calculated the semantic similarity according to the equation (2.8).

$$sim_{\text{JC}}(w_1, w_2) = \frac{1}{\big(\mathrm{IC}\ (\mathrm{w}_1) + \mathrm{IC}\ (\mathrm{w}_2)\big) - \ (\ 2 * \mathrm{IC}\ (\mathrm{LCS}(\mathrm{w1}, \mathrm{w2}\ )\ )} \tag{2.8}$$

C. Feature Based Measure

This measure differs from the previous measures. It depends on describing every concept through a group of words which are separate from taxonomy and it attempts to exploitation the feature of "gloss" in a WordNet to determine the degree of similarity. One classical measure is Tversky model contributes greater to the features between a subclass and its super class in evaluation the similarity [2], which is depend on Eq.(2.9).

$$sim_{\text{Tversky}}(w_1, w_2) = \frac{|V_1 \cap V_2|}{|V_1 \cap V_2| + A|V_1 \cup V_2| + (A-1)|V_1 \cup V_2|} \qquad (2.9)$$

Where A ∈[0, 1] changeable , V1, V2 describe the vectors of concepts $\boldsymbol{w_1}$, $\boldsymbol{w_2}$.

D. Hybrid Measures

It combines the relations of semantic similarity measures which are mentioned earlier. Rodriguez is a suggested method for hybrid similarity measure which have three parts: synonyms sets, neighborhoods and features. The hybrid measure depends on the" is-a" relationship, take information content with path based measure as parameter, besides to the weight operator. Zhou suggested a measure both information content, path based measure and K parameter. If K=1, this mean a similarity measure is based on path based measure, while if K=0, this mean a similarity measure depend on measure of information content, Zhou similarity depends on Eq. (2.10) [22].

$$\text{Sim}_{\text{Zhou}}(\text{w}_{1,}\text{w}_2) = 1\text{-k}\left(\frac{\log(len(w1,w2)+1)}{\log(2*(\text{depth}_{\max}-1)}\right) - (1-k) * (IC(w1)+IC(w2)) - 2 * \frac{IC(LCS(W1,W2))}{2} \qquad \textbf{(2.10)}$$

2.2.2 Corpus Based Methods

Corpus based methods are a group of methods used to measure the semantic similarity depend on words correlation that learn from huge texts called corpora which is a great set of texts written of a particular type or about a particular subject in a public or private field. These methods following the distributional hypothesis, when the surrounding contexts of two words are more similar or frequently appear together, these words supposed to be more similar. The calculation of these methods is depends on the statistics of the distribution of word or word co-occurrences. According to various computational models, there are count based methods and other predictive based methods [25] [21].

2.2.2.1 Count Based Methods

In these methods compute a word co-occurrences and build a word to word matrix. The most common types of these methods are:

A. Latent Semantic Analysis (LSA) Technique

Is the most popular technique of Corpus-Based similarity. LSA assumes that words that are close in meaning will occur in similar pieces of text. A matrix containing word counts per document (rows represent unique words and columns represent each document) is constructed from a large piece of text and a mathematical technique which called singular value decomposition (SVD) is used to reduce the number of columns while preserving the similarity structure among rows. Words are then compared by taking the cosine of the angle between the two vectors formed by any two rows. Cosine similarity represents the angle cosine between two vectors which is computing from the dot product between two vectors and dividing on their magnitudes. The value of angle cosine represents the similarity between the two texts, and higher similarity when the angle is to 1 and equal 0 when no similarity between text [21], as in Eq. (2.11).

$$\cos(\theta) = \frac{A.B}{\|A\|\|B\|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n}(A_i)^2} \times \sqrt{\sum_{i=1}^{n}(B_i)^2}} \qquad (2.11)$$

Where A, B the vector for values words in the text documents or two sentences.

B. Pointwise Mutual Information - Information Retrieval (PMI-IR)

Is one of the corpus based methods that compute the semantic similarity distance between pairs of words. Statistical data is collected from a very big corpora net by AltaVista's search engine after collecting these data will be used later to calculate the probabilities between words. This method depends on whether or not two words occur on the same webpage. Often two words occur close to each other in the webpage, the higher the degree of similarity for PMI-IR [2] [21].

$$PMI - IR(w_1, w_2) = log_{10} \frac{\mathrm{p}(w_1 \& w_2)}{\mathrm{p}(w_1) * \mathrm{p}(w_2)} \tag{2.12}$$

P : The probability of occurring w_1,w_2.

C. Explicit Semantic Analysis (ESA)

Is one of the corpus based measures, in this measure the representations of texts are depend on large knowledge like Wikipedia. To determine the semantic similarity among these texts whose meanings are represented in vectors of concepts by high dimensional space and then apply (TF-IDF) function for every vector depending on one article from Wikipedia. After that to determine the semantic relatedness among vectors the cosine measure is used [26], as shown in Eq. (2.11).

D. Normalized Google Distance (NGD)

Is a measure that derived from the number of results has been returned from Google for a specific group of words. Tend the words which have same or similar meanings to be" near" in the same page whereas words whose have different meanings tend to diverge. NGD between two search words $w1$ and $w2$ is given by Eq.(2.13) [27] .

$$\mathrm{NGD}(\mathrm{W}_1, \mathrm{W}_2) = \frac{\max[\log_{10} \mathrm{f}(\mathrm{W}_1), \log_{10} \mathrm{f}(\mathrm{W}_2)] - \log_{10} \mathrm{f}(\mathrm{W}_1, \mathrm{W}_2)}{\log_{10} \mathrm{M} - \min[\log \mathrm{f}(\mathrm{W}_1), \log_{10} \mathrm{f}(\mathrm{W}_2)]} \tag{2.13}$$

$M: no. of\ all\ returned\ pages.$

f(W_1), f(W_2) : The number of pages returned of W_1 , W_2.

$f(W_1, W_2)$: The number of pages returned that contain both words.

2.2.2.2 Predictive Based Methods (distributed representation)

These methods are used to learn the dense vectors directly by anticipating a word from its surrounding context, such of these methods is word embedding tool Word2Vec for learning dense vector representation to the word. Word2Vec has a good performance in many applications [28].

2.2.3 Word Embedding

Word embedding considered unsupervised training and can be utilized with a different text (data set) that are not labeled [21]. It is an effective research domain that try to find the best representation of words in the document set (corpus), it is the most popular representation for document vocabulary and the main idea behind it, is to catch as much contextual, semantic, and syntactic information as potential from the corpus (set of documents) [29]. Distributional vectors follow the distributional hypothesis, in which words that have the similar meanings tend to appear in a similar context. Consequently, these vectors attempt to catch the properties of the word's neighbors. The main characteristic of these vectors is that they can catch the similarity among words, measure similarity among these vectors by using measures like cosine similarity. Learned word vectors can carry the syntactic and semantic information. So, these embedding proved effective in capturing similarity of context and because of their smaller dimensions, they are fast and effective in calculating basic natural language processing tasks [29].

One of the most common tools to learn word embedding is Word2Vec, this model has been proposed by Mikolov et al. [28, 30]. Word embedding can apply in various tasks such as in sentiment analysis [31]. It can be used as a features, in the similar way that Bag of Word features are exploited for textual representation in SA [28].

A- Word to Vector (W2Vec) Model

Is one of the most popular technique to learn word embedding using shallow neural network architecture to train word vector, which consist of three layers are input, hidden and output layer. W2Vec is a neural network based implementation that learns distributed vector representations of words based on a continuous bag of words (CBOW) and skip-gram architecture (SG) these methods shows in figure (2.2) [32]. Continuous bag of words calculates the probability for the goal word specified by looking for the context words surrounding it using a window of size N while the SG model does the exact reverse of the continuous bag of words model, by predicting the surrounding context words by looking at the central goal word. Context words supposed to be existing symmetrically with the goal word into a distance that equal to the size of the window in both directions [29].

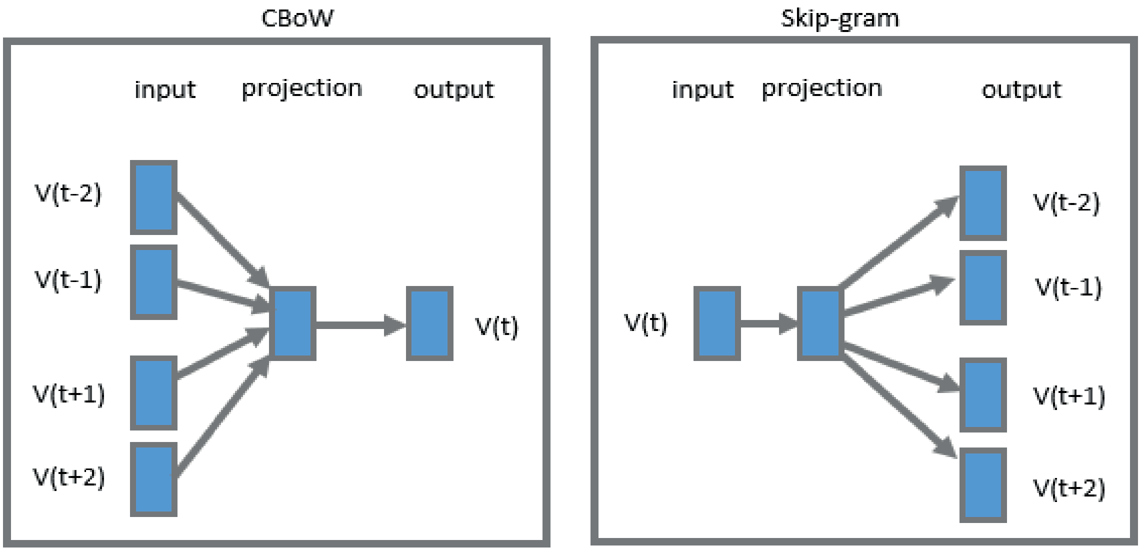


Figure (2. 2) W2Vec models (CBOW and SG) [32]

This model does not require manual labels to create a meaningful representation between words. W2Vec builds a model that can be used to extract word vectors for each word in a document. The output probabilities will depend on how likely each word is found in the vocabulary near the input word. For example, if deep learning user gives the input word “European" to the trained network, the output probabilities will be much higher for words such as “Union” with “Germany” than for irrelevant words such as “Orange” and “Tiger.”

The NN is trained to do this by feeding it word pairs found in the training documents. As in figure (2.3) shows some of the training samples (word pairs) that is taken from the sentence “The quick brown fox jumps over the lazy dog.” a small window size of 2 has been used just for example. The word highlighted in blue is the input word [33] [34].

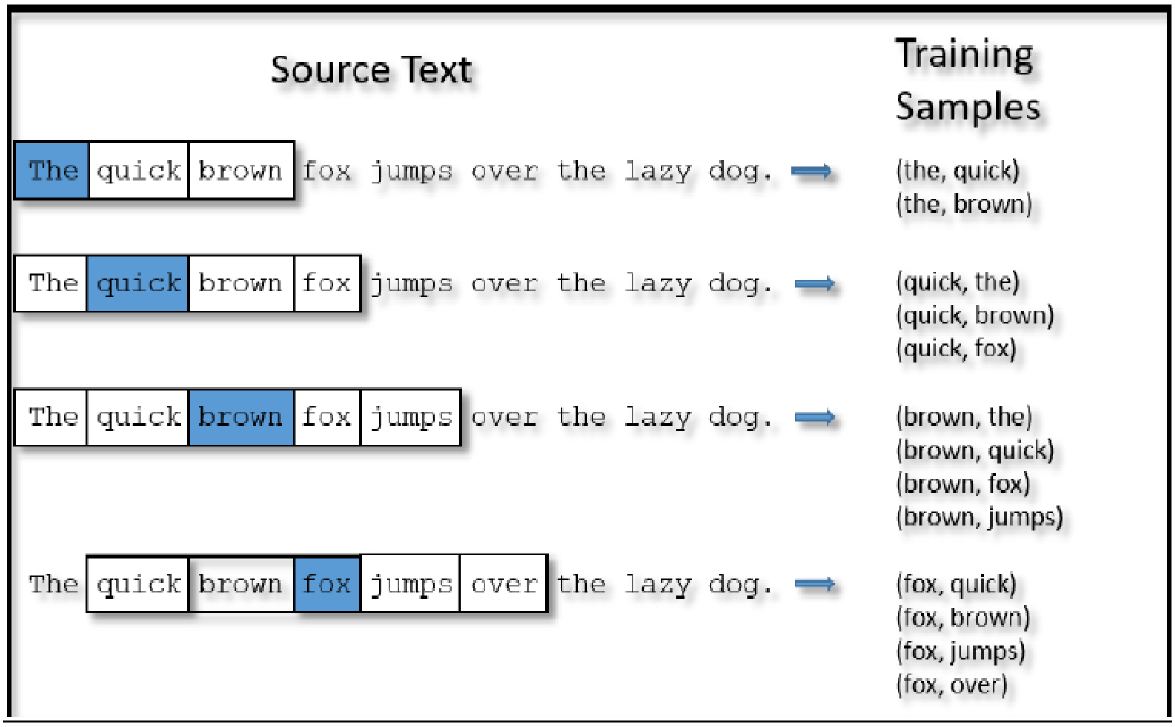


Figure (2. 3): An example of Word2vec implementing model [33]

The network will learn the statistics from how often each pair appears. Therefore, for example, the network will likely get training samples ("brown", "fox") more than ("lazy", "fox"). After completing the training, if you give the network the word "brown" as an entry, it will result in a much higher probability output for "fox" or "quick" than "dog" [33].

2.3 Text Preprocessing (Tweet Preprocessing)

Text pre-processing involves the use of various techniques to convert raw text to a distinct sequence of language components that have standards and symbols [32]. For most machine learning based classification problems, it is necessary to pre-process the text. If the natural language processing system not process the texts appropriately, it will have finished with unwanted and unrelated result from natural language processing implementation. Text preprocessing is to clean up the text, which helps increase the accuracy of classifiers. The most common methods for text preprocessing are (tokenization, tagging, stemming and lemmatization). In addition to these mentioned techniques, natural language processing system need some other process, like handling with text that contains spelling errors, removing stop-words and processing unrelated information according to the issue that will be solved [32]. The preprocessing steps have been explained in details as following:

2.3.1 Text Tokenization

It is the first step from preprocessing stage for many applications in text mining and NLP. Tokenization is defining as the operation of dividing the text to smaller parts known as tokens. Depending on the boundaries of a word that depend on white spaces and punctuation marks as delimiters between words (".", " ", ",", ";" …..)[37]. According to the mentioned, there are two types of tokenization:

1. Sentence tokenization: Is the operation of dividing the text group into sentences. The goal of this operation is to divide the text into meaning sentences. This process is implemented by searching for delimiters among sentences, for example the period (.) or the letter (\ n) for a new line [32] [39].

2. Word tokenization: It defined as the operation of fraction the sentence to the component words known tokens. These tokens can be used to clean (normalize) operation such as stemming and lemmatization [40].

2.3.2 Text Normalization

It is defined as a set of operations which implementing for cleaning the text from noisy and make it into standardized form, that treatable processing by the natural language processing analysis system [41]. The following techniques are used for text normalization (text cleaning):

1- **Lowercasing:** One of the most popular preprocessing techniques is writing all words in lower case. By doing this, many words are integrated and also the problem dimensionality is reducing [42].

2- **Substituting**: Multiple spaces with a single space.

3- **Replace Slang and Abbreviations:** Usually the users of social media write comment informally way and their texts contain many abbreviations and colloquial languages. In order for these words to be interpreted correctly, they must be replaced to refer their correct meaning. Some examples of these words are "4u", "b4","ilu" and "gr8" which respectively mean and replace "for you ", "before","i love you" and "great".

4- **Expanding Contractions**: It is a process of replacing acronyms of negation to a standards tokens. The best way to handle with these words is to restore them to their origin such as " can't ", " won't " to "cannot", "will not" respectively [43].

