



**A LEXICON BASED METHOD FOR SUBJECTIVITY AND SENTIMENT
ANALYSIS USING AN ARABIC TWITTER CORPUS**

NASEER ALBUHRUZI

AUGUST 2017

A LEXICON BASED METHOD FOR SUBJECTIVITY AND SENTIMENT
ANALYSIS USING AN ARABIC TWITTER CORPUS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED
SCIENCES OF
ÇANKAYA UNIVERSITY



BY
NASEER ALBUHRUZI

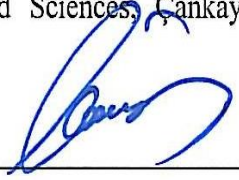
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE
DEGREE OF
MASTER OF SCIENCE
IN
THE DEPARTMENT OF MATHEMATICS
INFORMATION TECHNOLOGY PROGRAM

AUGUST 2017

Title of Thesis: **A LEXICON BASED METHOD FOR SUBJECTIVITY AND SENTIMENT ANALYSIS USING AN ARABIC TWITTER CORPUS**

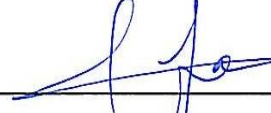
Submitted by **Naseer Mohammed Jasim Albuhruzi**

Approval of the Graduate School of Natural and Applied Sciences, Çankaya University.




Prof. Dr. Can ÇOĞUN
Director of Institute

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.



Assoc. Prof. Dr. Fahd JARAD
Head of Department

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.



Asst. Prof. Dr. Abdül Kadir GÖRÜR
Supervisor


Examination Date: 01.08.2017


Examining Committee Members

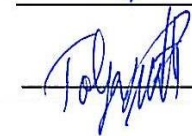
Asst. Prof. Dr. Abdül Kadir GÖRÜR (Çankaya Univ.)

Asst. Prof. Dr. Shadi Al SHEHABI (THK Univ.)

Asst. Prof. Dr. Tolga PUSATLI (Çankaya Univ.)







STATEMENT OF NON-PLAGIARISM

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Surname : Naseer Albuhruzi

Signature : 

Date : 7/8/2017

ABSTRACT

A LEXICON BASED METHOD FOR SUBJECTIVITY AND SENTIMENT ANALYSIS USING AN ARABIC TWITTER CORPUS

ALBUHRUZI, Naseer

M.Sc., Information Technology Department

Supervisor: Assist. Prof. Dr. Abdül Kadir GÖRÜR

August 2017, 66 pages

Sentiment analysis for social media is an interesting area of data mining for decision making in various domains. Therefore, continuous research is carried out in this area to cover the huge amount of data being pushed by users. Arabic is one of the ten important languages used in social media; therefore, interest in decision making anywhere needs knowledge about this. Twitter provides a platform for the exchange of opinions and ideas among users, leading decision making to building a knowledge base towards the development and planning of future outcomes. We present and illustrate how to obtain models with a high accuracy of classification by using the Lexicon-based approach. Our approach is implemented in three phases, beginning with preprocessing steps for Arabic words. The second phase discusses the extraction of more features relating to statistical and semantic orientations. We demonstrate how the extracted features (weight, score and negation) depend on two types of Arabic lexicon being clearly useful. Finally, the third phase applies a feature selection method with the Information Gain attribute evaluation and Ranker search method to find the features that have greater impact on the performance measures. We keep the features that have high rankings and remove those that have low rankings from the dataset. In the last two phases, we carry out our evaluations for all tasks using two machine-learning algorithms, namely K-Nearest Neighbor and Naïve Bayes. The accuracy for classification was found to have reached 93.56 with the Naïve Bayes classifier with a

score feature, and this task determined which one of the two selected machine-learning models is more suitable for classifying the sentiment of Arabic tweets.



Keywords: *Arabic sentiment analysis, lexicon-based, feature extraction, feature selection, KNN, Naïve Bayes, Ranker, information gain attribute.*

ÖZ

ARAPÇA TWİTTER KÖRÖSÜ İLE ÖZNEİİK VE SENTİMENT ANALİZİ İÇİN SÖZLÜK TABANLI YÖNTEM

ALBUHRUZİ, Naseer

Yüksek Lisans, Bilgi Teknolojileri Anabilim Dalı

Tez Yöneticisi: Yrd. Doç. Dr. Abdül Kadir GÖRÜR

Ağustos 2017, 66 sayfa

Sosyal medya için duyarlılık analizi, her alanda veri madenciliği yapmak için çok ilginç bir alandır. Bu nedenle, kullanıcılar tarafından her gün itilen büyük miktarda veriyi kapsayacak şekilde bu alanda sürekli araştırma yapılmaktadır. Arapça, sosyal medyada kullanılan on önemli dillerden biridir; Bu sebeple karar verme konusundaki ilginin her yere ihtiyacı vardır. Twitter, kullanıcılar arasındaki görüş ve fikir alışverişi için bir platform sağlar; gelecekteki kararların gelişimine ve planlanmasına yönelik bir bilgi tabanı oluşturmak için önde gelen kararlar verir. Çalışmamızda lexicon temelli yaklaşımı kullanarak sınıflamanın yüksek doğruluk derecesine sahip modellerin nasıl elde edileceğini sunuyoruz ve gösteriyoruz. Yaklaşımımız Arapça kelimeler için önışleme adımlarından başlayarak üç aşamada uygulanmaktadır. İkinci aşamada, istatistiksel ve semantik yönelimlerle ilgili daha fazla özellik çıkarılması tartışılmaktadır. Ayıklanan özelliklerin (ağırlık, puan ve olumsuzlama) açıkça yararlı olabilecek iki Arapça sözlüğün türüne nasıl bağlı olduğunu gösteriyoruz. Son olarak, üçüncü aşama, performans ölçümleri üzerinde daha fazla etkiye sahip özellikleri bulmak için Bilgi Kazanım Özellik Değerlendirme ve Sıralama yöntemi ile bir özellik seçme yöntemi uygular. Yüksek sıralamaya sahip özellikleri korur ve veri kümesinden düşük sıralamaya sahip olanları kaldırırız. Son iki aşamada, değerlendirmelerimizi, K-Nearest Neighbor ve Naive Bayes olmak üzere iki makine-öğrenme algoritması kullanarak tüm görevler için yerine getiriyoruz. Sınıflandırma doğruluğunun, Naive

Bayes sınıflandırıcısı ile skor özellikli 93.56'ya ulaştığı tespit edildi ve bu görev, seçilen iki makine öğrenme modelinden hangisinin Arapça tweetler için daha uygun olduğunu belirledi.