5- **Correcting Words:** It is a critical step in normalization that do to correct incorrect words. It contains a letter that is repeated more than twice and these errors involve words with repeated letters, such as someone, who write in the tweet finally word as "finallyyyyyy" therefore it must be corrected to be properly analyzed by the natural language processing system [44]. Another type is spelling errors that may occur because human error. The method that is used to correct the words through finding a word nearer to the corpus and its frequency in this corpus [45].

6- **Remove Numbers:** It is a popular method for removing numbers from the text, because numbers do not include any opinion or sentiment. However, some of researcher’s dispute that keeping the numbers may be improve the classification effectiveness [46].

7- Removing Punctuation Marks and Special Characters: This process includes removing the special characters or some punctuation marks that may be needless. The characters selection are depend on natural language processing applications. [39] [41]. The assumption with the stop words, it also extends to punctuation marks and special characters. Where excluded the exclamation mark "!" and question mark "؟" and @ at the start of words and URLs from removing.

8- Remove Stop Words: Although, stop words have a role in complete the meaning of a sentence but they are leads to a low performance of classifiers. Based on a specific list of words, the tokens are removed from tweet text. Multiple lists exist in the literature such as in [47]. These words can be indicators that reflect a certain type of user’s feelings towards a specific topic. For example, question, negation, and conjunctions words except the negation tokens (list of negation words), some of conjunctions words such as (but, although) and question words. excluded stop words belong to negation words, some of conjunctions words such as (but, although) and question words [47].

9- Part of Speech (POS) Tagging: It is a process of labeling (tagging) each word with its correct part of speech. It’s a tool can tag the parts of speech in a tweet.

10- Replace Negations with Antonyms: Negations are words which are affecting the sentiment orientation in a sentence, such of these words are involve: not, no, never, etc. Negation handling is an approach used to search in each sentence for determining the negation words, if the negation is found we check to see if the next word after negation contains an antonym. If yes, then we replace the two words (negation and the next after the negation) with antonym, this is done by using WordNet. For example, "not good" will be replaced with "bad" [23].

2.3.3 Stemming

Is an operation of removing any affixes (prefixes that added to the beginning of the word, infixes that added to the middle of the word, and suffixes that added to the ending of the word) from the words to reduce these words to their roots (the original word without affixes), such as "studying" can be stemmed by removing the affixe (ing) from the word to obtain the original word "study"[48][49]. The stemming algorithm can be classified in two types of category.

1. Rule Based Approach: In this approach the "stemming" is implemented through a set of principles and rules for converting the word to its derived root (stem).

2. Statistical Approach: This approach is depend on removing affixes from word after performing some of statistical steps [50]. And this approach works to determine the closer word to the target word from the corpus [51] [52].

2.3.4 Lemmatization

Lemmatization is very similar to stemming, it works to define the basic form of the word. The difference between it and stemming is that, in the stemming the stem word not always represent the origin word (root), it only removes the suffixes from the word while in "lemmatization" it operates to remove the affixes from the word only if the lemma is found in the WordNet, such as the word "better" has "good" as its lemma [48]. The basic function of the stemming and lemmatization algorithms are similar, both of them attempt to deal with the diverse words through conversion them to the stem and lemma, but between both concepts subtle difference [52].

Stemming algorithm is faster and easier than lemmatization which consider more complex operation, therefore it is difficult to implement, but lemmatizing has a better performance than stemming [53].

2.4 Features Extraction in Textual Data

Features are a unique measure of every point or data in the data set. Generally, features are being digital and it used to improve the performance of ML algorithms. The process of extraction, selection and normalization for these features is known as feature extraction or feature engineering process [30]. This process can be reducing the amount of redundant data by dimensionality reduction that is through deleting uncorrelated or unnecessary features and it also can improve the accuracy of ML algorithms and shorten the time [54].

2.4.1 Bag of Word (BOW) Model

It is a simplest and earliest model for feature extraction in NLP. This model operates to transform each document or text into a vector that contains the frequency of each word exists in the document. This model suffers from a drawback that it may give high importance for a word that frequency occurs in all document and ignore other may be unique for such a document to be distinguished [55]. The features of text are represented by using the term vector the model which is defined as an algebraic model for converting text document to a numeric vector [32].

$\mathrm{D}: \{\mathrm{Wd1}, \mathrm{Wd2}, \ldots, \mathrm{Wdn}\}$

Where

$- Wd$ refer to the weight of a word N in the document D .

2.4.2 Emotion

Most sentences consist of two parts: “text” and “symbols”. Emotion is very popular in sentiment analysis because it is very useful in determining sentiments for the sentence. For example, a sentence " school begin after 5 day". This sentence clarifies the fact that the school will start after 5 days. It is a neutral phrase because it does not consist of positive feelings or negative feelings. Suppose what happens if we attach an emotion to this phrase.

“school begins after 5 days :)”, the sentiment of this sentence is clearly positive. It appears that the person behind this sentiment is happy with the fact that his school is about to begin. It can be assumed that the person likes going to school.

Now, take the same phrase written with an another emoticon symbol, for example "school begins after 5 days :(". The sentiment of this phrase is clearly negative. This tweet indicates that the person behind this phrase does not wish school to begins. This example shows that adding an emoji at the end of a sentence greatly changes the feeling of a sentence. Thus, emojis can be very helpful in determining sentiment of the sentence (tweet) [56].

2.4.3 User Mention

User mention is a feature that can be used in the tweet to reflect the user's behavior, so when the user wishes to indicate to other user he can write name of user begin with @ icon. This is called as user mention and it is also represented as @username [57].

2.4.4 Uniform Resource Locator(URL)

URL It indicates the position of a resources on the web such as a street address which indicate where a person lives, it is also can be used in the tweet to reflect the user's behavior. Many tweets share a link to refer something [57].

2.4.5 Hashtag

Hashtag is a word begins with the symbol "#". It indicates a word about the content of the text or refers to the topic of the tweet [57].

2.4.6 Part of Speech Tagging (POS Tagging)

It is the process where each word in the text (corpus) are tagged such as adjectives, nouns, adverbs, etc. It has been found that some of these parts more often express polarity [58].

2.4.7 Negations

Words of negation are the words that effect on the direction of sentiment for the other words in the sentence. Such as these words are including (not, no, never) and other words. Negation processing is an automatic method for determining a scope of negation and reversing the polarity of words that are influenced by negation [44]. The word negation, such as "not" reflects the value of the emotional word. For example, "not beautiful" is like to say "ugly" [58].

2.4.8 Punctuation Marks

Consider one of the main features in this thesis. Many studies have shown that punctuation has a lot of influence in text classification, especially in the area of sentiment analysis, two features related to punctuation (exclamation mark and question _mark) are used. These two features could be useful for some cases included them in our sentiment classification system by computing the number of exclamation marks and number of question marks in the tweet [59].

2.4.9 Coordinating Conjunctions(CC)

Conjunctions tools are links that are usually come in the middle of a sentence, and their function is to link words or phrases. The major objective of applying conjunction rules is to elicit the exact sense or expression from a particular sentence. In general, a sentence expresses only one opinion direction unless there is some of conjunctions tools that changes the direction of the sentence, such of these tools: and, or, so, but, etc. For example, " **the appearance of the car is not beautiful, but it is very practical”.** In this case, the first phrase can be clipped and the second phrase is used to define sentiment [58].

2.4.10 Document to Vector Model (Doc2Vec Model)

This model is proposed by Le and Mikolov in 2014. Doc2Vec performs the same operations that Word2Vec does with phrases or paragraphs. In some resources refers to Doc2Vec with name Paragraph2Vec. Doc2Vec is adjusted version of Word2Vec model [60]. The only alteration is made to the Word2Vec model is to add the document ID, as shown in Figure (2.4). Doc2Vec is a simple extension of word2vec to extend the learning of embedding from words to word sequences that learn continuously distributed vector representations for chunks of texts. Texts can be have changing length, ranging from the sentence to the document. Paragraph vector name is an affirmation of the fact that method can be applied to changeable-length parts of text, anything from the phrase or the sentence to a big document. After training, these paragraph vectors can be used as features and these features can be feed directly into ML algorithms such as Naive Bayes, SVM and other [61].

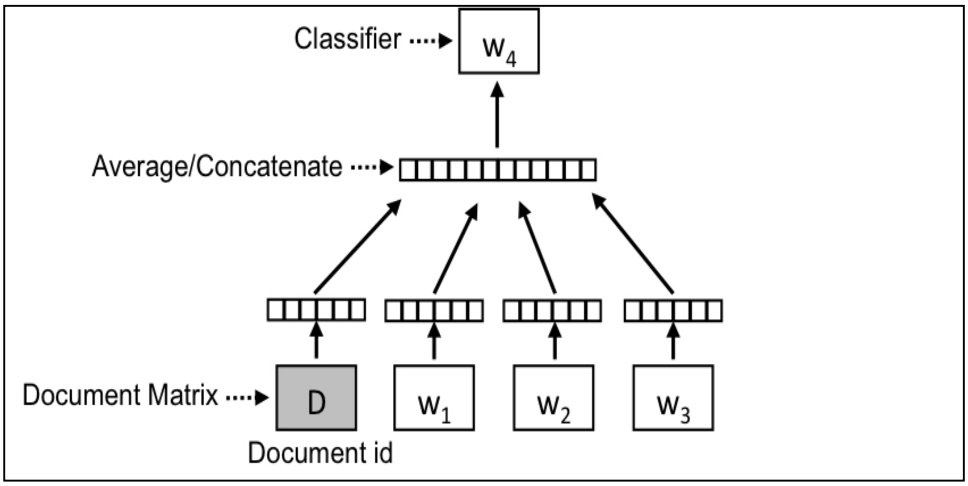


Figure (2.4) Doc2Vec model [61].

Doc2Vec comes in two methods shows in figure (2.5): These methods are Distributed Memory Model of Paragraph Vectors (PV-DM) and the Distributed Bag of Word's version of Paragraph Vector (PV-DBOW). PV-DM is basically the same as CBOW except that a paragraph vector is additionally averaged or concatenated along with the context and that whole thing is used to predict the next word. In the PV-DBOW model a paragraph vector alone is used/trained to predict the words in the paragraph [61].

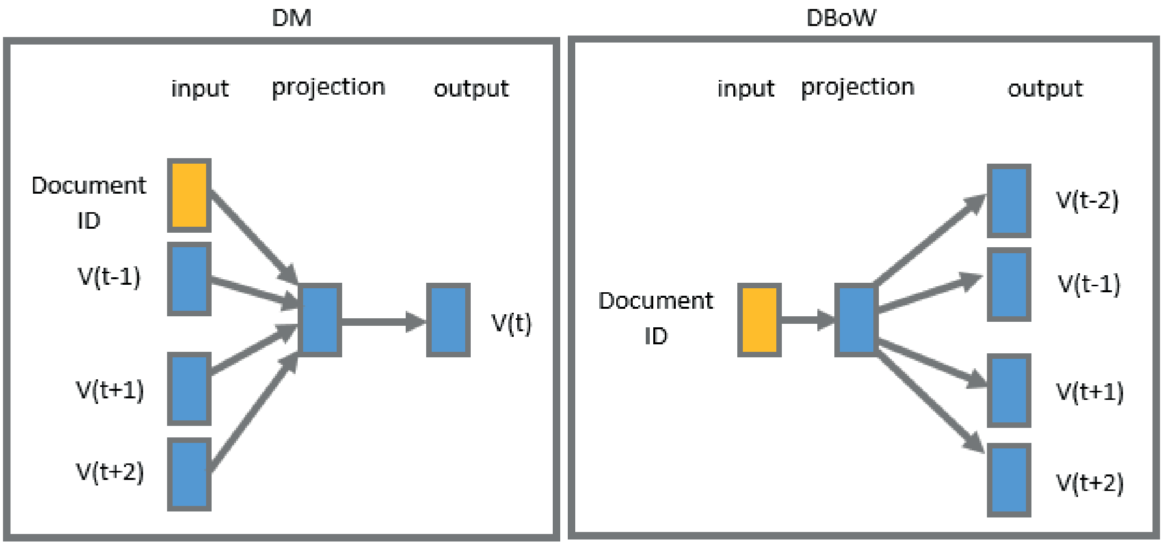


Figure (2.5) Doc2Vec (PV-DM and PV- DBOW) methods [61].

Doc2Vec operates on the logic that the meaning of a word also depends on the document that it occurs in. The vectors generated by Doc2Vec can be used for finding similarities between documents [62].

2.5 Sentiment Analysis

Opinions are crucial to an almost human decision and play a vital role in influencing a person behaviors. For this reason, when a person needs to make a decision, this person often search the opinions of other people related to any topic [63]. However, the massive amount of opinions on the web makes it hard to group useful information quickly in addition to that reading all reviews consumes much time so the sentiment analysis is a concept used to indicate to a general domain of the study which is define as "analyzes people opinions, sentiments, evaluations and emotions towards particular topics like products, services, persons, issues, events topics and their attributes"[64] [65]. The aim of the SA is to develop automatic tools can extract subjective information from the text in natural languages, such as opinions and sentiments.

2.6 Levels of Sentiment Analysis

SA is investigated generally at three levels of texts, this illustrated in figure (2.6) and these levels are:

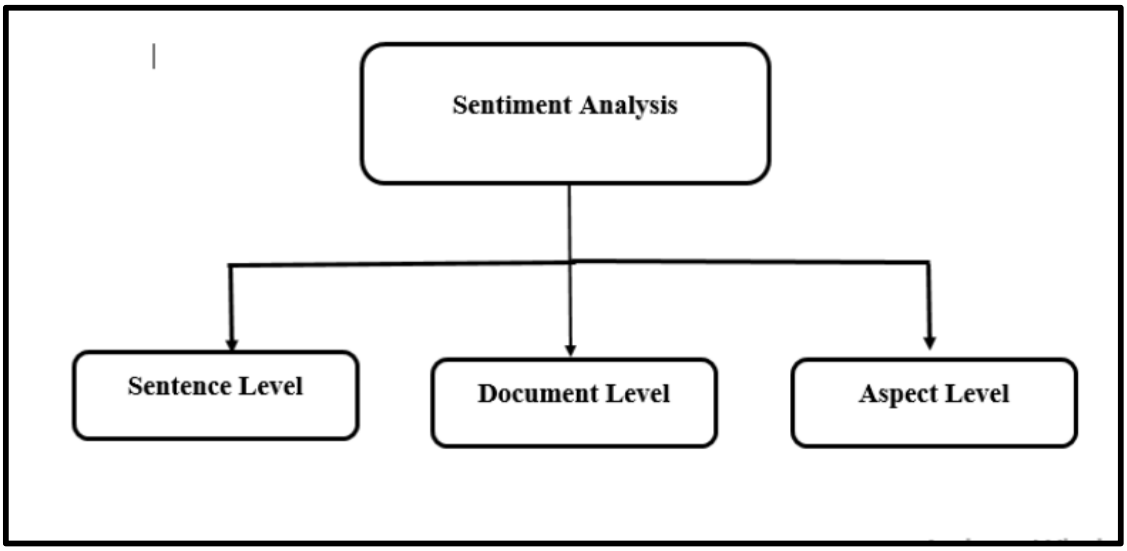


Figure (2. 6): Levels of sentiment analysis [32]

1. Document Level: The goal is to define whether a whole document holds positive, negative, or neutral that opinion may contain only one opinion containing an entity (like a single product) [66] [32].

2. Sentence Level: SA in this level aims to specify the polarity of each sentence independently. The assumption is that each sentence holds only one opinion about one entity.

3. Entity and Aspect–Level: Perform a more realistic analysis than document and sentence level. This level relies on the assumption that opinion consists of sentiment and an aim. An aspect is usually expressed by a word or a small set of words, which are names of the aspects of the entity (nouns) [65]. For example, the phrase, "LG G4 has a wonderful camera, but a bad battery life", evaluates LG phone into two aspects, camera, and battery life, the sentiment about the camera is positive, but the sentiment on the battery is negative [64]. The analysis at aspect-level is more challenging than SA in document and sentence levels [64] [32].

2.7 Application of SA

There are many applications of sentiment analysis which shows as following [67]:

1.Applications in Web Sites: Internet contains a big group of reviews and comments about roughly everything, this involves reviews of products and movie, the comments about political issues, comments on services provided by companies and others. Therefore, there is a necessity for existence a SA system that be able to extract sentiments about a specific product or service. This will serve the needs and requirements of both clients and sellers.

2. Applications in Technology: The sentiment analysis system can be useful in a recommendation system because these systems will not advise the items that receive more negative comments or items that have few ratings.

3. Applications in Business Intelligence: Nowadays people tend to look of product reviews available online before purchasing them. For many companies, online opinions determine the success or fail of their products. Hence, SA plays a significant role in business, also companies desire to extract emotions from online reviews in order to enhance their products and thus improve their repute and assist to contentment of clients.

4. Applications in Many Domains: Many recent research in various fields such as medicine, politics, economics, sociology, sports and other are benefit from SA that appear orientation in human sentiments particularly on social media networks.

5. Applications in Smart Homes: It is supposing that the smart homes be technology of the future. In the future the whole houses will become connected to a network and people will have the ability to controlling whatever part of the house using a tablet device depend on the current sentiments or emotions of the person. For example, the house can change its weather (air) for creating a quiet environment.

2.8 Sentiment Classification

Sentiment Classification is a task to extraction and classification the text whose objective to classify according to a polarity of the opinion it contains, e.g. positive or negative, good or bad, like or dislike. Sentiment classification contains multiple techniques, and it is classified into three main techniques, namely machine learning approach, hybrid techniques approach, and lexicon-based approach [68].

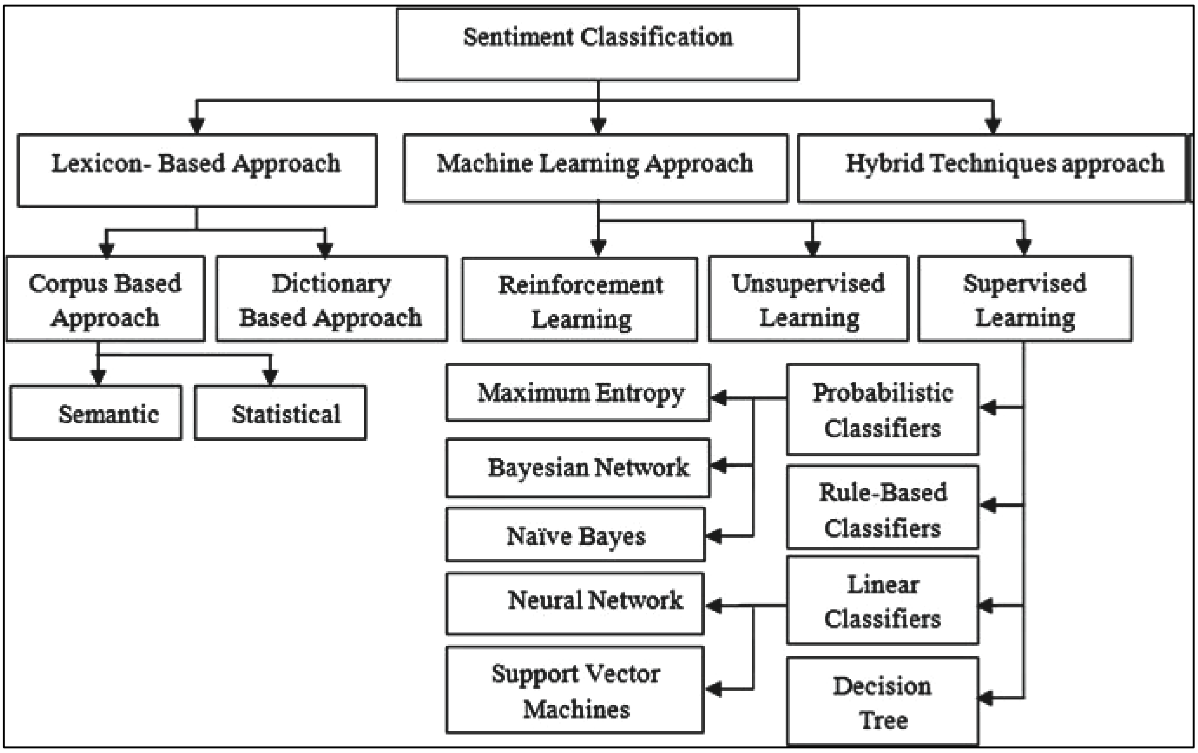


Figure (2.7) Sentiment classification techniques [68]

In the above figure (2.7) illustrated the most popular techniques of sentiment classification. There is a brief clarification of these approaches in the later sections.

2.8.1 Lexicon -Based Approach

Multiple words are used to classify sentiment and use positive words for the desired things, while using negative words for undesired things. So, lexicon-based approach relies mainly on finding opinion lexicon, which is used for text analysis. There are two methods according to lexicon-based approach. The first one is corpus-based approach, and the second one is dictionary-based approach [67].

A- **Corpus-Based Approach**: The corpus-based approach starts with a seed list of opinion words and then finds other ideas from the words in a large corpus to get opinions from certain directions. In another meaning, most methods rely on grammatical patterns or that occur together with the seed list of opinion words to find other words from a large corpus. To implement corpus-based approach, we use two different approaches: statistical approach and semantic approach as illustrated in the following.

1- **Statistical approach**: It is used in many applications that have a relation in the field of SA. The famous of them is the one that can detect the manipulation of the review by conducting a statistical test of randomization which is called runs test.

2- **Semantic approach:** It gives values to sentiments while relies on more than principle to calculate the affinity and similarity of different words. The basis of this principle is to support the Sentiment value in the words and words close from thesaurus such as WordNet.

B- **Dictionary-Based Approach:** This method depends on the idiom use (seeds) that are usually collected and annotated manually. This set growing by researching synonym and antonym of a dictionary. Such as the dictionary of WordNet, which is used to develope a SentiWordNet dictionary that cannot handle with domain specified orientations.

2.8.2 Machine Learning Approach

It is used to solve the problems related to text classification that contain syntactic or linguistic features. Machine learning approach divided into reinforcement learning, unsupervised learning and supervised learning [68].

2.8.2.1 Reinforcement Learning Approach

Its entirety indicates how to make an optimal decision an important technique that differs relatively from its counterpart unsupervised learning. This technique is highly concerned with improving the efficiency of text classification to show that the reinforcement learning technique is important and prominent.

2.8.2.2 Unsupervised Learning Approach

It is a unique type of machine learning algorithm and is used in most cases to draw and diverse inferences of data; these groups of data consist of input data without any labeled responses. It is used when it is impossible to obtain labeled training documents.

2.8.2.3 Supervised Learning Approach

It is a type of machine learning approach that uses a data set called training data set to make predictions. These data set contain input data as well as response values. In supervised learning methods, it makes use of a large number of assorted training documents. In this thesis two types of classifiers are used that are widely used in the field of sentiment analysis like probabilistic classifier such as Naïve Bayes (NB) and linear classifier such as Support Vector Machine (SVM).

A. NB classifier

Naive Bayes Classifier is a probabilistic classifier based on Bayes' theorem. Bayes' theorem describes the relation between conditional probabilities of a hypothesis(y) and observations (x) as given in equation (2.14) [69] .

$$\mathrm{P}(\mathrm{Y}|X) = \frac{\mathrm{P}(\mathrm{X}|Y)\ \mathrm{P}(\mathrm{Y})}{\mathrm{P}(\mathrm{X})} \qquad (2.14)$$

where:

- P (y) = prior probability of hypothesis y
- P (x) = prior probability of observations x
- P (y ǀ x) = probability of hypothesis y given x (posterior probability)

- P(x |y) = probability of x given hypothesis y (likelihood or conditional probability).

Typically, the Maximum A Posteriori (MAP) hypothesis is used to assign to the class (y) having maximum P (Y|X). It is expressed as shown in equation (2.15) [70].

$$\mathbf{y_{MAP}} \equiv \arg\max_{y \in Y} \equiv \arg\max_{y \in Y} P(x \mid y)\, P(y) \qquad (2.15)$$

- Where Y is the set of the hypotheses.

NB Classifier assumes that the conditional probability of observations (x) given hypothesis (y) equals to the prediction of conditional probabilities of each observation given the hypothesis according to equation (2.16).

$$P(x1,x2,x3,\ldots..,x_n \mid y_j)=\prod_i p(\,x_i \mid y_j\,) \qquad (2.16)$$

By substitution of p(x1,x2,x3,…..,x_n |y_j) by $\prod_i p(x_i \mid y_j)$ in equation (2.15) , NB classifier is given by equation (2.17)

$$y_{NB=}\arg\max_{yj \in Y} p(y_j) \prod_i p(\,x_i \mid y_j\,) \qquad (2.17)$$

NB classifier is a supervised learning algorithm which means it needs to be trained before being able to do classification. Therefore, it must have a training set. The training set contains a number of observations and the classes in which they are classified. The aim of a NB classifier is to classify an unseen sequence of parameter values into one of the classes in training set [71]. The general concept of the process illustrated in figure (2.8).

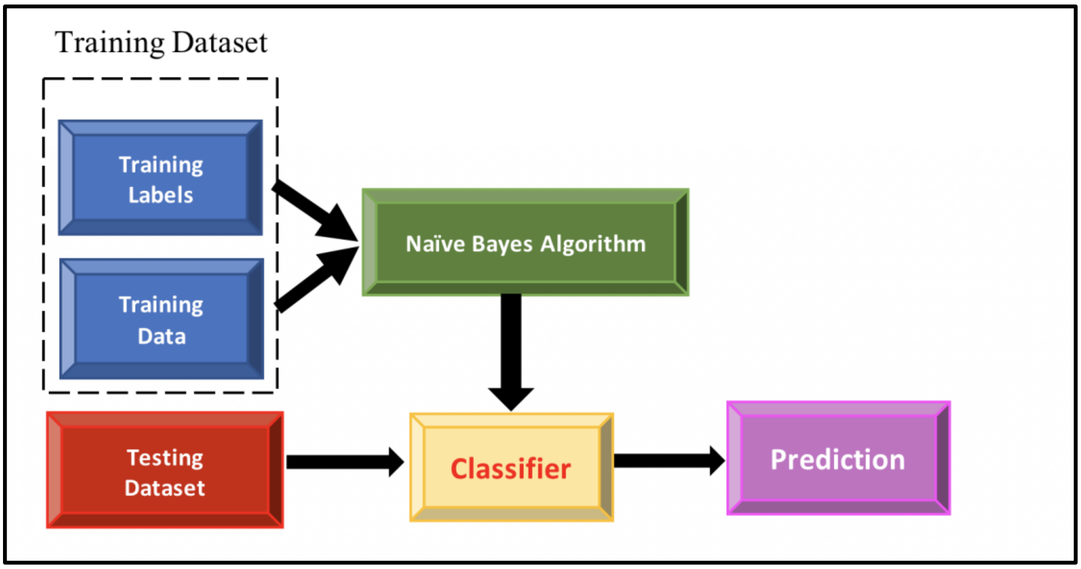


Figure (2.8) NB algorithm process [71]

1- Gaussian NB Classifier

In this classifier [72]. The value of the numeric features is usually distributed. This distribution is represented in relation to the mean (μ) and the standard deviation (σ) that will help in calculating the probability of the values observed using the estimations. The probability of the features has been computed as follows:

$$\text{Prob}\ (\mathbf{x_i}|\mathbf{c}) = \frac{\mathbf{1}}{\sqrt{\mathbf{2}\pi\sigma_c^2}} \exp\left(-\frac{(\mathbf{x_i}-\mu_c)^2}{\mathbf{2}\sigma_c^2}\right) \qquad (2.18)$$

Where:

$$\mu = \frac{1}{n} \sum_{i=1}^{n} x_i$$

$$\sigma = \left[\frac{1}{n-1} \quad \left(\sum_{i=1}^{n} (x_i - \mu)^2\right)\right]$$

$\boldsymbol{n}$= refer to number of instances.

$\boldsymbol{x_i}$ = refer to features .

$\mathbf{c}$ = refer to class .

2- Bernoulli NB Classifier

This classifier supposed that the features are binary and demand only two values. Equation of Bernoulli distribution is described as follow [73].

$$p(x) = p^x(1\text{-P})^{1\text{-x}} \tag{2.19}$$

Where x was the Bernoulli distribution, with a value ranging between 0 and 1. If it was 0, failure occurred while it was successful if it was 1. Based on the below equation [74].

$$P(x = 1) = P^1(1 - P)^{1-1} = P \tag{2.20}$$

$$P(x = 0) = P^0(1 - P)^{1-0} = (1 - P) \tag{2.21}$$

The likelihood of the word not occuring in the class document was $(1 - \mathrm{P}\ (\mathrm{x}_{\mathrm{i}}|\mathrm{c}))$ where x was a word in the document [75]. As in equation below:

$$\mathrm{P}\ (\mathrm{x}_{\mathrm{i}}|\mathrm{c}) = \mathrm{P}\ (\mathrm{x}_{\mathrm{i}}|\mathrm{c})\mathrm{b}_{\mathrm{i}} + (1 - b_i)(1 - \mathrm{P}\ (\mathrm{x}_{\mathrm{i}}|\mathrm{c})) \tag{2.22}$$

This equation can be used for all the words. If the word X_{i} was found in the document, then $b_i = 1$ and the likelihood was $\mathrm{P}\ (\mathrm{x}_{\mathrm{i}}|\mathrm{c})$. If the word $\mathrm{X}_{\mathrm{i}}$was not found , then $b_i = 0$ and the probability was $(1 - \mathrm{P}\ (\mathrm{x}_{\mathrm{i}}|\mathrm{c}))$. The Algorithm (2.1) explains the general NB algorithm [76].

ALGORITHM 2.1: Naïve Bayes
Input: T = {(x_i, y_i) ׀ x_i∈ n ,y_i ∈ m , i∈ {1,2,...,*N*}} set of *N* training samples and class. Z = {z_i ׀ z_i ∈ m ,i∈ {1,2,...,*t*}} the set of *t* test samples;
Output: Y = {y_i|y_i∈{1, -1}} set of predicted labels for the test samples in Z.

```
// Initialization
Y= ∅
Read the training T
Calculate the parameter for predict class
//Computation Learning Algorithms
For each $z_i \in Z$ do
    $P_c$=calculate the probability of Tweet class
    $P(x_i|c)$= calculate the likelihood for each class depend on model (Gaussian, Bernoulli)
    $y = P_c \prod_{i=1}^{N} P(x_i|c)$
    $Y = Y \cup y$
End for
```

B. Support Vector Machine(SVM)

It is one of the methods of supervised classification algorithm, where a set of inputs are given with their labels and these inputs represented by attributes vector. It analyzes the data based on separating different classes by finding a hyperplane, which can best maximize the margin among different classes, hyper planes as a fundamental concept for decision boundaries and for separating the different classes [77].

The performance of SVM falls short when it comes to non-linearly separable data, the solution to this problem is to use kernel functions to shift data into high dimension space, with this move, the data can be separated linearly. The main idea of this classifier is to specify an appropriate kernel function, as well as adjusting of kernel parameters. In terms of computations, searching for the most appropriate decision plane is an optimization issue. An appropriate decision plane would serve to facilitate the generation of linear decisions by the kernel function, through a nonlinear transformation as shown in equation (2.23) .The Algorithm (2.2) explains the general SVM algorithm [77].

$$f(x) = w^T x_i + b$$
$$f(x) = \sum_{i=0}^{N} \lambda_i y_i \left(w_i^T x + b \right) \tag{2.23}$$

w^T: represent the weight of vector.

f (x): represent the features sets of both classes.

λi: represent the dual function that was returned after training.

x: represent a training dataset.

y: represent the classes (output).

b (bias): represent omega 0.