Anahtar kelimeler: *Arapça duygu analizi, sözlüğe dayalı, özellik çıkarma, Özellik seçimi, KNN, Naïve Bayes, sıralaması, bilgi kazanma özelliği.*

ACKNOWLEDGEMENTS

I would first like to thank my thesis supervisor, Dr. Abdül Kadir GÖRÜR of the Computer Engineering Department at Çankaya University, without whose helpful advice, valuable comments and guidance, this thesis could not be completed. His door was always open for me whenever I needed his help. Thanks also to my mother (God bless her), to Dr. Tolga PUSATLI, and to those who had lighted up my way through darkness. In addition, I would like to thank my wife, friends and for everyone else who had suffered with me during my journey.

TABLE OF CONTENTS

STATEMENT OF NON-PLAGIARISM.....	iii
ABSTRACT.....	iv
ÖZ	vi
ACKNOWLEDGEMENTS.....	viii
TABLE OF CONTENTS.....	ix
LIST OF FIGURES.....	xii
LIST OF TABLES.....	xiii
LIST OF ABBREVIATIONS	xiv

CHAPTERS:

1. INTRODUCTION	1
1.1. Overview	1
1.2. Features of Tweets in Twitter	2
1.3. Thesis Objectives	3
1.4. Research Questions	4
1.5. Sentiment Analysis and Decision Making	5
1.6. Research Contribution	7
1.7. Thesis Structure	8
2. BACKGROUND and LITERATURE SURVEY	10
2.1. Subjectivity and Sentiment Analysis (SSA)	10
2.2. Sentiment Analysis with Arabic Features	11
2.3. Lexicon Based Approach	12
2.3.1. Dictionary Based Approach	14
2.3.2. Corpus Based Approach	15
2.4. Machine Learning Approach	16
2.5. Related Work	17

3. METHODOLOGY OF THE RESEARCH	23
3.1. Introduction.....	23
3.2. Steps of Preprocessing.....	25
3.2.1. Tokenization.....	25
3.2.2. Normalization.....	25
3.2.3. Remove Stop Words.....	26
3.2.4. Stemmer.....	27
3.3. Feature Extraction.....	28
3.3.1. Counters for Opinion Words Features.....	28
3.3.2. Weight of Opinion Words Features.....	29
3.3.3. Score of Opinion Words Feature.....	31
3.3.4. Negation Counter and Negation Weight Features.....	33
3.4. Feature Selection.....	34
3.4.1. Filter with Information Gain Method.....	35
3.4.2. Wrapper Methods.....	36
3.5. Dataset	37
3.6. Hardware Platform.....	38
3.7. Natural Language Toolkit.....	38
3.8. Python.....	38
3.9. Anaconda Environments Manager	38
3.10. Arabic Natural Language Processing.....	39
3.11. WEKA.....	40
3.12. Supervised Learning Algorithms.....	41
3.12.1. Naïve Bayes Classifier.....	41

3.12.2. K-Nearest Neighbors Classifier	42
3.13. Cross Validation	43
3.14. Performance Measures	44
3.14.1. Precision	45
3.14.2. Recall	46
3.14.3. F-measure	46
3.14.4. Accuracy	47
4. RESULTS AND DISCUSSIONS	48
4.1. Introduction	48
4.2. Apply Preprocessing Steps	48
4.3. Display Feature Extraction Result	51
4.4. Display Feature Selection Result	54
5. CONCLUSIONS	59
5.1. Limitations	59
5.2. Research Conclusions Illustrated	59
REFERENCES	62
APPENDIX A	66
CURRICULUM VITAE	66

LIST OF FIGURES

FIGURES

Figure 1	Workflow for SSA for tweets classification	3
Figure 2	Comparison between sentiment analysis and customer feedback	6
Figure 3	General approaches and tools for Sentiment Analysis	10
Figure 4	Different between English and Arabic research	18
Figure 5	Workflow of our research through three Phases	24
Figure 6	Steps of preprocessing sequence	25
Figure 7	Workflow for extraction features with the first method	30
Figure 8	Workflow for extraction features with the second method	32
Figure 9	Workflow for feature selection with filter method	35
Figure 10	Workflow for feature selection with wrapper method	37
Figure 11	Probability of NB classifier effects on the class label	41
Figure 12	Boundaries of KNN classifier effects on the class label	43
Figure 13	Methodology for evaluation in our tasks	45
Figure 14	Comparison NB and KNN through tasks in feature extraction	53
Figure 15	Comparison of NB and KNN through tasks in feature selection	57

LIST OF TABLES

TABLES

Table 1	Features extracted from Arabic tweets with Python programs	8
Table 2	Classification examples of Arabic opinion words counters	12
Table 3	Classification examples by weight for positive and negative	29
Table 4	Classification examples of Arabic with scores of opinion words	31
Table 5	Results report for score task with Naïve Bayes classifier	44
Table 6	Outcomes of first method with lexicon has two split lists	50
Table 7	Outcomes of second method with lexicon has intensity scores	51
Table 8	NB performance measures with Feature extraction phase	52
Table 9	KNN performance measures with Feature extraction phase	52
Table 10	Number of features through feature selection phase	55
Table 11	NB performance measures with Feature selection phase	56
Table 12	KNN performance measures with Feature selection phase	56

LIST OF ABBREVIATIONS

API	Application Programming Interface
F.S.	Feature Selection
KNN	K-Nearest Neighbor classifier
NB	Naive Bayes classifier
NLP	Natural Language Processing
NLTK	Natural Language Toolkit
ML	Machine Learning
MSA	Modern Standard Arabic
SSA	Subjectivity and Sentiment Analysis
SA	Sentiment Analysis

CHAPTER 1

INTRODUCTION

1.1 Overview

With the rapid growth of social media and web applications as a means of public communication between people, huge data is being pushed at all times to carry opinion and sentiment. The analysis of sentiment and opinion started with movie reviews, which then became one of the growing fields of research [1] concerned with processing and analyzing, followed by classifying what people write or publish, thereby representing their opinions and thoughts with respect to many subjects such as events and products. Subjectivity and Sentiment Analysis (SSA) has garnered a very large amount of interest over the past few years due to the countless benefits of performing it; therefore, this becomes the focus for many researchers in this type of research in order to reach a full understanding of dealing with sentences (i.e. tweets, news, reviews, comments, etc.). SSA determines and analyzes the opinions (judgments, attitudes, views, and emotions) for users with respect to topics, general service and so on.

The importance of Sentiment Analysis (SA) is so great that whenever we need to make a decision about something, we want to hear or read opinions from other people about that subject. Therefore, most governments, business organizations and individuals become interested in SA. Moreover, Subjectivity and Sentiment Analysis (SSA) is able to determine, with good probability, whether something in written form is an opinion, and whether it can even be classified as a positive or negative opinion [2].

Generally, opinion words used in sentiment classification tasks occur as two types of state: positive opinion words expressing a desired state (e.g., *good*, *wonderful*, *amazing* and *beautiful*). On the other hand, negative opinion words, i.e., words that are

ostensibly negative, express or indicate an undesired state (e.g. *terrible*, *bad* and *unclean*).