In figure (2.9) the two planes which are parallel to classifier that passes through several points are called "bounding planes" and the points on these planes are called "Support Vectors”. Finally, the distance between bounding planes is known as "margin"[78]. SVM algorithm is classified into two types: linear and non-linear as will explain in the following sections.

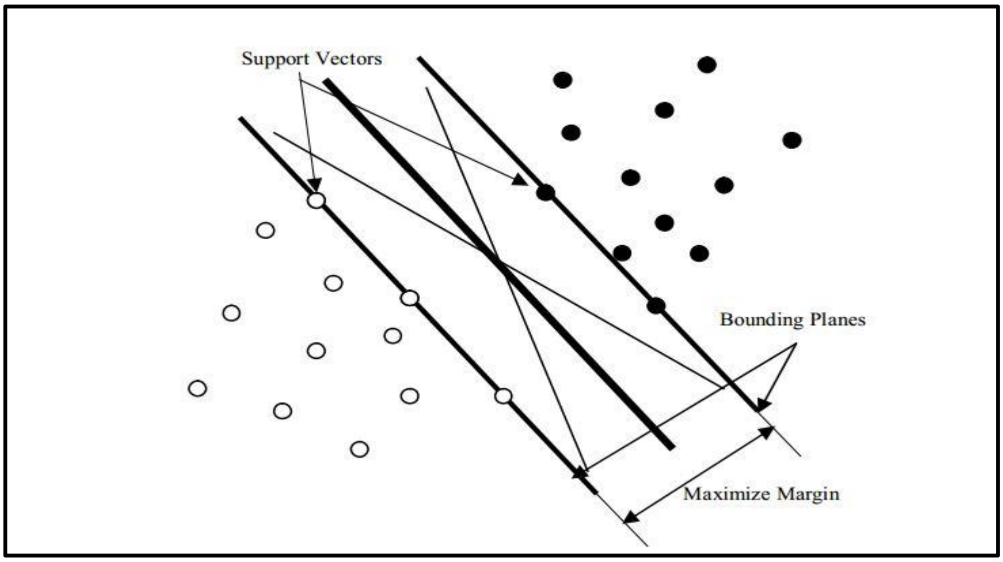


Figure (2.9) SVM Hyperplanes between two classes [78]

1- Linear SVM

SVM is called linear or non-linear this is depending on the hyperplane if it is linear, then SVM will be known as linear. For example, if k represents the training pairs (x_i,y_i) ,where i=1,2... k, with class label y∈(1,-1).The below equation is used to defined the hyperplane.

$$W.x+b=0 \tag{2.24}$$

Where:

W: represent a vector of weight ,W={w_1, w_2,…. w_n}.

b: represents bias.

x: represent the features.

The following equation use to illustrate the data classifier.

$$f(x.w.b)=sing(w\ .x+b) \tag{2.25}$$

f(x) is the function of a hyperplane with (m) dimensions which is given as the set of all points x $\in R^m$ that satisfy the equation f(x)= 0. Therefore, the function of the hyperplane f (x) acts as a linear classifier that predict the class (y) for any given point (x) , according to the following decision rule:

$$W^T.x+b\geq 1\ for\ y=+1 \tag{2.26}$$

$$W^T.x+b<0\ for\ y=-1 \tag{2.27}$$

Maximizing the margin is a problem of constrained optimization; it can be solved by using Lagrange method. Each(xi)training point is describe by the Lagrange Multiplier (α_i):

$\alpha_i = 0 \quad \Longrightarrow \quad \boldsymbol{x}_i$ has no effect on the hyperplane.

$\alpha_i > 0 \quad \Longrightarrow \boldsymbol{x}_i$ these support vectors points are located near to the hyperplane.

After getting the value of α_i , we can compute the weight and bias, the weight compute using the following equation:

$$W = \sum \alpha_i x_i \tag{2.28}$$

The points with ($\alpha_i = 0$) consider not support vectors , thus these support vectors does not play any role in determining while the support vectors with (α_i) not equal to zero value will be taking [78].

2- Nonlinear SVM

The linear classification in most cases fails to determine the optimal classification solution for that nonlinear classification which is used in such cases therefore a nonlinear kernel function is used [78].

- **Kernel Functions**: These functions are presented to transfer the training and testing samples to a high dimensional features space. The commonly used of these functions are [77]:

1. Linear kernel:

$$k(x_i , x_j) = (x_i , x_j) \tag{2.29}$$

2. Polynomial kernel of degree d

$$k(x_i , x_j) = (x_i . x_j + c)^d \tag{2.30}$$

3. Radial Basis Function (**RBF**) Kernel:

$$k(x_i , x_j) = e^{(-\gamma \| x_i - x_j \|^2)} \tag{2.31}$$

ALGORITHM 2.2: Support Vector Machine

```
Input: $S = \{(x_i, y_i) \mid x_i \in R^n, y_i \in \{1, -1\}\}$ set of $N$ training samples and class.
       $Z = \{z_i \mid z_i \in R^m\}$ the set of $t$ test samples;
Output: $Y = \{y_i \mid y_i \in \{1, -1\}\}$ set of predicted labels for the test samples in Z.
// Initialization
  $Y = \emptyset$
//Computation Learning Algorithms
For each $z_i$ in Z
    K= the kernel functions according to S
    $y$ = the class predicted by applying K on $z_i$
    $Y = Y \cup \{y\}$
End for
```

2.8.3 Hybrid Techniques Approach

Hybrid techniques approach is a combination of multiple computational techniques which provide greater advantages than individual techniques and improve sentiment (data) analysis. Use of this technique is very convenient form any because it combines two or more technologies, so it shows much better results than other methods [69].

2.9 Sentiment Analysis Accuracy Measuring

In order to measure the performance of sentiment classifier, a number of measures can be used show as follow.

1- Recall: It is used to measure the classifier ability to identify the correct positive samples. Recall formula is given by equation [80].

$$\textbf{Recall} = \frac{TP}{TP+FN} \qquad (2.32)$$

2- Precision: It is used to show the accuracy of the classifier; this measure shows the percentage of all samples are labeled positive that are actually positive. Precision is calculated by the following formula [80].

$$\textbf{Precision}=\frac{TP}{TP+FP} \qquad (2.33)$$

3- **Accuracy**: It is used to determine the performance of classifier in terms of the percentage of data that are predicted correctly [81].

$$\textbf{Accuracy}=\frac{TP+TN}{TP+TN+FP+FN} \qquad (2.34)$$

OR

$$\textbf{Accuracy} = \frac{\textbf{Number of correct predictions}}{\textbf{Total number of predictions}} \qquad (2.35)$$

4- **F-Measure**: It is a measure of statistical analysis that takes both precision and recall into account and calculates the result between 0 and 1. The closer the value is to 1, the higher the accuracy of the classifier will be [80]. F1 is calculated as:

$$\textbf{F-Measure} = 2\ \frac{\textbf{precision}\times\textbf{Recall}}{\textbf{precision}+\textbf{Recall}} \qquad (2.36)$$

where

- **TP** and **TN** denote True positive and True negative respectively, including positive and negative case ratio those were categorized correctly.

- **FP** indicates the False Positive that includes all the negative cases that were labeled incorrectly as positive whilst **FN** indicates the False Negative that includes all positive status that were labeled incorrectly as a negative. Machine learning methods are measured their accuracy using the confusion matrix, which is a table that contains a number of TP, TN, FP, and FN cases and it is a handy tool to detect efficiency of SA [81]. See Table (2.1) about the confusion matrix.

Table (2.1) Confusion matrix of two classes

Actual	**Predicted**		
		Positive	**Negative**
	Positive	**TP**	**FN**
	Negative	**FP**	**TN**

Chapter Three

The Proposed Framework

Chapter Three

Sentiment Polarity Identification Framework of Tweets

3.1 Introduction

In the previous chapter, various SA approaches reviewed and prominent techniques have been presented in the SA field. Also, identifying the research gaps from a perspective the semantic similarity and reflection of user behavior in the content. This chapter presents the research proposal, that tries to handle part of the gaps. It describes a framework that relies on machine learning.

The proposed framework exploiting the semantic similarity methods and proposed features that are related to user behavior in order to achieve the research's objectives to identify the user sentiment in tweet.

The general framework structure is introduced at the beginning of this chapter, it explains the essential framework structure along with its components and the sequence of activities. Then, more details of the methodology that uses in proposed framework are given in the following sections of this chapter.

3.2 The Proposed Framework Structure

The framework proposes a structure used to analyze tweet's text in order to identify and study affective polarity based on emotion or sentiment analysis. The suggested structure aims to integrate the corpus of tweets with a set of compact and independent-domain features involve lexical, user behavior, and semantic features through the training dataset, where automated extraction of the relevant features from training examples that labeled by humans or automatically.

The supervised classifier was adopted as a tweets-level sentiments polarity prediction methodology in the proposed framework, which in turn investigates the effect of these features and testing different classification models in process of sentiments polarity identifying.

Figure (3.1) shows the steps that associated with the proposed framework. A main idea, by receive the tweet text as input, preparing label and preprocessing is performed on the text in the preparation phase. Next, extract the relevant features through the feature extraction phase. Then, training the classifiers on a corpus of tweets within training phase and predictive the sentiment polarity or classification within testing phase.

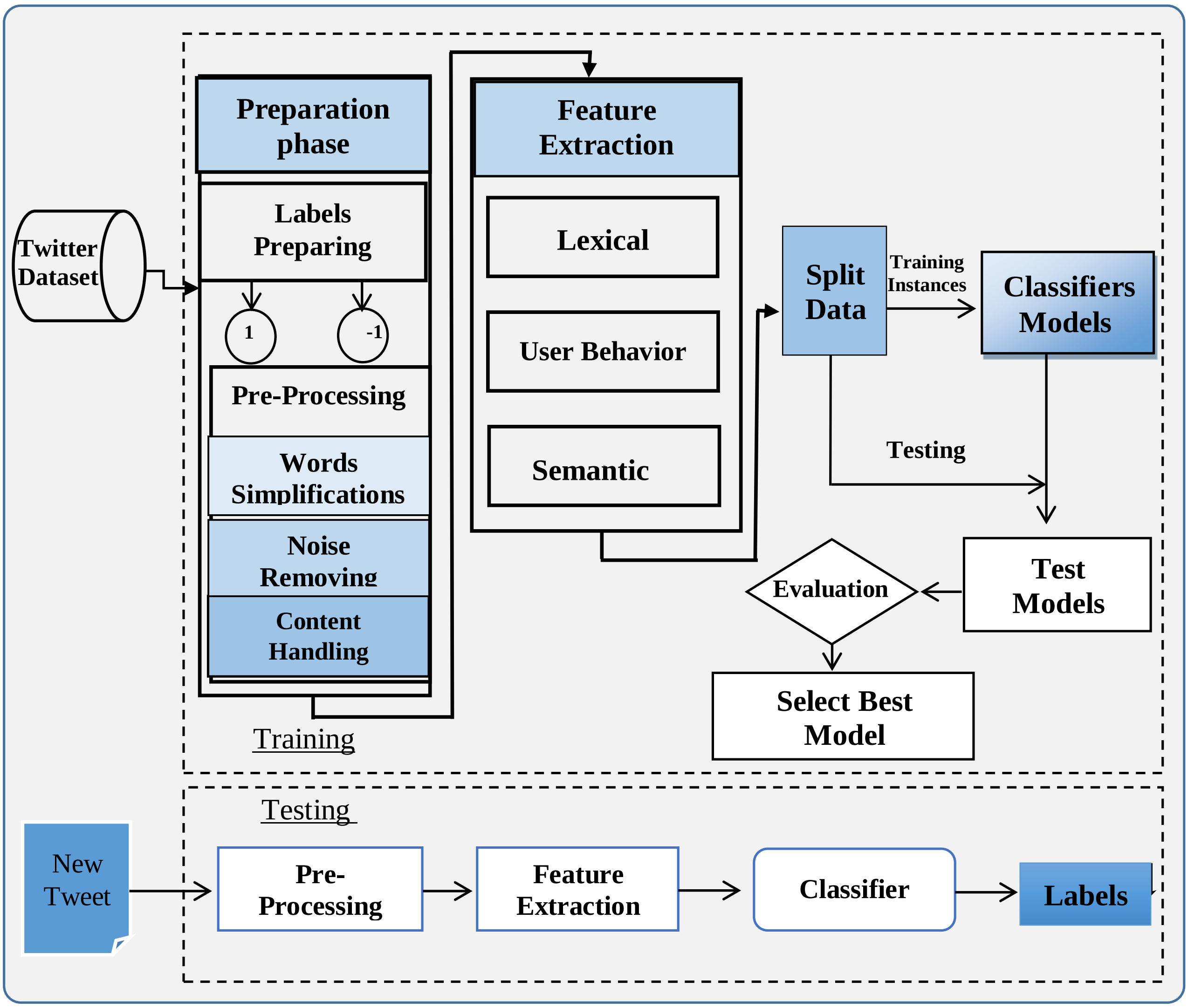


Figure (3. 1): General Block Diagram of Proposed Sentiments Polarity Identification

The framework's structure of sentiments polarity identification for tweets texts includes a series of stages, those stages look are isolated from each other. In fact, the pre-processing and feature extraction stages overlap in implementation due to the nature of the content of the tweets' texts and the features adopted in this work. For example, the keep only English characters in tweet text is a pre-processing step to noise removal. But, if executes first that leads to loss of many features that adopted by this work, such as an emoticon, hashtag, some punctuation and user mention features (type of each feature will be clarified later). Therefore, executing steps of extraction these features are required before some pre-processing step. To facilitate understanding of the framework, this chapter will discuss each stage separately from the other. The main phases and models of the proposed framework **SPTI** as a whole are described in Algorithm (3.1).

```
ALGORITHM 3.1: SPTI/ Sentiment Polarity of Tweets Identification
Input: Tweets Texts Corpus
Output: Sentiment Polarity of Tweet {Positive or Negative}
//Training Phase
// Text Preparation Stage
    For each Tweet in Twitter Corpus
        Call Labels Preparing
        Call Pre-processing
    End for
  // Create Semantic Similarity Model
    Enter all Clean Tweets of Twitter Corpus
    Call Semantic Similarity Model (Clean Tweets)
  // Features Extraction Stage
    Call Semantic Similarity Model
    For each Tweet in Twitter Corpus
```

```
            Execute Lexical Feature Extraction
            Execute User Behavior Feature Extraction
            Execute Semantic Feature Extraction
        **End** for
// **Building Classifiers Models Stage**
        Split Features Vectors in Training & Testing Instances Set
        Call Classifier Algorithm (Training Set)
        Testing Classifier Model (Testing Set)
//**End Training Phase**
//**Testing Phase**
    **For** each New Tweet
        Call Pre-processing
        Call Features Extraction
        Call Polarity Identification Classifier//predictive the sentiment of tweet
    **End** for
//**End Testing Phase**
//**End SPTI**
```

3.3 Tweets Preparation

Text Preparation is an initializing and cleaning processing for each tweet in a dataset to become more suitable for processing. The proposed framework methodology is adopting a combination of the max values of semantic similarity between tweets texts in the corpus and the label values of tweets that have max semantic similarity to compute the semantic features.

This phase in the training part, it includes two steps. The preparing labels of training dataset and perform a set of pre-processing, while in predictive part of the framework it includes only pre-processing process.

3.3.1 Preparing Label

Thesis's approach tries domain adaptation and overcomes vocabulary limitations by sharing the label's value for tweets in features values. The label indicates the specific tweet class. The labeling process is a crucial part of data preprocessing in this study, although it is considered simple. The training set of study paradigm is depending on tweets texts with pre- defined two goals (Positive/Negative) are represent sentiment polarity estimation for tweets. In this process, each tweet with a positive target mapping to a label = 1 and with a negative target to a label = -1. This mapping is an essential part of computing the semantic similarity feature that used through proposed framework. The preparing label process is describing an algorithm (3.2)

ALGORITHM 3.2:Preparing Label

Input :T set of N training samples.

Output : PT set of N training samples with class {1,-1}

```
// Initialization
PT= ∅
// Labels Preparation
 For each Tweet in T
     IF label of Tweet= Positive
          label = 1
     Else IF label of Tweet= Negative
         label = -1
    End IF
          PT = PT∪{Tweet}
End for
```

3.3.2 Pre-processing

Although tweets are limited in numbers of characters, it has many challenges beginning with a frequency appearance of slang and abbreviation, misspellings, emoticons, and special symbol much higher than reviews within other social media.

Therefore, Text pre-processing involves using various techniques to convert raw text into distinct sequences of linguistic components that have standard and notation and use it to ensure the validity of the texts of the tweets to improve performance relevant to sentiment classification. Figure (3.2) shows the pre-processing operations.

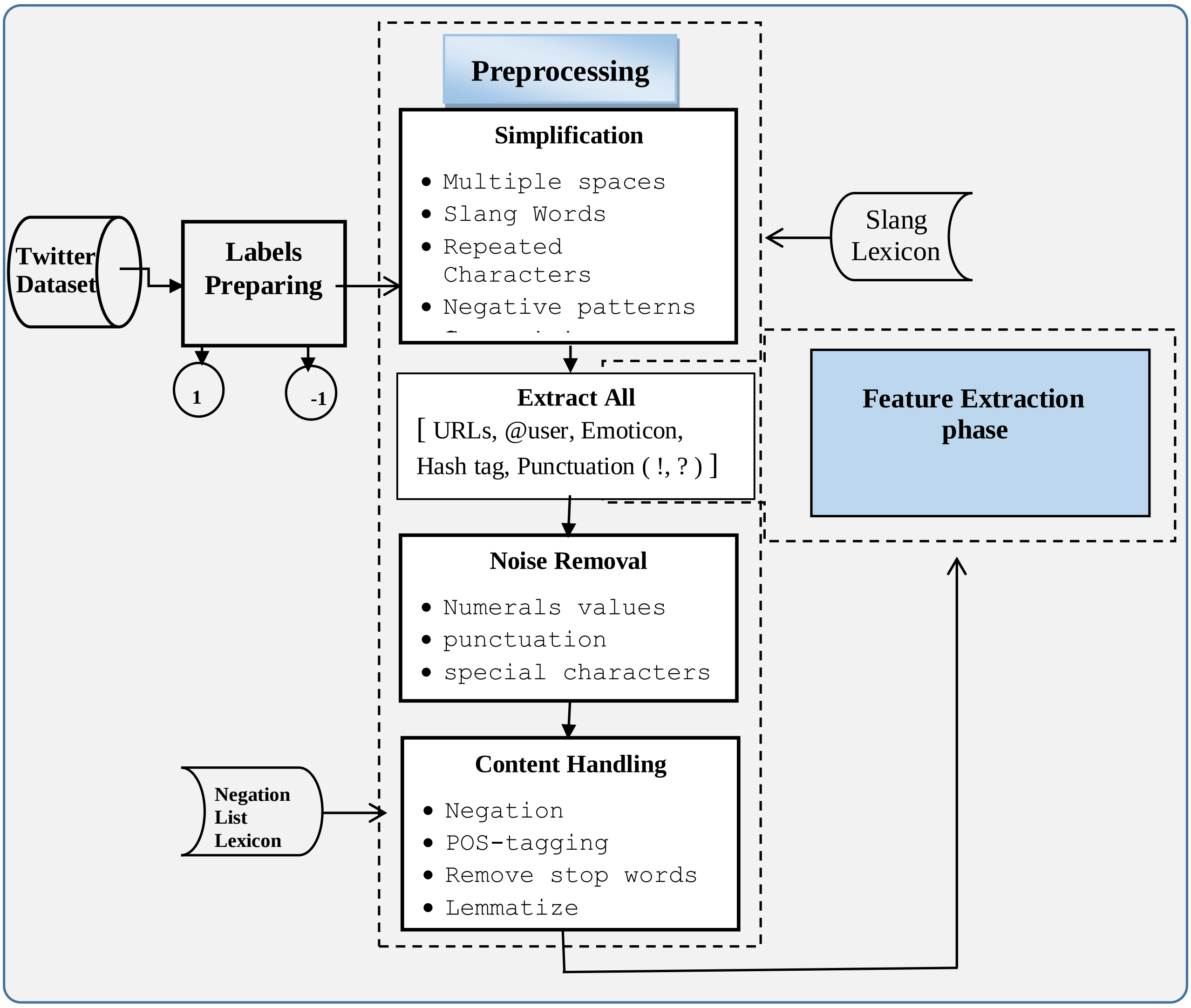


Figure (3. 2): Block Diagram of Pre-Processing Operations

As previously mentioned, the features adopted by this thesis impose interference between the pre-processing and extraction features phases. This assigns a specific sequence of pre-processing steps; as shown in Figure (3.2). In the pre-processing step, applies different steps to prepare and clean Tweets texts for processing. These techniques can be divided into three levels depending on the functional aspects (simplifications, noise removing, and content handling of tweets) as following:

1. **Tweet's Words Simplifications Aspect:**

- **Convert Text of Tweet in Lowercasing.**
- **Substituting Multiple Spaces with a Single Space.**
- **Words with Repeated Characters**: Which are commonly in tweets, users often write words without careful grammatical with repeating characters to emphasize the word meaning or an indication of confirmation e.g. "looooooove" into English. However, computers cannot recognize that word equal to "love". Therefore, this process tries returns the original word "love" based on WordNet. Algorithm (3.3) explains process. It’s based on the back reference approach and used WordNet to overcome the problem of repeated characters in original words e.g. "happy".
- **Simplifying Negative Mentions:** It is a process of replacing acronyms of negation to standards tokens such as "can't", "won't" into "cannot", "will not".

ALGORITHM 3.3: Repeat_ characters_ Replacer

Input: word, Wordnet

Output: Replaced word.

```
IF word in Wordet:
    return word
End if
replace_ word = Remove a single repeated character (word)
IF replace_ word! = word
```

```
        return Repeat_ characters_ Replacer (replace_ word)
    Else
        return replace _word
    End if
End
```

- **Slang and Acronyms**: Are handling by substituting or expanded to their original words based on external resources (Internet Slang Dict). In general, the user tends to use slang words to save keystrokes and tweet-length. Each slang or acronym token refers to an explanation. For example, "121" is "one to one", "lol" is "online love". Algorithm (3.4), it explains this process.

ALGORITHM 3.4: Slang Handling

Input: tweet, Slang Dictionary

Output: Tweet without acronyms

```
new tweet=∅
For each token in tweet
    term= Slang Dictionary(token)
    IF term! =null
        new tweet= term
    Else
        new tweet= token
    End If
End For
```

2. **Tweet's Noise Removing Aspect:**

- **Remove Digits and Numerals**: Based on the assumption that all the content the user writes in the Tweet has a purpose. In general, numbers are used to support user opinion and can classify as objective content. In general, numbers are removed from tweets with detection sentiment tasks.

- **Remove Stop Words:** Although, stop words have a role in complete the meaning of a sentence, but they are leads to low performance of classifiers. Therefore, it is advisable to delete these words from tweet text. Multiple stopword lists exist in the literature, this thesis used a default list with NLTK library except stop words belong to negation words list, some of the conjunction's words such as (but, although) and question words. Because the thesis methodology assumes these words can be indicators that reflect a certain type of user's feelings towards a specific topic. For example, question, negation, and conjunctions words.
- **Remove Punctuation Marks and Special Characters** :The assumption with the stop words, it is extends to punctuation marks and special characters. Where excluded the exclamation mark "!" and question mark "؟" and special character @ at the start of words and a Uniform Resource Locator (URLs) from removing.

3- **Tweet's Content Handling Aspect:**

- **Tagging:** Is a process of assign Part-Of-Speech tag to each of the tweet's words based on using it in the tweet (sentence). The effectiveness of the POS tag emerges to identify the Adjective and Conjunction tag of tweet's words that using several times in the proposed framework, were used in negation handling and features extraction steps.
- **Tokenization**: Is a process of splitting tweet into tokens.
- **Lemmatization and Stemming**: Both techniques used to reduce variants word (lemma, stem) forms, except stem may generate a word that doesn't exist in the dictionary, unlike lemma which can be found a word in the dictionary. Therefore, stemming may not be useful in the NLP application.

This method adopts Lemmatize instead of the stemming algorithm. Because it used SentiWordNet from NLTK library as a knowledge base to find the sentiments score or polarity of tweet's words. And the SentiWordNet relies on WordNet (English dictionary). As a result, SentiWordNet may not recognize too many stems or forms of words that result from the stemming algorithm.

- **Negation Handling:** represents an essential step, and it’s also a challenge to approaches for specifying sentiment polarity. The negation can reflect the polarity of tweets as in ("Um...Bobby Jindal & Scott Walker don't love America or its Constitution--particularly the 14th Amend"). Thus, it can constitute a weakness in the classifier's performance.

Many approaches were proposed to process a negation in texts as sub-tasks with sentiment analysis. Proposed methods to determine the scope or sequence of words affected by negation or reflecting the polarity of words. The proposed framework technique deals with negation words in two aspects.

First, it adopts a method of handling a negation based on antonyms of adjectives. Second, the negation words consider as features. Briefly, when appears any word of a tweet that belongs to the negation list. Searches for the first adjective that appears after negation word direct. If found, replace the word (adjective) by antonyms based on WordNet and remove the negation word. And if there is no adjective after the negation word, then it is counted (increase negation counter) to use as one of the features. The Algorithm (3.5), it describes the negation handling.

ALGORITHM 3.5: Negation Handling

Input: Tweet, Negation_ List, Wordnet

Output: updated tweet.

```
For each token in Tweet:
    IF token in Negation _list
        next_ Adj = retrieve next Adj wordi+1
        IF next_ Adj not null
            Ant_ Adj= Wordnet. Antonym (next_ Adj)
            Remove token
            Replace (next_ Adj, Ant _ Adj)
        Else
            Increase negation counter
        End if
    Else
        Next token
    End if
End For
End
```

3.4 Features Extraction

After preprocessing the content of the tweets in the previous phase, the dataset became more “clean” and “tidy”. More effective representation of the training dataset is the objective of this phase. Where, each tweet is transformed into a numerical representation, into a vector of 14 features called training example or instance. Each training instance associated with a label has two values (1 or -1), its predefined in the training phase and it's a target of the predictive phase. The Algorithm (3.6) describes the features extraction process.

ALGORITHM 3.6: Features Extraction

Input: **PT (polarity tweet)** set of *N* training samples *clean* and prepare class in {1,-1}.

Output: FM Matrix of Features.

```
FM= Ø
For each Tweet in PT:
    Initialize(Fv)                  // set the feature vector equal to 0
    Initialize (All parameter)// set the counters of features equal to 0
    For each token in Tweet:
        IF token in Negation _list
            Call Algorithm (3.5)
        Else IF token in Emoticon _list
            Emo _Pos. Or Emo_Neg ← Increasing by1based on Emo_pol.
        Else IF token start = '@'
            User_ M ← 1
        Else IF token start = 'http'
            URL ← 1
        Else IF token start = '#'
            Polarity = SentiWordNet(token)
            Pos. Or Neg. Or Nat_ Hashtag ← Increasing by1based on Pol.
        Else IF token in ['!', '?', question Words]
            Punctuation ← Increasing by 1
        Else IF POS (token) == 'CC'
            Conjunction ← Increasing by 1
        Else IF POS (token) == 'JJ'
            Polarity = SentiWordNet(token)
            Pos. Or Neg. Adjectives ← Increasing by 1 based on pol.
        End if
        SS = Call Doc2Vec (Tweet)
        Fv = Fv∪{Emo, Negation, User_ M, URL, Hashtags,
                 Punctuations, Conj, Adj, SS}
        FM = FM∪Fv
    End For
End For
```

A hybrid set of new, different, and traditional features are extracted from tweets, adopted by this thesis to measure the classification performance (increasing or decreasing). The features set include (14 attributes), can be divided into three categories lexical, semantic, and user behavior features. Where using the user mentions and URLs as new features to reflect the user behaved in the tweet, which has long been considered noise in previous works. And using semantic similarity in a different way to overcome on the style of writing and depend-domain problems.

At this phase, two processes are performed to extract 14 features. First, a set of 13 features is extracted, including traditional and user behavior features, that described in Table (3.1). Second, the process of extract the 14th feature, which discusses in section (3.4.3).

Table (3.1): Set of 14 features that extracted from each tweet

List of features	Set of Attributes for each feature	Description
Emoticon tokens	- Two features. - Lexical, Emo_Pos.& Emo_ Neg. - Range = {0,1, …n}.	- Obtained from Match with List of Emoticon. - Number of positive and negative emoticon in tweet. - Increasing by 1 for each emoticon counter of positive and negative, initiated by 0.
User Mention	- One feature - New, user's behavior - Range = {0,1}	Binary features when a user mentions appears within the tweet equal to 1, initiated by 0.
URL	- One feature - New, user's behavior - Range = {0,1}	Binary features when a URLs appears within the tweet equal to 1, initiated by 0.
Hashtag	-Three features - Lexical, Pos., Neg., Nat-Hashtag. - Range = {0,1, …n}	- Counter of positive, negative, and natural Hashtag. - Initiated by 0, increasing by 1 each counter corresponding to number of Hashtag appears within the tweet - Polarity value of each Hashtag is returned from SentiWord.

Continue of Table (3.1)

Adjectives	- Two features - Lexical, Pos. and Neg. Adjectives. - Based on POS tagging - Range = {0,1, …n}	- Counter of positive and negative Adjectives. - Initiated by 0, increasing by 1 each counter corresponding to number of Adjectives appears within the tweet - Polarity of Adjectives returned from SentiWord.
Count-Negation	- One feature - Lexical, Negation - Range = {0,1}	Binary features when a Negation appears within the tweet equal to 1, initiated by 0.
Punctuation	- Two features - Lexical, count of'!' & '?' - Range = {0,1, …n}	- Number of exclamation mark and question marks in list ["?", "what", "why", "where", "when", "who", "how"] that appears in a tweet.
Conjunction	- One feature - Lexical, conjunction words. - Based on POS tagging - Range = {0,1, …n}	- Number of conjunction words within each tweet
Semantic Similarity Score	- One feature - Semantic, similarity - Based on **Doc2Vec** - Range = {-1. 1}	- Value of Method semantic similarity between tweets modify with label of tweets (1, -1). - Using Doc2Vec Model to compute the score semantic similarity of tweets.

3.4.1 Lexical or Traditional Features

The sentiment orientation of a tweet can be processed based on the presence of lexical items more transparent to readers. They typically employ lexicons of items that may include keywords, emoticons, and punctuation that are direct indicators of sentiment polarities. As such, these approaches can achieve stable performance across domains. The thesis employs these features in the proposed framework by relying on integrating the SentiWordNet as a lexicon and NLP techniques such as POS to identify the polarity of tweet words. Also, it's used some punctuation that are can contribute to sentiment orientation. The lexical category includes the following features:

- **Emoticon**: Is a sideways facial or symbols expression using to represents an expression of attitude or emotions towards a particular case. Emoticons in sometimes can reflect the actual intent of the tweet. This work adopted two groups of emoticons to express the user’s mood, positive or negative. It's used the emoticons that represented by using punctuation and letters only (not a picture). Emoticon represents by two features, "Emoticon Pos" and "Emoticon Neg" that refer to a number of positive and negative emoticon are appearing in each tweet. The counters are increasing by 1 when any tweet's token matched with one of the emoticons.
- **Hashtag**: Twitter blogs allow users to add tags to tweets called Hashtag. The hashtag is beginning by ‘#’ followed by a chain of characters without spaces, e.g. #ipad and #i_hate_quotes. Adding hashtags to Tweets by the user can contribute in assigned or summarize his sentiment values to a very short text, they can be very helpful in identifying sentiment polarities.