1.2 Features of Tweets in Twitter

Twitter is a free online platform available for social media networks that provides microblogging services to broadcast what is happening anytime and anywhere in the world. Microblogging is a new source of data by using and processing with data mining techniques. This microblogging known as tweets are characterized as being short, constantly generated, and perfectly suitable for discovering knowledge by using data mining tools. Twitter provides a Search Application Programming Interface (API) by querying a special place during a limited time and by selecting the language to extract tweets related to these issues, users always push information and on a daily basis a large amount of information needs to be analyzed.

As already known, the limited number of characters of 140 per tweet makes it an appropriate environment to carry out significant scientific research related to opinion mining. In addition, it is considered appropriate when comparing it with longer bodies of previous sentiment classification, such as the case of movie reviews. Furthermore, many features can be seen and extracted from tweets, but we must take into consideration that not all features can handle the useful features of tweets. (See Figure 1 for SSA workflow and tweet classification and word features.) It is not easy to determine which feature is of benefit or not, starting with preprocessing followed by extracting features and ending with an evaluated classification. Many types of features can extract them from tweets related to morphology language, such as syntax, semantics and style. To progress well and analyze a tweet, we must process a variety of these features (e.g. positive word counters, negative word counters, negation word counters and others).

Sometimes extracted features from tweets are related to special characters; for example, users on Twitter present their opinions with Hashtags ‘#’ to sign their subscription as topics. Another uses the Target ‘@’ to alert other users automatically. In addition, they can extract features for found acronyms, spelling mistakes, or kinds of emoji's that express special meanings [2].

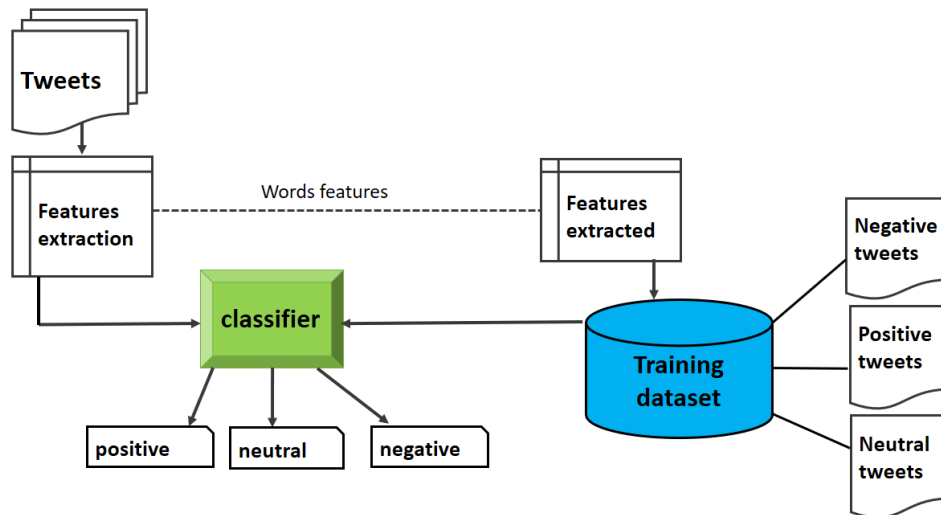


Figure 1 Workflow for SSA for tweets classification

1.3 Thesis objectives

We search for features of Arabic tweets that have positive impact through two kinds of lexicon. Moreover, we will compare classifications depending on these two lexicons. The first one will be two split files and lexicon content of more than 3100 positive and negative opinion words without scores. The second one will be 1366 Arabic scored opinion words generally used in Twitter.

The Arabic language has obtained high-level importance due to its being considered one of the top ten languages, as the fourth most used language on the Internet¹ and the sixth most used language used on Twitter². Moreover, the Arab population numbers more than 380 million. Therefore, our objectives are to extract more useful features through their information carrying Arabic opinion behind the text and to avoid those features that the process of classification would deem unsuitable. However, the Arabic language suffers from complexities of structure and morphology such that greater efforts are required to process Arabic sentences due to the need for individual solutions for each genre and task [3].

¹ <https://www.accreditedlanguage.com/2016/09/13/top-10-languages-used-on-the-internet>

² <http://www.statista.com/chart/1726/languages-used-on-twitter>

In this research, we will explain the application of our approach in three phases, the first phase being the preprocessing steps to detect opinion words of Arabic tweets in a dataset. The second phase extracts new features from the Arabic corpus dataset. These features are related to grammar, semantics and statistics, such as positive weight, negative weight, high score and negation counters. Feature Selection (F.S.) is the next phase during which a minimum number of features are obtained to achieve an improvement of classifier accuracy, followed by an application of two classifier algorithms so that we can measure and know the accuracy in the two final phases.

1.4 Research questions

In this research, we will answer the following main questions:

1. What is the importance of successful preprocessing for Arabic text on classification?
2. What is the role of using the Arabic lexicon in classification performance?
3. Which type of feature is useful for a classification task? What are the effects of adding more features to a training dataset on classifier performance measures? Do all these features have the same useful impact on classification?
4. What are the benefits of using feature selection? What method succeed with the Arabic corpus?
5. Which classifier algorithms do we need to use in tasks related to Arabic text and large datasets? In the evaluation for high accuracy obtained from our work, which is the better and more appropriate of these tasks for opinion mining?

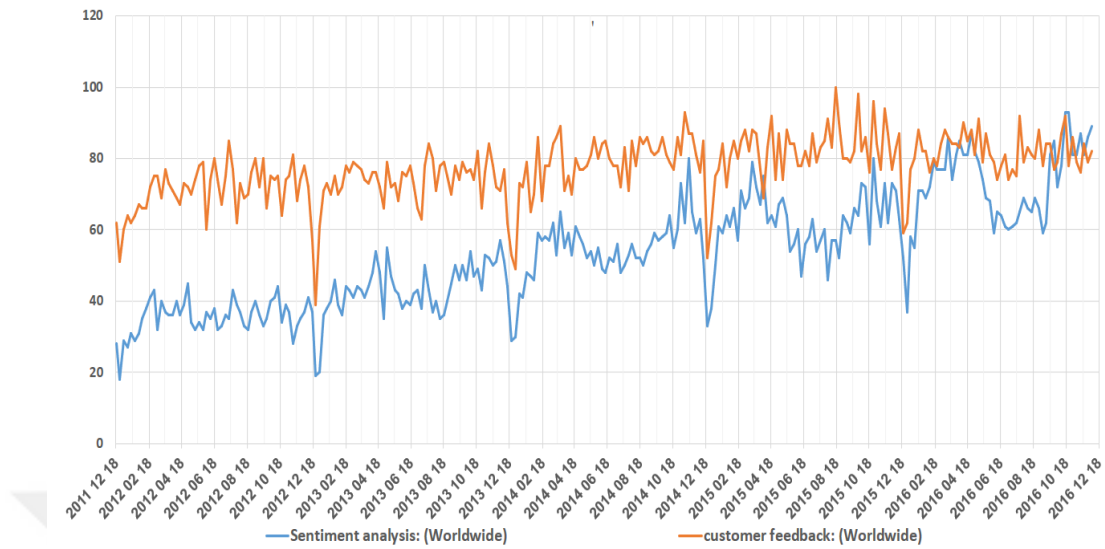
1.5 Sentiment Analysis and Decision Making

Sentiment analysis is an attractive research subject resulting from the analysis the feelings of people behind the texts written by users. Therefore, it is an important phase in information gathering and decision-making. These decisions include several scopes, such as politics, business, health, sports and so on. For example, popular social media websites (such as in the scope of our research on Arabic Twitter) encourage users to contribute content by posting their comments about different aspects of life that create a large stream represented as raw data. These data are converted into information and finally converted to knowledge about the users preferences for decision-makers.