Hashtag features involve three features represent a count of the positive, negative, and neutral hashtag that appears with the tweet text. Hashtag handling includes extraction, classification and count processes. The extraction process executed in the preprocessing phase, while the classification and count process performed in feature extraction. Each hashtag is extracted from tweet classify based on polarity into Hashtag_ POS, Hashtag_ NEG, and Hashtag_ NEU using SentiWordNet as a lexicon polarity.

- **Adjective Keywords:** The keywords or words represent the explicit aspect of expressing the sentiment of tweets. The proposed framework used only adjectives words where each adjective assigns to be positive or negative. The feature is extracted rely on a POS tagging process and polarity value is returned rely on an external lexicon SentiWordNet. There are two features of adjective keywords, The Pos. Adjectives and Neg. Adjectives. They are referring to a count of the positive and negative words, that have POS value equal to the adjective (JJ) and apparent in each tweet. The counters are increasing by 1 depending on the sentiment polarity of the returned tweet tokens. As follows, when any POS of tweet's token matched with 'JJ', then the sentiment polarity of the token is returned.

- **Punctuation**: Users tend to show some kind of emotions through the use of punctuations within them tweets, these punctuations have weight on sentiment polarity and can be used as features that can contribute in classifying the sentiment polarities of tweets. These punctuation features include three features, the question marks and words, exclamation marks, and conjunction keywords. These features represent the total count of question marks ["?", "what", "why", "where", "when", "who", "how"], the exclamation "!", and the number of conjunction words within each tweet.

3.4.2 User Behavior Features

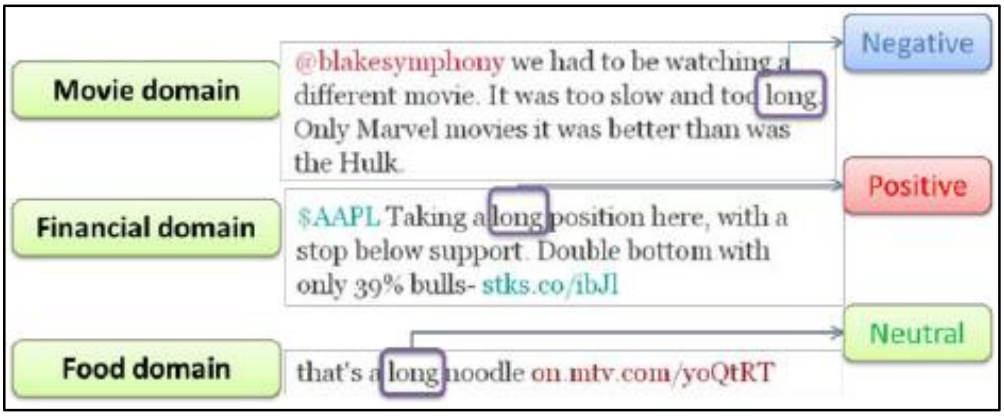
The proposed framework has a belief that any content that user-generated within a tweet, involve the sentences, words, numbers, punctuations, or symbols has a specific aim or tendency that contributes to the complete visualization or orientation of the tweet. Some of this content can reflect the user’s style, such as selecting phrases and words or report facts such as numbers, and others that reflect the user’s behavior to express his opinion or sentiment polarity.

The proposed framework uses new features "User Mentions" and "URLs" as two binary features to reflect user behavior within tweets texts, which has long been considered noise in previous works [82][83]. These features can contribute to improving the results of sentiment analysis. There are two binary features "MUser" and "URL" are equal (1) if exists within a tweet or equal (0) otherwise.

- **User Mentions feature**: It's an indication from a Twitter user to another user by writing the special character '@' before the other username (e.g. @TheEsquire). The assumption, the users tend to shows the attention to each other user is a behavior that indicates sentiment positive in tweets and their support the main subject.
- **URLs**: Is a unique address of Web page located on a World Wide Web, Twitter users able to share a link of other resources on web within tweets. Despite, the main purpose of sharing links in tweets to overcome the length tweet's limitation sending more information than 280 characters, but it can be considered an indicator of sentiment and often negative in the tweets that respond to other tweets, where the user is trying to transfer other information related to the targeted topic. In this work, the URL is a binary feature assign values (0,1) depends on appears or not it in the tweet and does not tracking the link's content in processing.

3.4.3 Semantic Features

In the second process, the 14th feature is extracted. This feature represents the semantic part of the framework. Where using semantic similarity in a different way as a feature to reflects the style of writing and overcome the context meaning and problem of dependence domain of words. The same words may have varying emotions in different domains, such as the word "long" in Figure (3.3).



@blakesymphony we had to be watching a different movie. It was too slow and too long. Only Marvel movies it was better than was the Hulk.

$AAPL Taking a long position here, with a stop below support. Double bottom with only 39% bulls- stks.co/ibJl

that's a long noodle on.mtv.com/yoQtRT

Figure (3. 3): Same word with varying emotions

The value of the 14th feature is found by calculating the average of two highest semantic similarity scores between the tweet under processing and rest corpus tweets multiply by them labels values (1 or -1).

The proposed framework exploits the Doc2Vec technique, using the cosine measure as in Eq. (2.11) to determine the semantic similarity score. The semantic similarity computes between an unlabeled tweet and all tweets in the training dataset based on the Doc2Vec model. Next, it takes the two highest semantic similarity scores and multiply by the label value (1 or -1) of corresponding tweets have a max similarity. Then the average is computed for the two resulting values. To encompass subjective sentiment information with semantic similarity into the suggested framework, using the equations described below:

$$\mathrm{Semsim} = (\max[_{\mathrm{i}}^{\mathrm{n}}\mathrm{Sim}(\mathrm{tweet}, \mathrm{T_i})]) \times \mathrm{T_i}.\mathrm{Label} \qquad (3.1)$$

$$\mathrm{Feature}_{14} = \frac{\sum_{\mathrm{j}=1}^{2} \mathrm{Semsim}_{\mathrm{j}}}{2}$$

$$= \frac{(\sum_{1}^{2}(\max[_{i}^{n}Sim(tweet, T_i)]) \ \times T_i.Label)}{2} \qquad (3.2)$$

Where:

n: number of tweets in dataset.

$T_{i:}$ training dataset tweets.

Eq. (3.1) finds (Semsim) the max value of similarity (sim) of the tweet under process and the rest training dataset tweets (T_i), then multiply by the corresponding label value of tweets that has high similar score. While Eq.(3. 2)finds the final value of the 14th feature, it computes the average of the two highest values. This methodology takes into account the possibility of similar the tweet's content to more one tweets have different sentiment polarity (positive and negative), due to the negation words, contextual polarity, and different styles of writing among the users. Therefore, the system proposed a type of normalized through Eq.(3.2). The hybrid features vector (14 features) is created for each tweet training dataset. After feature extraction, the ensemble classifier is designed for classification purposes.

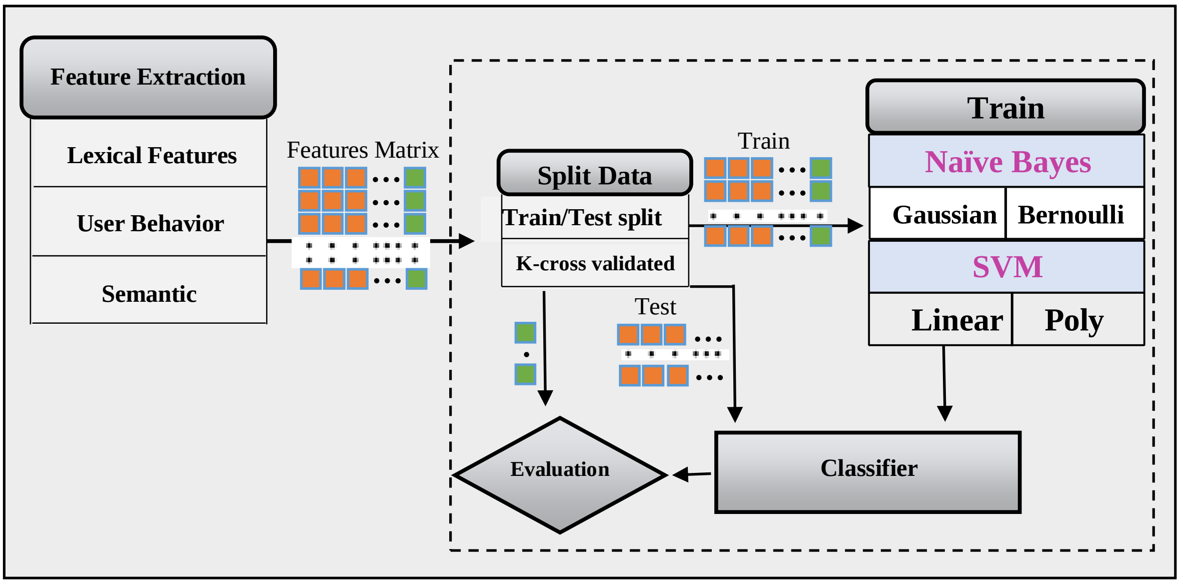


Figure (3.4) Training phase of sentiment polarity classifiers

3.5 Training Phase

The training part of the classifiers models is performed in this phase. Figure (3.4) illustrates this phase, it encapsulates some of the NLP processes and typical algorithms related to machine learning.

The objective of this phase, build, train, and testing the classifier models in order to generate a classifier that can map unlabeled tweets to appropriate

sentiment polarities. It starts by receiving the features matrix contains a set of isolated vectors, each vector has 14 features with the target label (1 or -1) as input. Next, split the features matrix using two approaches, into a 70%

training and 30% testing approach and K-cross-validated approach where k=10. Then, classifier models are generated by fed a set of training vectors to machine learning algorithms. Finally, Test the models using a set of testing vectors from the feature matrix according to the splitting approach. And uses the confusion matrix to examining the performance of the models, through the retrieved values from typical measures (precision, recall, accuracy, and F1 scores).

A. Classifier Model

The proposed framework examines two of the most popular classifiers (Naïve Bayes and SVM) in sentiment analysis to detect the emotional polarity of tweets. It examining several classifiers of Naïve Bayes and SVM models separately. Where many known parameters of learning algorithms are defined e.g. the parameter C, kernel and, gamma to set up the SVM algorithm.

- **Naive Bayes Classifier**: In this thesis, building two probabilistic classifiers model that uses the Bayes theorem in the learning pattern, a model for each Gaussian and Bernoulli of NB type. Then, tested every model separately and select the best model based on the results. The model that used with a proposed approach is dependent on the conditional probability, as described in Eq. 2.17 and Eq. 2.18 for Gaussian and Bernoulli respectively. Algorithm (2.1) in the chapter two explains the general Naive Bayes algorithm.

  Practically, in this approach for every tweet, calculated the 14 features, then NB compares the new tweet's features with the list of features to classify it to their right class or category of sentiment polarity.

- **SVM classifier**: Also, building two classifiers model to analyze the data based on hyper-planes as a fundamental concept for decision boundaries and for the separation of dissimilar classes. A model for each Linear and non-linear (polynomial kernel specifically) of SVM type. Then, tested every model separately and select the best model based on the results. The performance of the Linear model shows high accuracy from non-linear models as will see in chapter four. The linear classifier is one of the classifiers most frequently utilized. It relies on separate the vectors of the training features matrix linearly to identify the optimal separating hyper-plane which finds a margin with a maximum distance between the two classes that are far from any tweet. While non-linear classifiers depend on kernel method to overcome the situations of high dimensional space of training instances plotting through allows managing non-linear data using a linear classifier.

In order to overcome on non-linear situations in data SVM uses the polynomial kernel functions for performing computation on the non-linearly separable data to transfer the data to a more extensive dimensional space. The appropriate kernel function selection and adjusting its parameters is an optimization issue. The Algorithm (2.2) in the previous chapter explains the general SVM algorithm.

Chapter Four

Experiments and Results

Chapter Four
Experiments and Results

4.1 Introduction

This chapter is presenting and discussing the results of the tests set conducted on different datasets and it investigates the effectiveness of the features selected and the methodology that was presented in the previous chapter on overall performance. The comparison with some close works also presents in this chapter.

4.2 Specifications and Tools

The proposed framework and all phases of tasks are implemented using python 3.7. Python. Python is a high level, dynamic and widely used for general-purpose programming language. It is an efficient language with integrated systems especially the system that using artificial intelligence.

In the work of thesis used some of the major libraries and specialists for the performance of some functions such as scikit-learn which is a machine learning library, Gensim is a library for similarity retrieval, Stanford Core NLP used for identifying Parts of Speech and NLTK for natural language processing. The tests were performed on the environment: Windows 10 Pro 64-bit operating system; HP Laptop of Intel Core i7 processor 2.40 GHz speed. RAM 12.00 GB.

4.3 Twitter Dataset

Thesis uses standard Twitter datasets to evaluate the proposed framework of sentiment polarity identification. It has been used three different datasets; these are:

1. The Stanford Twitter Sentiment corpus contains two different sets [84]. First, Sentiment140 is a training set, it contains 1,600,000 tweets

extracted from Twitter and automatically labeled in positive and negative sentiments based on the emoticons expressing and second, test set (STS-Test), it is an unbalanced label contains 498 tweets distributed in 139 neutrals, 177 negatives, and 182 positive tweets. STS-Test is manually annotated. The thesis used a subset from Sentiment140 include 20,000 tweets with a balanced label and ignores the 139 tweets with neutrals label and used the rest in the evaluation process. This dataset came in CSV file, Table (4.1) shows the Sentiment140 fields.

Table (4.1): Fields of Sentiment140 dataset tweets

Seq.	Column Name	Description
0	Target	Tweet polarity (0 = negative, 2 = neutral, 4 = positive)
1	ID	Number is Tweet ID
2	Date	The date of the Tweet (Mon Apr 06 22:20:34 PDT 2009)
3	Flag	The query strings. If there is no query, then this value is NO_QUERY.
4	User	The user that tweeted
5	Text	Tweet text

2. Sentiment Strength Twitter Dataset (SS-Tweet) [85]. SS-Tweet is a dataset of tweets that are manually labeled; it includes 4,242 tweets. Each item in SS-Tweet consists of tweet's text and the values of sentiment strengths of positive and negative i.e. number between 1 (not positive) and 5 (extremely positive), while the -1 (not negative) and -5 (extremely negative).

The approach of this thesis adopts a relabeling process to dataset tweets, where each tweet is labeled with positive or negative labels only rather than sentiment strengths. To allow using this dataset in sentiment polarity classification. The tweet label positive if the value of positive sentiment strength larger than the value of negative sentiment strength, and vice versa. All tweets that have equal values of positive and negative sentiment strengths are removed. The final dataset consists of 1333 positive and 945 negative tweets. Table (4.2) shows the SS-Tweet fields.

Table (4.2): Fields of SS-Tweet dataset tweets

Seq.	Column Name	Description
0	Mean Pos.	positive sentiment strengths (1 to 5)
1	Mean Neg.	negative sentiment strengths (-1 to -5)
2	Tweet	Text of Tweet

Some statistics information about the datasets used in the evaluation of the proposed framework performance are shown in Tables (4.3).

Table (4.3): Statistics information of twitter dataset

Dataset	Total Tweets	Positive	Negative
Sentiment140	20.000	10.000	10.000
STS-Test	359	182	177
SS-Tweet	4.242	1333	945

4.4Test Strategy

This section explains the test strategy adopted in evaluating the performance of the proposed framework. Using three different datasets of

Twitter (discussed in the previous section) and adopting two testing techniques within three experiments, the framework performance is tested.

4.4.1 Testing Techniques

The proposed framework adopted two scenarios to measure for performance testing. First, the traditional Train/Test split where 70% training to fit the training of the classifier model in order to predict the test data 30% of the dataset overall. Second, a K-fold cross-validation scenario to measure the generalization of the proposed model and overcome the over fit and under fit that may be happen with the model. The k =10 fold cross-validation is similar to train/test split scenario but more subsets are applied. Where dataset split into 10 subsets one of them used to test and training on k-1 the rest of the dataset in each round. Then computes the average of accuracies of rounds.