In the scope of business, knowledge represents how people think about products, services, events and so on. Decision makers using this knowledge can check whether or not they achieve customer satisfaction for decisions to enhance quality or/and quantity of their products. Moreover, there are benefits for decision-makers to obtain a comprehensive view of society and its needs by monitoring people's reviews on social networks. In addition, social networks are useful for posting ads. Many contextual advertising systems look at what people post on their microblogs and anyone interested can access them to gain ideas about their opinions or feelings [4].

Figure 2 is the chart which we created using Google Trends by entering the search terms "Sentiment Analysis" and "Customer Feedback." It is notable that these two terms have more closely paralleled each other in the last two years.

Google Trends search terms ‘Sentiment Analysis’ and ‘Customer Feedback’ in last five years.



Source: (<https://www.Google.com/Trends>)

Figure 2 Comparison between sentiment analysis and customer feedback

Decision-makers always attempt to come to an accurate decision that will be of benefit to their businesses, institutions and remaining organizations associated with them. For example, the tourism sector, which include, (Airlines, Restaurants, Hotels and so on) will be interested in extracting opinions of Arabic people in order to develop and improve business areas.

For example, ('شاهي وجلسة خطيرة توب كابي قصر السلطان # تركيا ') which mean, “Drinking tea with fantastic session Topkapı Palace Sultan # Turkey” is classified as positive and refers to an opinion about tourism.

The same can progress in other scopes of importance such as politics, health, sports, and so on.

1.6 Research Contribution

Our research contributes to building a better model of a variety of scopes of decision-making (e.g. business, politics, health, sports and global security) by processing from three phases, starting with preprocessing for words in instances, followed by extracting more features of Arabic tweets through the lexicon. Therefore, our contribution demonstrates the preprocessing and extraction of more features related to semantic and statistical linguistic analysis, the evaluation of all features (already existing and new) to decide which will have more impact on the classifier performance so as to keep them, as well as ignoring those with almost no effect on the implementation of the classifier.

We write our Python programs to extract more features from each tweet in our dataset. The outcomes of features for one of these programs are described as follows: **positive words counters**, **negative words counters**, **high count**, **state**, which represent the polarity of the tweet, **positive weight** which represents positive word percentage in the tweet, **negative weight**, which represents negative word percentage in the tweet, and **negation counters** and **negation weight**, which represent negation word percentage in the tweet. These are list of positive words in the tweet, negative words in the tweet and negation words.

While the second of our program depends on opinion words found in the available lexicon, which is open source. In order to calculate the intensity of the **score of positive** tweets by every word having a score in the lexicon, the intensity of the **score of negative**, **state**, **high score**, **negation counters** and **negation weight**, these extracted features of tweets are added to the training dataset to improve the performance measures of the classifier. We will experiment with these features and select those that have more impact. Table 1 shows the features extracted from Arabic tweets.

Table 1 Features extracted from Arabic tweets with Python programs

Features extracted by method 1	Features extracted by method 2
Positive words counters	score of positive words
Negative words counters	score of negative words
Positive words weight	High score for tweet
Negative words weight	Negation words counters
High weight for tweet	Negation words weight
Negation words counters	
Negation words weight	

Moreover, our contribution depends on using one of the feature selection methods to reduce data dimensionality and improve accuracy of the model classifiers. Our study is one of a few attempts to prove the role of F.S. for the Arabic language which enhances implementation by removing features that do not provide more information for a classifier.

The Ranker search method and the Information Gain attribute evaluation for every feature found in the training dataset depends on the evaluation of information gained from each feature in order to evaluate every feature in the training dataset with numbers representing a ranker for every feature.

1.7 Thesis structure

Through Chapter 2, we will explain the general concepts of subjectivity and sentiment analysis, and features of Arabic relating to SA. In this chapter, we describe a lexicon based approach in the manner of supervised learning and word features.

The third chapter will be a type of paper methodology for doing our tasks on the dataset with two types of Arabic lexicon. A sentiment classifier will be conducted with some major experimental evaluations, using the Naïve Base algorithm and a comparison with KNN (k-Nearest Neighbors). We will explain and describe the steps of

preprocessing of the training dataset and we will describe the features extracted (weight, score and negation counter). Furthermore, this chapter will explain features selection methods and the method we are implementing on the training dataset.

In chapter 4, there will be an explanation and discussion of the results of these two classifiers and finally in Chapter 5, we will present the research conclusions and our findings.



CHAPTER 2

BACKGROUND AND LITERATURE REVIEWS

2.1 Subjectivity and Sentiment Analysis (SSA)

As previously stated, subjectivity in natural language generally pertains to one's feelings, opinions, and personal evaluations. The process of subjectivity classification is the action of determining the objectivity or subjectivity of texts; for example, “*new Nikon released*” being objective, “*a great new Nikon camera*” being subjective with positive polarity and “*The picture quality of this camera is poor*” being subjective with negative polarity.

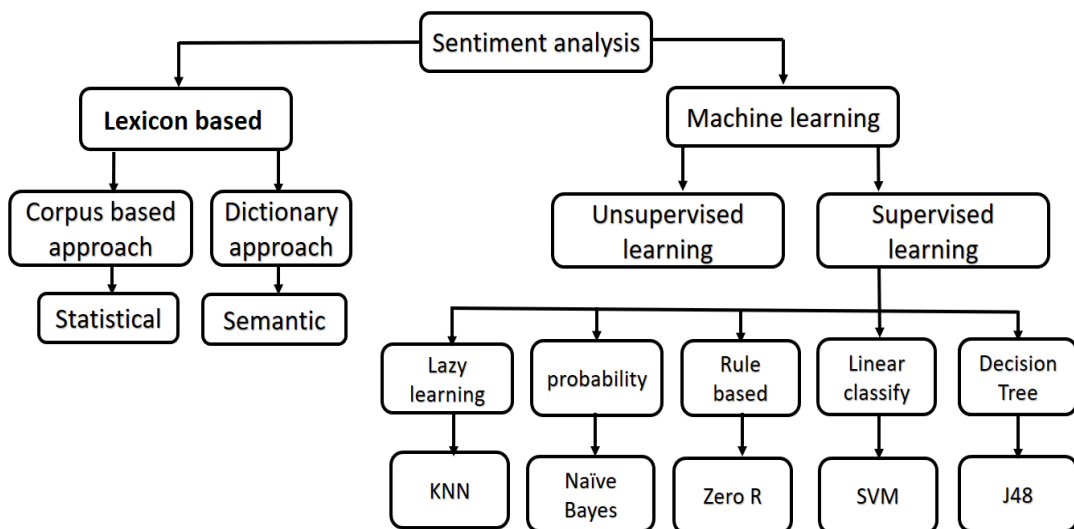


Figure 3 General approaches and tools for Sentiment Analysis

In most cases, sentiment analysis is carried out by processing and analyzing any text by following two methodologies, one of which is lexicon based approach and the other

being Machine Learning, as shown in Figure 3, (clarified later in this chapter). Three important elements for the implementation of these approaches are the lexicon, corpus and tools.