The proposed framework applies two machine learning algorithms. Naive Byes (Gaussian and Bernoulli algorithms) and SVM (linear and polynomial algorithm) with default parameters C=1.0, degree=3, gamma=2. During the performance tests of the proposed framework two testing scenarios of techniques are implemented with each classifier. Where adopted the (70%Train, 30%Test split) and (10-fold cross-validation technique).

The results summarization that obtained are depicted from using the Sentiment140 dataset with two techniques for both classifiers in Table (4.5) and Table (4.6) respectively.

4.4.1.1 Train/Test Split Scenarios

This section presents the performance results of classifiers models that were obtained from adopted the training/testing splitting approach in building these classifiers with the twitter dataset. In experiments of the proposed framework, it used the Naive Byes and SVM algorithms. Tables

(4.4), (4.5), (4.6), (4.7) depicts the confusion matrix, which summarizes the performance of Naive Byes and SVM classifiers.

Table (4.4): Confusion Matrix for Bernoulli NB classifier with the Sentiment140 Dataset

Actual	Predicted		
		Positive	**Negative**
	Positive	**2726**	183
	Negative	178	**2635**

Table (4.5): Confusion Matrix for Gaussian NB classifier with the Sentiment140 Dataset

Actual	Predicted		
		Positive	**Negative**
	Positive	**2012**	897
	Negative	32	**2781**

Table (4.6): Confusion Matrix for Linear SVM classifier with the Sentiment140 Dataset

Actual	Predicted		
		Positive	**Negative**
	Positive	**2777**	132
	Negative	205	**2608**

Table (4.7): Confusion Matrix for Polynomial SVM classifier with the Sentiment140 Dataset

		Predicted	
		Positive	**Negative**
Actual	**Positive**	**2067**	883
	Negative	103	**2780**

Table (4.8) shows the performance of both classifiers with the Sentiment140 tweets dataset. It presents the results produced from the different classifiers algorithm. The best performance at accuracy 94% is achieved with both Bernoulli classifiers for NB algorithm and Linear classifier for SVM algorithm.

TABLE (4.8): Train/Test split results of NB and SVM classifiers with sentiment 140 dataset

Classifier	**Label**	**Precision**	**Recall**	**F1**	**Accuracy**
Bernoulli NB	N	0.94	0.94	0.94	**0.94**
	P	0.94	0.94	0.94	
Gaussian NB	N	0.98	0.69	0.81	0.84
	P	0.76	0.99	0.86	
Linear SVM	N	0.93	0.95	0.94	0.94
	P	0.95	0.93	0.94	
Poly SVM	N	0.91	0.96	0.94	0.93
	P	0.96	0.91	0.93	

Despite the performance results of the SVM (linear and polynomial) algorithms are close but, the linear shows a slight percentage accuracy better than polynomial and the execution time of the linear is much better than the polynomial algorithm. Therefore, thesis adopted the linear algorithm with SVM classifier.

4.4.1.2 K-Fold Cross-Validation

This section presents the performance of classifier models specifically the Bernoulli NB and Linear-SVM and discuss the results that were obtained from adopted the 10-fold cross-validation approach to building these classifiers. Table (4.9) depicts the performance of both classifiers with the Sentiment140 tweets dataset. It presents the results produced from the different classifier algorithm.

TABLE (4.9): A 10-fold cross results of NB and SVM classifiers with sentiment 140 dataset

Classifier	Precision	Recall	F1	Accuracy
Bernoulli-NB	0.94	0.9**3**	93.8	93.8
Linear – SVM	95.7	92.6	0.94	**0.94**

The close results of both scenarios (K-fold cross-validation and Train/Test Split) give a more reliable to the accuracy obtained from classification models. The stability of some features that adopted in this thesis such as semantic similarity can be explained the reason for these close results, where these features have a critical role in the classification decision.

4.4.2Experiments

Three experiments are implemented in the proposed framework with each experiment used the two classifiers and the same 14 features.

- The first and second experiment used each dataset (Sentiment140 and SS-Tweet) in isolation from other, split for training and testing sets then examining their performance.
- The 3rd experiment used the 20.000 tweets of the Sentiment140 dataset to train the classifiers only and evaluate the performance by STS-Test and SS Tweet as testing sets.

4.4.2.1 Experiment 1

Through this experiment used the Sentiment140 twitter dataset for measures the performance of the proposed framework. The Sentiment140 dataset is an environment that supports the orientation of this thesis in terms of the features adopted. Experiment 1 is discussed in detail in section (4.4.1).

4.4.2.2 Experiment 2

SS-Tweet twitter dataset is used in isolation for measures the accuracy of the classifier's models within the proposed framework. The SS-Tweet dataset is an environment that does not supports some of the features adopted such as emoticon and user behavior. In addition to the challenge of tweets labels which not discrete, where each tweet has sentiment strengths of positive and negative values.

The thesis objective from these experiments to evaluate the classifier comparison to automatically and manually labeling. As well, evaluating the impact of the proposed features under the independence of domain and small training data size. As seen in Table (4.10) the experiment results of classifiers training on small and challenging dataset SS-tweet.

Table (4.10): Experimental Results of Classifiers with SS-Tweet Dataset.

Classifier	Label	Precision	Recall	F1	Accuracy
Bernoulli NB	N	0.68	0.72	0.70	0.75
	P	0.81	0.77	0.79	
Linear SVM	N	0.72	0.74	0.73	**0.79**
	P	0.83	0.82	0.82	

4.4.2.3 Experiment 3

In the third experiment used the Sentiment140 tweets dataset to train the classifiers model only, then evaluate the performance of these classifiers using the STS-Test and SS-Tweet datasets as testing dataset. Table (4.8) shows the results of the third experiment.

TABLE (4.11):THE SEMANTIC SPACE IMPACT ON THE PERFORMANCE OF CLASSIFIERS MODELS.

Testing Dataset	Label	SVM				Naive Byes			
		Precision	Recall	F1	Accuracy	Precision	Recall	F1	Accuracy
STS-Test	N	0.83	0.84	0.83	**0.84**	0.99	0.75	0.85	**0.87**
	P	0.84	0.84	0.84		0.80	0.99	0.89	
SS-Tweet	N	0.73	0.72	0.72	0.77	0.93	0.74	0.83	**0.87**
	P	0.80	0.81	0.81		0.84	0.96	0.90	

Thesis objective from this experiment is to show the effect of semantic space on the framework performance. The results in Table (4.11) support the effectiveness of the semantic similarity feature proposed within this thesis. Can be seen the improvement of performance more than 6% for the same classifiers and the same dataset when used large semantic space to measure the similarity between tweets. A Doc2Vec model training process on Sentiment140 dataset, it gives more accurate similarity results. Ultimately, the accuracy of the classifier models improved.

4.5 Evaluates the Features Categories

This thesis investigates the role of each category of features in the overall performance of the proposed framework. This section helps to understand the impact of different features. As seen in Table (4.12) the lexical features have the highest rate impact on the results. This supports the exploitation of these features with other features to achieve better results. The semantic similarity feature yields acceptable improvement in the results compared.

TABLE (4.12):THE ROLE OF FEATURES CATEGORIES IN THE OVERALL PERFORMANCE IN TERM OF CLASSIFIERS ACCURACY

Features Removed	**SVM**	**NB**	**improvement percentage of features**
Semantic Similarity	0.85	0.86	8.5%
User Mentions	0.89	0.90	5.5%
URLs	0.93	0.94	0%
Emoticon	0.926	0.93	0.7%
Lexical Adjectives	0.74	0.75	19%
#Hashtag features	0.93	0.94	0%
Punctuation &Conjunction	0.92	0.93	1%
Count-Negation	0.89	0.90	4%

From experiences, found the performs of doc2vec model improve when used a large training dataset. This explains the difference in results between Table (4.10) and Table (4.11) (part of the SS-Tweet dataset), where the semantic similarity results were improved. Due to the size of the training dataset that used with the doc2vec model in Tables (4.11) larger than that used in Table (4.10), thus reflected on the classification results in general.

The proposed framework has some assumption; the users tend to shows the attention to each other using "User Mentions". It is a behavior that indicates sentiment positive in tweets and their support the main subject. Also, users tend to include "URLs" with tweets of negative sentiment toward the main subject. When considering user behavior features, Table (4.12) shows that the User Mentions feature contributes to the sentiment polarity classification, but on their own, these features are insufficiently discriminative. While the "URLs" feature did not contribute or affect the classification results, the reason it's considered a type of addressing the tweet length limitation. URLs allow sending more information than 280 characters[86]. Therefore, tracking the content of these links can improve sentiment polarity.

The hashtags feature is weaker patterns for sentiment polarity detection with independent domains. It's a set of words mostly does not reflect any sentiment polarity and sometimes an indicator of irony, but is directed towards specific subjects and may be effective with specific domain or subject or can be used corpus hashtags with predefine sentiment polarity.

4.6 Comparison Between Proposed Framework and Related Works

The proposed framework is implemented by two kinds of classification algorithms and compared with related works. Table (4.13) illustrates a comparison between the proposed framework and the existent works.

Table (4.13): Comparison between other existing works and the proposed work

Authors	**Ref. No.**	**Technique**	**Data sets**	**Accuracy**
A. Barhan and A. Shakhomirov	[9]	SVM, NB	Twitter messages	recall: 74% precision : 81%
G. Gautam and D. Yadav	[11]	NB,SVM and ME	product reviews based on twitter data	88.2%
D. Zhang, H. Xu, Z.Su and Y.Xu	[12]	SVM^{perf}	Chinese comments on clothing products	90%
O.Araque , G. Zhu and C. A. Iglesias	[15]	semantic similarity and lexical metrics	Twitter related (sentiment140, SemEval2013, SemEval2014, Vader and STS) Movie reviews (IMDB,PL04 and PL05)	89.55%
The proposed Framework		NB,SVM	Sentiment140 STS-Test SS-Tweet	94%

Chapter Five

Conclusions and Suggestions For Future Works

Chapter Five
Conclusions and Suggestions for Future Works

By applying the proposed framework to a dataset (Sentiment140, STS-Test and SS-Tweet) and discussing the results. The following conclusions and suggestions were obtained for future work.

5.1 Conclusions

This thesis has presented a proposed framework for detection sentiment polarity from Twitter texts, it targets performing a framework rely on hybrid techniques includes semantic similarity, lexical and Part-of-Speech through a set of features that reflect the areas of semantic similarity, lexical, and behavior of user. There are some conclusions and suggestions were obtained by this thesis can be listed below:

1- Although the sentiment analysis models with the independent domain do not have extremely accuracy, they can detect the changes the emotions polarity that are negative or positive. It is also worth noting, the desired application or domain supports the specific design options of sentiment polarity classification models. E.g. the words, entities also the most aspects of the debate, then empirically can be determined what is suitable for the target.

2- The semantic similarity techniques can be handling a type of implicit sentiment within tweets when taking into account the manually labeled in computing the semantic similarity. For example, when humans manually labeling the tweets, they take into account the implicit sentiment positive or negative of the tweet and on this basis, there tagging each tweet. Then, this processing is transferred to the classifier model through semantic similarity features.

3- Exploit the user behavior can provide valuable features when combined with other features supporting the ability to meet and improve the objective of sentiment polarity detection from Twitter texts.

4- According to the results of experiments in Table (4.9), Although the lexical features have the highest rate impact, the hybridization of features from various fields (semantic, lexical, user behavior) adopted by the proposed framework is achieving more accuracy with the independence domain.

5- The tweet normalization method in the preprocessing phase has an optimal effect on the performance of the proposed work. Also, the used a large semantic space has improved the doc2vec model, which in turn enhanced the semantic distance features that support the accuracy of classifiers models.

5.2 Suggestions and Future Work

Based on the conclusion from the proposed framework. There are several suggestions that can be adopted for future work. The following points are some of these suggestions:

1- The thesis suggests a method to improve the hashtag feature by enhancement the detects sentiment value of hashtag. The suggestion adopts building a hashtag lexicon from a specific Twitter dataset (depend domain) that are manually labeled in the preprocessing phase. Each entry in the lexicon consists of a hashtag text and sentiment value in a range (-1 to 1). The sentiment value is calculated based on the tweet's label where it appears in the dataset. The weight of hashtag sentiment is calculated if the same hashtag appears with many tweets that belong to different classes. While in the feature extraction phase, used a type of lexical similarity measure to determine the hashtag feature value.

2- Enhancement of the negation handling contributes to improving the overall performance. Therefore, using NLP techniques for sentence boundary detection such as parse tree is expected to support the negation treating with tweets.

3- Combining Naïve Bayes and SVM classifiers is an interesting suggestion to improve sentiment polarity recognition. Ensemble both classifiers to overcome some disadvantages of baseline algorithm.

4- Sentiment polarity detection is a very challenging process, maybe tweet's polarity changes due to one word. Therefore, the expansion in studying other areas such as argumentation mining to provide a deep understanding of the whole content and context of the tweets is one of future works. That will improve the classification accuracy of sentiment polarity and treat sarcasm or spam problems.

References

References

[1] S.Simmons and Z. Estes, "Using latent semantic analysis to estimate similarity". Department of Psychology, Warwick, Coventry, UK, 2005.

[2] R. Mihalcea and C. Corley, "Corpus-based and Knowledge-based Measures of Text Semantic Similarity". American Association for Artificial Intelligence, USA, 2006.

[3] T. Ptacek, "Advanced Methods for Sentence Semantic Similarity", (master thesis). Department of Computer Science and Engineering, Faculty of Applied Sciences, West Bohemia, Czech Republic, 2012.

[4] G. Miner, J. Elder, and T. Hill, "Practical Text Mining and Statistical Analysis for Non-structured Text Data Applications". ELSEVIER, ch2, 2012.

[5] R. Feldman, “Techniques and applications for sentiment analysis”, Communications of the ACM volume 56 issue 4 pages 82-29, April 2013.

[6] S.Jayasanka ,Th.Madhushani, E.Marcus, I. Aberathne and S.Premaratne.," Sentiment Analysis for Social Media", 2013.

[7] A.B. Eliaçık, and N.Erdogan , "User-weighted sentiment analysis for financial community on Twitter."Innovations in Information Technology (IIT), 201511th International Conference on IEEE, 2015.

[8] K. J. Kadhim," Text Similarity Based On Latent Semantic Analysis Technique" , Degree of Master thesis, Computer Science at the university of Al-Mustansiriyah in Iraq, 2015.]

[9] T. K. Landauer, P. W.Foltz and D. Laham, "Introduction to Latent Semantic Analysis" .Discourse Processes, 25, 259-284. Department of Psychology, Colorado, Boulder, 1998.

[10] A. Barhan and A. Shakhomirov, "Methods for Sentiment Analysis of twitter messages," in 12th Conference of FRUCT Association, 2012.

[11] P. Bellot, H. Hamdan and F.Béchet," Experiments with DBpedia, WordNet and SentiWordNet as resources for sentiment analysis in micro-blogging" in Second Joint Conference on Lexical and Computational Semantics (* SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013), 2013, vol. 2, pp. 455-459.

[12] G. Gautam and D. Yadav, "Sentiment analysis of twitter data using machine learning approaches and semantic analysis," in Contemporary Computing (IC3), 2014 Seventh International Conference on, 2014, pp. 437-442.

[13] D. Zhang, H. Xu,Z.Su and, Y.Xu," Chinese comments sentiment classification based on word2vec and svmperf", Expert Systems with Applications, Vol. 42, No. 4, pp. 1857–1863, 2015.

[14] A.Tripathy, A.Agrawaland S.Rath,"Classification of Sentiment Reviews using N-gram Machine Learning Approach", Expert Systems with Applications. 57. 2016.

[15] K.Kavitha and Ch.Suneetha,"Sentiment Analysis using Multiple Classifiers", Advanced Science and Technology Letters, Vol.147, pp.453-462, 2017.

[16] O.Araque , G. Zhu and C. A. Iglesias , " A semantic similarity-based perspective of affect lexicons for sentiment analysis " , Knowledge-Based Systems 165 (2019) 346–359 , 2018.