2.2 Sentiment Analysis with Arabic Features

Recently, subjectivity and sentiment analysis has gained significant attention, but most resources (tools and lexicons) deal with English and built systems for this. For other important languages, such as Arabic (our research scope), there has been until now continuous insufficiency, most important reason of which is the lack of resources and dataset collections this from hand and from another hand being rare with regard to the Arabic sentiment lexicon [3].

Arabic is a Semitic language, which in its classical form, is used in the Quran Kareem. It generally consists of two approaches for writing, the first being dialects of the Arabic language, which are variations from one Arab country or region to another. This approach is most commonly used in social media, such as on Facebook and Twitter, which uses informal Arabic language due to their not requiring much attention in writing rules, grammar for sentence construction. Hence, a rather informal manner of writing may be found in Arabic tweets.

Modern Standard Arabic is another approach used in written forms and expressions of opinion, this differs from the first approach to writing and expression by using terms of words and writing in which there is commitment to grammatical rules. Moreover, there is a mixture of the two approaches to writing and expression in the Arabic language in microblogs [5].

Arabic has the advantages a rich configuration system (morphology), which necessitates the use of a number of tools and dictionaries (lexicons) for research so that a fuller, deeper and better understanding of the textual content of the words can be developed.

We explain in Table 2 the approach of using a lexicon-based approach for two examples of Arabic sentences, their meaning in English, and the sentiment analysis result for each in terms of the value of their respective polarities.

Table 2 Classification examples of Arabic opinion words counters

Arabic tweets	English meaning	Pos. count	Neg. count.	Polarity
نفرح عندما نرى ابتسامة الفقير	We <u>rejoice</u> when see the poor man's <u>smile</u>	2	0	Positive
الجو الممطر ممل و مزعج للقيادة	The rainy weather is <u>boring</u> and <u>tired</u> drive	0	2	negative

2.3 Lexicon Based Approach

This methodology requires the lexicon to contain opinion words and phrases with polarity (positive and negative). Additionally, the approach depends on several steps of preprocessing, beginning with splitting the sentences to tokenize the parts, removing the words not carrying opinions, and benefitting from the stemmer and searching for every part throughout the lexicon on polarity. Moreover, the domain of opinion words that is considered important covers the scope of the dataset [4].

Such an approach enables opinion words to be handled by the system which are the most important parts of the selected subjectivity, and by determining opinion word polarity, it becomes possible to know what these words are expressing terms of desirability or undesirability. The desirable states of opinion words represent positive opinions, while undesirable states represent negative opinions.

These are respectively combined into lexicon sets of such desirable words, and similarly for undesirable states, to build an opinion lexicon. An important indicator of opinion extraction is the adjectives, these being a dominant element in subjectivity content in text and the most important factors when extracting opinions e.g., for

positive polarity (*great, wonderful, good, beautiful* and *amazing*) and for negative polarity, (*bad, awful, terrible* and *poor*) [3].

When using the Lexicon Based approach, it is important to note that lexicon content comprises primary opinion words (such as the number of positive and negative adjectives), which are the qualities measured for the lexicon since they have a great influence on text coverage in subsequent results and on overall accuracy . In addition, when an extracted opinion of a text is calculated through word polarity, some words which come in the sentence are considered for their effect, for example, negation phrases will give an inverted meaning and switch the expressed opinion (e.g. *not, no, un*). An example can be seen thus, “*The weather is not clear*” most probably meaning “*The weather is cloudy or foggy*”. Moreover, the same situation applies to other opinion word types such as to verbs and nouns. These can be used to express opinions with verbs such as *hate* and *like*, and with nouns such as *junk* and *rubbish* [6, 7].

Moreover, opinion words can be divided into two types: direct opinion words and opinion words for comparatives. Lexicons for direct opinion expression have been explained above. While opinion words for comparative expressions are not used to express opinions directly about an object, they are used to present superlative or comparative similarities and differences of opinion between one or more objects.

Opinion words of this type can also be divided as desirable or undesirable (e.g., *most, least, best, worse*). Comparative sentences need to be analyzed more for text to extract the two polarities of expression opinions. Such a style is frequently found on social media platforms such as Twitter. For example, (“*The iPhone is better than the Lenovo phone*”) which expresses a positive opinion about the iPhone and simultaneously a negative opinion about the Lenovo [8].

The lexicon is one of the main aspects of sentiment analysis through opinion words or phrases which can be used to express subjectivity about products, services, things, and events. In this research, we used two types of Arabic lexicon for positive and negative combined words and phrases.

Although the lexicon based approach is a simple and widely useful method, it should be improved in order to overcome a number of difficulties, such as when word opinion has positive polarity in a domain, but the lexicon does not recognize it when being used in another domain (e.g., *'quite'* being positive with car, but negative with speakerphone). Therefore, an approach that makes use of lexicon does not work as desired with several languages or in various domains.

Occasionally, the basic opinion words or opinion phrases in the lexicon are insufficient to determine the polarity to which group these tweets belong. Here, the words of a sentence will not belong in the lexicon, so the tweet will be considered neutral, which, in fact, belongs to one side of a polarity [9].

2.3.1 Dictionary Based approach

This is the approach of processing by the simple technique of manually collecting a small set of opinion words represented as a seed, and doing this with a known orientation (positive or negative). Sometimes a lexicon is insufficient to cover and classify the text; therefore, there is the need to extend to more opinion words. One approach is to add newly found words to a set list by searching for synonyms and antonyms of the seed words. For example the 'WordNet,' which is an online lexical database dealing with the English language.

Repetition of this process on opinion words to acquire more words is very important in addition to the implementation of manually adding new words and having a preference for using a better opinion word list due to the profound effect on extracting and determining which words of a text belong to which polarity.

The main shortcoming of this approach is its inability to specify the orientation domain of the opinion word (e.g., *quite* as an opinion word for *car* representing a positive opinion, but when expressing it with a speakerphone, it will be negative) [1].

2.3.2 Corpus Based Approach

A corpus is an essential component that can help to deal with the shortcomings of a dictionary based approach, in which content pieces of text are assigned a polarity (positive, negative, neutral). Building a corpus for sentiment is not easy. Such a corpus can be used to train a classifier algorithm to detect the sentiment of a new text [10].

This approach depends on a seed opinion words list such as a positive adjective to find other positive adjective opinion words in a large corpus by using linguistic limitations that can help to determine adjectival opinions. In order to find other opinion words, linguistic constraint is applied, and this depends on a number of words such as *and*, *or*, and *but*. For example, with *and*, it usually represents linking adjectives that have the same semantic orientation in the same sentence (e.g., “*This car is beautiful and spacious.*”). Here, ‘*beautiful*’ is known to have a positive polarity. It can be concluded that the second adjective ‘*spacious*’ is of positive polarity and will be defined as a new positive word for the lexicon [1].

The importance of the corpus lies in its linking connections between the seed opinion words associated with other words to give a specific opinion, while in the lexicon, the words sorted as positive, negative or neutral solely determine the analysis [8].

An Arabic corpus has hitherto not been widely or generally available due to the complexity of the morphology of Arabic. In spite of this, Arabic is a language of natural language processing and it has importance related to how hundreds millions of users use it in multiple genres of social media and the Internet web 2.0, which lead businesses to care about the ideas and feelings of Arabic people.

Occasionally, manual labels represent the absolute truth about the classification of an opinion or sentiment of a tweet because it is done with human effort. This depends on manual labels that will teach the classifier the actual opinion class, which is a good approach to predict with high accuracy, but at high cost and difficulty [9].

2.4 Machine Learning Approach (ML)

The more significant part is a supervised machine learning algorithms, which are classifier algorithms training on classified instances previously labelled manual or automatic. The aim is to build a more accurate model according to our dataset. New unclassified instances have no need to be searched again through the lexicon to determine the polarity of sentences.

We have carried out the classification of training dataset with several classifications to know which one is appropriate to conduct the application of our approach, therefore we search to access several kinds of ML algorithms (e.g. Zero R, decision trees, Naive Bayes, K Nearest Neighbors, and Support Vector Machines, etc.) to find which are appropriate to our tasks. The authors [11] explain not all of them are appropriate for all tasks, which instruct the algorithm to use data or experience to solve a classification problem.

Most of the time, researchers focus on using different machine learning algorithms and build classifiers which require much effort to annotate documents.

Training a classifier on a dataset labeled before with consideration for the fact that some supervised machine learning techniques requires a large corpus of training dataset. However, some classifiers may fail because they are sensitive to the quality and quantity of the training dataset, or at times, when a training dataset is insufficient [12].

By using machine learning applied through this approach, most researchers focus on measuring the weakness and strength of supervised learning classification algorithms through their datasets [13].

Further explanation about our supervised machine learning algorithms is found in Chapter 3.

2.5 Related Work

Much research on sentiment analysis has been done with the English language. The authors [2], has examined data on Twitter to analyze sentiment depending on feature extraction. Some of these features count the negation words, the positive words and the negative words. Execution of the mission of the work occurs by searching through an English dictionary for the polarity of words. Using unigram to split tweets and to count opinion words with natural numbers will be represented for each label.

In recent years, research on sentiment analysis of the Arabic language has been increasing around the world. SSA as an approach processes in two methods: one with a class feature for objectivity vs positive vs negative. The other method processes in two stages: the first of which is subjectivity, then reaching polarity. The author [3] deals with many features of extraction and applies the "*has POS adjectives*" feature and the "*has NEG adjectives*" feature by manually creating a polarity lexicon. For subjectivity classification, there is a search through the sentence for the adjective followed by a search for the indicated adjective as being positive and negative. With these two features, there are binary labels for each. The best results were achieved when combining the noun feature "part of speech tagger" with other features. Their experiments demonstrated the difficulty and complexity of the characteristics for SSA in the Arabic language.

The authors in [6] used an approach to the lexicon semantic orientation to calculate the extracting sentiment from the text by words annotated to both of their semantic orientations in terms of polarity and strength, considering the effect of intensification and negation.

The authors [9] adopt a lexicon-based approach and represent it as the heart of their framework, additional to opinion indicators (word and token). First, they identify the opinion words in the sentences by matching words of each sentence with words found in the opinion lexicon. One of the methods used by the author is the orientation score +1 for words of positive opinion and a score of -1 orientation for negative opinion words. Then the author processes and computes a score for each entity by summing. This method is presented as automatically labeled instead of manually labeled by

training the classifier on examples given in the lexicon-based approach to assign polarities to linguistic items in new Twitter posts.

The amount of Arabic language research for SSA is low compared to that devoted to English. Figure 4 shows the great differences between the two languages, as prepared by the author [10], by using the Google Scholar website. A few studies applying sentiment analysis to the Arabic language apply comparisons between two or three classifier algorithms, especially those related to extracting, analyzing and then classifying written Arabic text in Twitter in order to clarify and reach a full understanding of Arabic opinion mining.

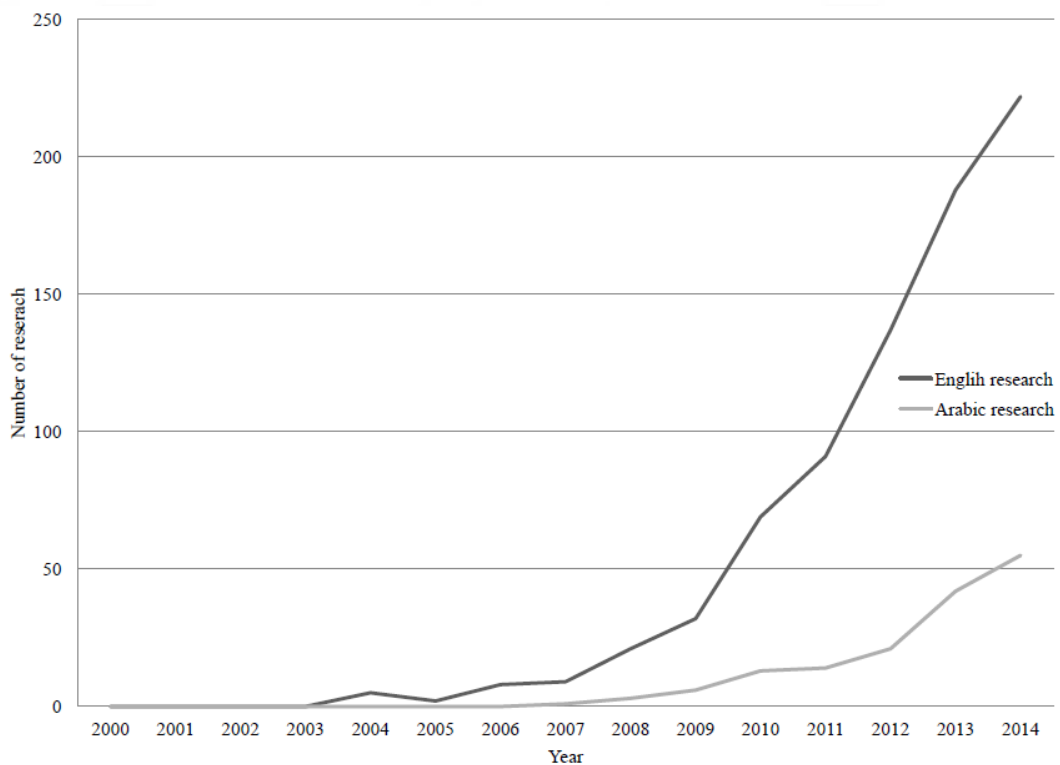


Figure 4 Differences between English and Arabic language research

In addition, the author [10] captured sentiment orientation by proposing different kinds of features for Arabic morphology, semantics and stylistics. Moreover, the author explains the same for the English language such that negation is effective in Arabic polarity classification, that is, the two types of negation words Modern Standard

Arabic and Dialects use to build lists of negation words, which helps in tasks of natural language processing.