[17] A.Oussous,A.A.Lahcen and S.Belfkih,"Impact of Text Pre-processing and Ensemble Learning on Arabic Sentiment Analysis", Proceedings of the 2nd International Conference on Networking, Information Systems & Security, No. 65, pp. 1–9, 2019.

[18] J.A. Chevalier, D. Mayzlin, "The effect of word of mouth on sales: Online book reviews", J. Mark. Res. 43 (3),345–354,(2006).

[19] F. Zhu, X. Zhang, "Impact of online consumer reviews on sales: The moderating role of product and consumer characteristics", J. Mark. 74 (2),133–148, 2010 .

[20] B. Liu, "Sentiment Analysis: Mining Opinions, Sentiments, and Emotions", Cambridge University Press, 2015.

[21] W.H. Gomaa and A. A. Fahmy ," A Survey of Text Similarity Approaches". International Journal of Computer Applications, 68, 13, 0975 – 8887, 2013.

[22] L. Meng , R. Huang, and J. Gu, "A Review of Semantic Similarity Measures in WordNet ". International Journal of Hybrid Information Technology, Vol. 6, No. 1, January, 2013.

[23] G.A. Miller, "Wordnet: a lexical database for english, Commun". ACM 38 (11), 39–41, 1995.

[24] I. Dmitry, P. Bernard, and A. David,"Robust Text Similarity and Its Applications for the LSA". The Law School Admission Council, Research Report 13-04, November, 2013.

[25] P.D. Turney and P. Pantel, "From frequency to meaning: Vector space models of semantics" , J. Artificial Intelligence Res. 37 (1) ,2010.

[26] E. Gabrilovich and S. Markovitch,"Computing Semantic Relatedness using Wikipedia-based Explicit Semantic Analysis". Proceedings of the 20th International Joint Conference on Artificial Intelligence, pages 6–1,2007.

[27] L. Rudi and M. Paul, (2007). "The Google Similarity Distance". IEEE transactions on Knowledge and Data Engineering, Vol. 19, No. 3, 370-383,2007.

[28] T. Mikolov, K. Chen, G. Corrado and J. Dean, "Efficient estimation of word representations in vector space", arXiv preprint arXiv:1301.3781,2013.

[29] T. Young, D. Hazarika, S. Poria, and E. Cambria, “Recent trends in deep learning based natural language processing,” 2018, http://arxiv.org/abs/1708.02709v5.

[30] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, “Distributed representations of words and phrases and their compositionality,” in Advances in Neural Information Processing Systems, pp. 3111–3119, Advances in Neural Information Processing Systems Conference (NIPS 2013), 2013.

[31] T. Schnabel, I. Labutov, D. Mimno and T. Joachims, "Evaluation methods for unsupervised word embeddings", in: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, 2015, pp. 298–307.

[32] A.B. Abdul-Wahhab ," Opinion Extraction Framework using Aspect Based Sentiment Analysis" , doctor of philosophy thesis, Computer Science at the university of technology in Iraq, 2019.

[33] C. Aggarwal, "Machine learning for the text". Springer international publishing.DOI: https://doi.org/10.1007/97-3-319-73531-3, 2018.

[34] B.Agarwal , N. Mittal ,"prominent feature extraction for sentiment analysis". Springer, DOI: 10.10071978-3-319-25343-5 (eBook), 2016.

[35] M.Bates , "models of natural language understanding". In Proceedings of the national academy of Sciences of the United States of America, 92(22). Pp: (9977-9982), 1995.

[36] M.Akhtar , P.Sawant , S.Sen , S .Ekbal and P.Bhattacharyya, "solving data sparsity for aspect-based sentiment analysis using cross-linguality and, multi-linguality". In Proceedings of NAACL-HLT, pp (572-582), 2018.

[37] D.Sarkar ,"Text analytics with python". Apress DOI: https://doi.org/10.1007/978-1-4842-2388-8, 2016.

[38] H. M. Dawoud, "Combining different approaches to improve Arabic text documents classification," MSc Thesis, Islamic University, 2013

[39] N. Indurkhyan, F. Damerau,"Handbook of natural language processing". Second edition. CRC press, 2010.

[40] Q. Aseel"finding the relevance degree between an Arabic text and its title". Master thesis, computer science dept., University of technology. Baghdad,2017.

[41] S.Brid , E.Klein and E. Loper,"Natural language processing with python". O'Reilly media, 2009.

[42] S.V.Prasad, "Micro-blogging sentiment analysis using bayesian classification methods", Technical report ,2010.

[43] Z. Jianqiang and G. Xiaolin, "Comparison research on text pre-processing methods on twitter sentiment analysis", IEEE Access, vol. 5, no. c, pp. 2870-2879, 2017.

[44] O.Farooq , H.Mansour , A.Nongaillard , Y.Ouzrout and M. Abdul Qadir ,"Negation handling in sentiment analysis at the sentence level", a journal of computers, VOL12, number 5, pp: 470-478, 2017.

[45] R. Al–shalabi ,G.Kennan , J.M.Jaam , A.Hasan and E. Hilat **,**"stop-word removal algorithm for Arabic language". In Proceedings of the 1st international conference of information and communication technologies from theory to applications, pp: 545-510, 2004.

[46] C.Lin and Y.He ,"Joint sentiment/topic model for sentiment analysis". In: Proceedings of the 18th ACM Conference on Information and Knowledge Management, CIKM 2009, Hong Kong, China, 2–6 November 2009, pp. 375–384 (2009).

[47] C. Fox, "Information retrieval data structures and algorithms". Lexical Analysis and Stop lists, pp.102–130, 1992.

[48] L. Pavlopoulos, "Aspect-based sentiment analysis "Ph.D. thesis, Department of informatics, Athens University, 2014.

[49] M. Al-Samadi, O. Qwasmeh, B. Talafha, M. Al-Ayyoub, J. E. Benkhalifa, "An enhanced framework for aspect-based sentiment analysis and hotels reviews: An Arabic reviews case study". In proceedings of a the1th international conference for internet, technology and secured transactions,2016.

[50] Sh.Ismail, A. Alsammak, T. Shishtawy, "A generic approach for extracting aspect, and opinion of Arabic reviews". In proceedings of INFOS, a 10th international conference on informatics and systems, pp: 173-179, Egypt,2016.

[51] M. Ferreri and M. dragon, "a branching strategy for unsupervised aspect-based sentiment analysis", in Proceedings of the 3rd international workshop on emotions, modality, sentiment analysis and semantic web co-located with 14th ESWC 2017, Portoroz, 28 May, 2017, CEUR workshop proceedings, volume (1874), CEUR-WS.org (2017).

[52] M. Al-Samadi, O. Qwasmeh, M.Al-Ayyoub, Y.Jararweh and B. Gupta, "Deep recurrent neural networks vs. support vector machine for aspect-based sentiment analysis of Arabic hotels reviews". Journal of computational science, volume (27), pp:386-393. DOI: https://doi.org/10.1016/j.jocs.2017.11.006, 2017.

[53] A. G. Jivani , "A Comparative Study of Stemming Algorithms". International Journal of Computer Technology and Applications, Vol. 2, N. 6, Computer Science & Engineering, Maharaja, Gujarat, India, 2011.

[54] H.Liang,X. Sun , S. Yunlei and Y. Gao, "Text feature extraction based on deep learning: a review". EURASIP Journal on Wireless Communications and Networking. 10.1186/s13638-017-0993,2017.

[55] P. Resnik, "Using information content to evaluate semantic similarity in a taxonomy", in: Proceedings of the 14th International Joint Conference

on Artificial Intelligence - Volume 1, IJCAI'95, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, pp. 448–453, arXiv:cmp-lg/9511007, 1995.

[56] Sh. Bahri , P. Bahri and S. Lala , " A Novel approach of Sentiment Classification using Emoticons ", International Conference on Computational Intelligence and Data Science (ICCIDS 2018) , 2018.

[57] K. Pawar, Pukhraj P Shrishrimal and R. R. Deshmukh , " Twitter Sentiment Analysis: A Review " , International Journal of Scientific & Engineering Research (ISSN 2229-5518) , 2015.

[58] C.Bhadane, H.Dalal, and H. Doshi ,"Sentiment Analysis: Measuring Opinions". Procedia Computer Science, 45, 808–814.doi:10.1016/j.procs.2015.03.159, 2015.

[59] G.Gezici, B.A.Yanikoglu, D.Tapucu and Y. Saygin, "New Features for Sentiment Analysis: Do Sentences Matter?". CEUR Workshop Proceedings. 917,2012.

[60] M. Bilgin and I. F. Senturk , " Sentiment analysis on Twitter data with semi-supervised Doc2Vec",International Conference on Computer Science and Engineering (UBMK).doi:10.1109/ubmk.8093492, 2017 .

[61] Q. Le and T.Mikolov, " Distributed representations of sentences and documents". In: Proceedings of the 31st International Conference on Machine Learning,2014.

[62] Q.Chen, M.Sokolova ,"Word2Vec and Doc2Vec in Unsupervised Sentiment Analysis of Clinical Discharge Summaries",2018.

[63] S.Abbasi, **2013**, "aspect-based opinion mining in online reviews" Doctor Thesis, school of computer science, Simon Fraser University, Canada.

[64] B.Liu, "sentiment analysis and opinion mining". Morgan and Claypool Publishers,2012.

[65] H. Rubio,"Aspect-based sentiment analysis and item recommendation," master thesis, University of Autonoma, Madrid, Spain, 2017.

[66] Kh.Khan ,B.Baharudin , A. Khan and A. Ullah, "mining opinion components from unstructured reviews: a review", a journal of King Saud university- computer and information sciences, volume (26), Issue (3), pp:258-275, 2014.

[67] V. A. Kharde and S.S. Sonawane, "Sentiment Analysis of Twitter Data: A Survey of Techniques “, International Journal of Computer Applications (0975 – 8887) , 2016 .

[68]. M. Bairam and R L.Naik ,” A Study of Sentiment Analysis: Concepts, Techniques, and Challenges”,2019.

[69] A. Sh. Shirazi," Machine Learning methods to detect improper and irrelevant citations ,master's thesis, faculty of science and technology, department of electrical and computer engineering at the university of Stavanger in Norway,2018.

[70] Pang-Ning Tan, Michael Steinbach, Vipin Kumar, and Zhao Hui Tang. Introduction to data mining.

[71] Jiawei Han, Jian Pei, and Micheline Kamber. Data mining: concepts and techniques. Elsevier, 2011.

[72] S. Xu, " Bayesian Naïve Bayes classifiers to text classification" . J. Inf. Sci. 44(1), 48–59 ,2018.

[73] J.Chen, H.Huang, S.Tian and Y. Qu , " Feature selection for text classification with Naïve Bayes" ,Expert Syst. Appl. 36(3), 5432–5435,2009.

[74] B.Tang, S.Kay, H. He, " Toward optimal feature selection in naive Bayes for text categorization", IEEE Trans. Knowl. Data Eng. 28(9), 2508–2521, 2016.

[75] H.Shimodaira, "Text classification using naive Bayes" . Learn. Data Note 7, 1–9, 2014.

[76] D. H. Abd, A.T.Sadiq and A.R.Abbas , “Political Articles Categorization Based on Different Naïve Bayes Models.” ACRIT, 2019.

[77]. D. H. Abd , A.T. Sadiq and A.R.Abbas , “Classifying Political Arabic Articles Using Support Vector Machine with Different Feature Extraction.” ACRIT, 2019.

[78] P. kolluru, "SVM Based Dimensionality Reduction and Classification of Hyper Spectral Data", electronic thesis, Geo-information Science and Earth Observation of the University of Twente, 2013.

[79] M.Taboada,J. Brooke, M. Tofiloski,K. Voll and M.Stede, M, "Lexicon based methods for sentiment analysis", Computational linguistics,37(2), 267-307, 2011.

[80] T. P. Sahu and S. Ahuja, “Sentiment Analysis of Movie Reviews: A study on Feature Selection & Classification Algorithms”, IEEE International Conference of Microelectronics, Computing and Communications (MicroCom) Durgapur, India, 2016.

[81] S. Kubler, R.McDonald ,J.Nivre ,G. Hirst,"Dependency parsing" Morgan and Claypool Publishers, 2009**.**

[82] R. S. Karan, P.L. Kasar, K.K. Shirsat and R. Chaudhary, "Sentiment Analysis on Twitter Data: A New Approach ", IEEE International Conference on Current Trends Toward Converging Technologies, Coimbatore, India,2018.

[83] R. Wagh and P. Punde, "Survey on Sentiment Analysis using Twitter Dataset", International conference on Electronics, Communication and Aerospace Technology (ICECA),2018.

[84] A. Go, R. Bhayani and L. Huang, "Twitter sentiment classification using distant Supervision" , CS224N Project Report, Stanford, 2009.

[85]M . Thelwall , K .Buckley and G. Paltoglou, "Sentiment strength detection for the social web", Journal of the American Society for Information Science and Technology, Vol. 63, No. 1, 2012.

[86]https ://ar.wikipedia.org/wiki/twitter.

الخـــــلاصة

في السنوات الأخيرة ، أصبح تويتر مصدرًا لإستخراج المعلومات والمعرفة لكل من الأفراد والمؤسسات ، حيث يتم تبادل آراء وأفكار المستخدمين وتبادلها في شكل نصوص تسمى تغريدات ، حول كل ما يتعلق بحياة الناس اليومية. لذلك ، يتعلق تحليل المشاعر بتحليل مشاعر الناس وتصنيف هذه الآراء إلى سلبية أو إيجابية.

في هذه الرسالة ، تم بناء إطار عمل فعال لتصنيف المشاعر على تويتر لزيادة الدقة وتقليل معدل الخطأ الذي قد يحدث في عملية التصنيف. يتكون الإطار المقترح من ثلاث مراحل رئيسية: المعالجة المسبقة ، واستخراج الميزات ، ومرحلة تصنيف المشاعر. في مرحلة استخراج الميزات تم أستخراج مجموعة من (14) ميزة تتضمن (13) ميزة تم إستخراجها إحصائيًا من التغريدة نفسها ، أما الميزة رقم (14) فتم استخراجها باستخدام تقنية Document to Vector (Doc2Vec حيث تم حسابها من أجل زيادة دقة تصنيف المشاعر. في هذه الرسالة ، تم استخدام نوعين من المصنفات الشائعة (Support Vector Machine, Naïve Bayes).

تم اختبار إطار العمل المقترح باستخدام ثلاث مجموعات بيانات تويتر (Sentiment140, SS-Tweet and STS-Test) تشير النتائج إلى أن معدل دقة (Naïve Bayes) باستخدام مجموعة بيانات Sentiment140 هو 94٪ وعند استخدام مجموعة بيانات SS-Tweet يكون معدل الدقة 75٪ ، وعند استخدام مجموعة بيانات Sentiment140 كبيانات تدريب و SS-Tweet أو STS-Test كبيانات إختبار كان معدل الدقة هو 87٪ في كلا النوعين من البيانات ، وعند استخدام خوارزمية (Support Vector Machine) ، كان معدل الدقة باستخدام مجموعة بيانات Sentiment140 هو 94٪ وعند استخدام مجموعة بيانات SS-Tweet كان معدل الدقة 79٪ ، وعند استخدام مجموعة بيانات Sentiment140 كبيانات تدريب و SS-Tweet ، STS-Test كبيانات إختبار كان معدل الدقة 77٪ ، 84٪ ، على التوالي.

جمهورية العراق

وزارة التعليم العالي والبحث العلمي

جامعة ديالى - كلية العلوم

قسم علوم الحاسوب

إطار تحديد قطبية المشاعر للتغريدات

رسالة

مقدمة الى قسم علوم الحاسبات/ كلية العلوم / جامعة ديالى

وهي جزء من متطلبات نيل شهادة الماجستير في علوم الحاسبات

مقدمة من قبل

سناء حماد ضاحي

بأشراف

أ.م.د جمانه وليد صالح

2020 م

1442 هـ