The authors [11] use the steps of the sentiment analysis algorithm to calculate the strength of the positive and negative of the text. For example, if we analyze a text and extract three positive words and two negative words, the result of the classification will be 60% positive strength. The authors explain the lexicon with more opinion words, which can be more beneficial to the classifier.

Additionally, another method by the authors [14] is to use through their work unigram, bigram and trigram for bags of words with tweets. They consider unigrams to be the simplest features used for extraction and they provide good coverage for data. Moreover, bigram and trigram provide good abilities for the negation and sentiment phases. The authors represent tweets as holding an overall feeling with one of three labels as being of positive, negative or neutral sentiment. Looking to improve the performance measures at the sentence level related to Egyptian dialect sentiment analysis, the authors study the effect of increasing the size of the corpus in equally incremented instances, every time increasing by 600 tweets, until reaching a total number of classified tweets numbering 4800. Follows the measurements of the accuracy each time.

Furthermore, some work focuses on explaining the significant gain in detecting subjectivity and sentiment analysis with the difference on sentiment analysis.

The authors [15] address the tasks of automatic SSA for Arabic corpus tweets. The corpus is manually labeled for class with automatic annotations in a variety of extraction features. The proposed approach is to compare the effect of the subset from these features on Arabic classification. The dataset is classified as polarity vs neutral with labels of three types: positive vs negative and the third being neutral. The authors explain the effectiveness of different feature sets using comparisons between algorithms to determine subjectivity and sentiment analysis.

The authors [16] explain that the extraction and collection of features often gives better performance of the progress for a classifier algorithm with two class labels of negative

and positive. The presence of a neutral label class reduces the performance of these features. Explain the use of a large lexicon for positive and negative may be not enough: A word may appear in the positive lexicon, but in a sentence, the writer may be meaning a negative phrase or vice versa. Moreover, in many cases, neutral words in context are represent positive or negative outside context or in another context.

One of the important features is n-gram, which is used by authors [17] to extract unigrams and bigrams for the corpus, followed by calculating the frequency of each one if they appear more than 5 times when extracted to be a candidate feature. Then, for each tweet count, the frequency of each candidate is found. Their experiments are process without consideration for the negation factor in Arabic phrases.

The authors [18] explain the challenges found in Arabic sentiment analysis, one of which is the negation word or phrase, by detecting and counting the negation words. In addition, classification for the Arabic language faces the challenge of dialect (internal Arabic language for each Arabic country), especially when written in social media. While doing experiments with the use of the Remove Stop Words filter and Stemming for words and without their use, the authors finally depended on labeling the training dataset with four attributes per tweet, namely positive, negative, neutral and not applicable.

The authors in [19] present free source available lexicon for Arabic opinion words taken from the Arabic Twitter domain. Manually annotated words of this lexicon use Best-worst Scaling, start with a high degree score for positive and decrement according the strength of words until reaching the last negative words. For example, '*succeed*' is more positive, or we can say (less negative), than '*improve*' while '*fail*' is more negative or we can say (less positive) than '*setback*'. The results of three teams for progress sentence tweets appending the single words are more noticeably higher than multi-word phrases, especially for the Arabic language.

The authors [20] organized techniques of sentiment classification into a number of categories; in one of the categories, score-based methods are normally used along with semantic features. Generally, this technique classifies the sentiment of messages based on the sum total included for positive and negative sentiments.

Most of these literature reviews attempt to extract a variety of new features; however, not every feature is useful for classifier implementation; therefore, we need to search for the features with more benefits and which are have little or no benefit using a method known as features selection.

For the reason above, researchers are continuously endeavoring, through special techniques and methods, to search for those features with high values for the classification process and disregard other features which have low value.

The authors in [21] did their experiment to improve the performance of classifiers, and so used a feature selection depending on Ranker search methods and Information Gain attribute evaluation. The behavior of classifiers will have a clear impact through reduce data space dimensionality, redundancy and they will increase the classifier speed. In addition, they explain this by implementing the removal of low ranker features, thereby improving the accuracy of the classifier.

Moreover, the authors [22] implemented their experiments with features selection on tweets to explain the impact of their own classifier performance by using different classifiers as well as a number of features. The results of their experiments demonstrate that feature selection can significantly improve classification performance in comparison to those that do not use feature selection. They explain that the dataset could be of very high dimensionality in sentiment classification; this state comes from a large number of features that can be generated from tweets which are combined with a large number of instances. While their results are explained, not every ranker improves the results.

For each of the above studies we will build methodology for our approach. The outlines of this will depending based on implement following concepts, implement the steps of the preprocessing to extracting for the words carrying opinion from the tweets. Using two methods for search through two of the Arabic lexicons, first one content two list for positive and negative opinion words, the second lexicon content on opinion words with value represent intensity score for each word, extract more features depending and related with these two lexicons. Then adding them to training dataset to get a better for performance of our classifiers. Then we will implement one of

methods for selecting features have more useful and influence, by evaluate these features that have more impact on the classifier to keep and these have a little impact on classification process will remove.



CHAPTER 3

METHODOLOGY OF THE RESEARCH

3.1 Introduction

In this chapter, we provide clarification for the execution of more preprocessing and configuration dataset to make the words more analytical and understandable by the classifier.

We implemented the extraction of features for the dataset after we finished preprocessing them, then we added these new features to the training dataset so that they could be evaluated with widely used performance measures. The new features that are extracted are related to statistical and semantic orientation features as a type will carry information from two types of Arabic lexicon for each instance included: counters for positive words, counters for negative words, positive weight, negative weight, positive score, negative score, high score, negation counter and negation weight.

Then every feature will be evaluated by features selection through the Information Gain attribute evaluation and the Ranker search method for each feature so as to reduce the effort expended and the use of the data dimensionality, in addition to increasing the accuracy.

Figure 5 explains the workflow of our methodology in this research over the our three phases, in addition, important parts of our research the corpus of Arabic tweets, the Arabic lexicon and the machine learning algorithm.

The environment of every variety of processing and tools which we are using will be clarified in this chapter.

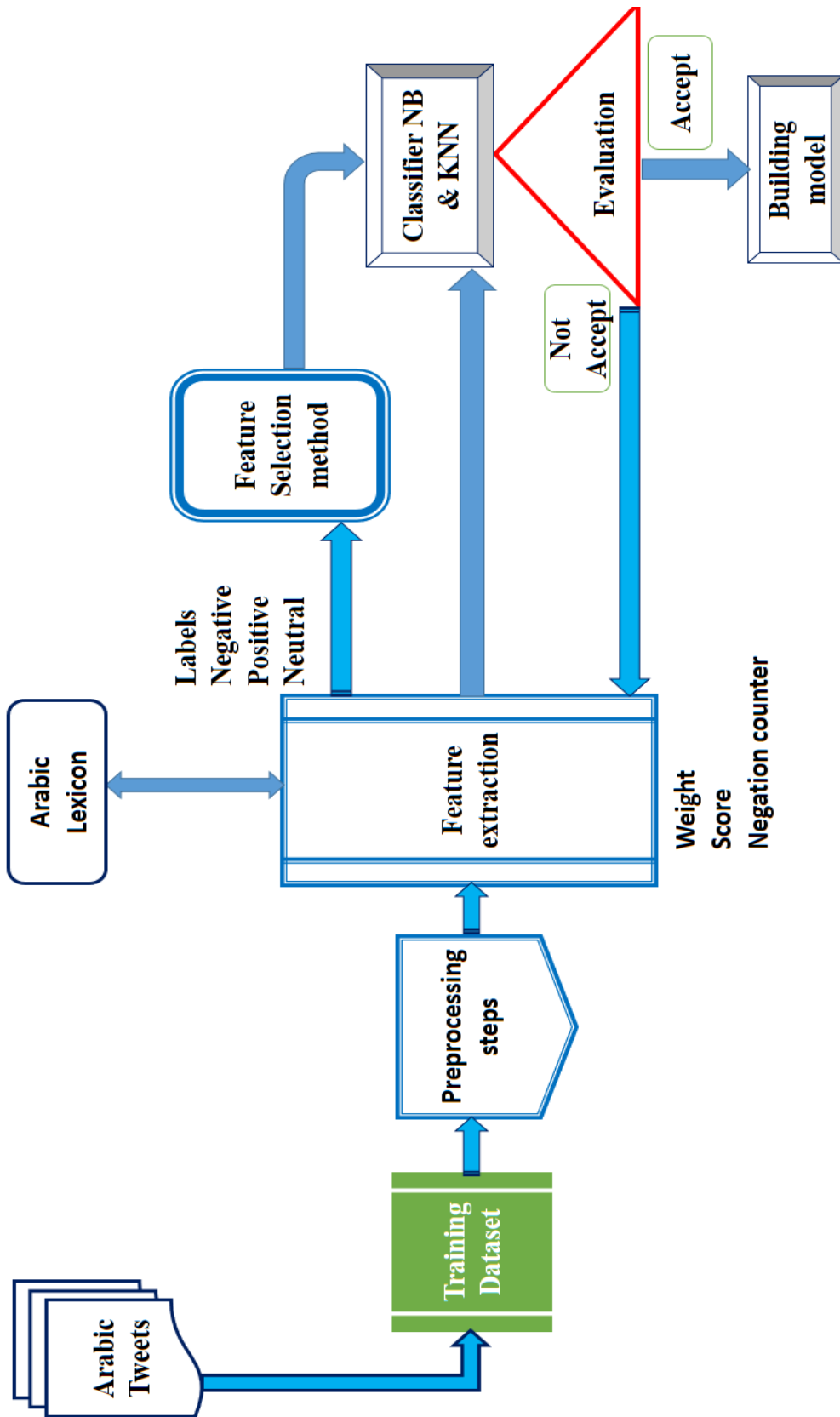


Figure 5 Workflow of our research through three Phases

3.2 Steps of Preprocessing

Preprocessing is a very important set of steps in the any methodology when analyzing text, especially when adopting a lexicon-based approach as we will be dealing with simplified word forms. Preprocessing has many steps and forms. We use the main steps on our dataset to improve the processing of the tweets, and the simplest method starts taking all words from instances, every single word is extracted and treated as separate token. In addition, the position of the words in sentence is ignored [23], which helps us to search through the lexicon for every word found in the tweets then, another calculation occurs, as shown in Figure 6, which explains the steps of preprocessing in our approach.

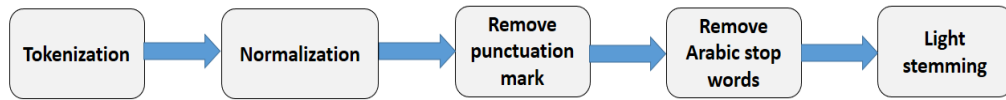


Figure 6 Steps of the preprocessing sequence

3.2.1 Tokenization

Tokenization is the first step in every preprocessing method, which depends on extracting every single word found in a text, followed by treating them as tokens separately. For example, the Arabic sentence ("نفرح عندم نرى ابتسامة الفقير") will be five tokens after applying this step as follows ("نفرح", "عندما", "نرى", "ابتسامة", "الفقير"); then for every word, it will apply another step of preprocessing

3.2.2 Normalization

The importance of this step in preprocessing comes from the Arabic language such that the state of the letter can be of a different shape and form. Furthermore, some

words in Arabic are written in letters with more than one shape depending on their context and position. One of the famous instances of writing is the Arabic letter ‘A’, which has several shapes ("أ", "آ", "إ", "إ"). Normalization will convert the four possible letters into one, which will be ("ا").

For example, ("السفر بالجو أحسن", "السفر بالجو أحسن", "السفر بالجو إحسن", "السفر بالجو أحسن") meaning “Travel by air is better,”

Implementing Arabic normalization for the underlined tokenization's gives the same word for each one in the four sentences, which is “احسن”.

Our processing will move then towards next step.

3.2.3 Remove Stop Words

Arabic stop words are words that do not carry opinion or sentiment; therefore, they are not important and they can always be removed. Reducing the number of words in the training dataset will help to reduce the effort expended on the treated words in the preprocessing, searching through the lexicon and classification processing. Sometimes, the user writes these words as style, but it does not carry any opinion or feeling. This step is implemented with a list of these words available in a public Arabic website. Unfortunately, some of these words are classified as negation words, which impact the extraction features and classifier performance [11]. Some limitations on our preprocessing occur because some words are combined between the Remove Stop Words list and the list of Negation words.

In the following, the list of remove stop words in English language does not carry sentiment. Therefore, these words do not benefit the sentiment analysis.

in, the, all, did, not, his, who, is, the, has, since, has, to, he, the, first, within, it, past, time, next, day, may, what, why, with, this, evening, one, added, add, but, she, he, has, her, confirmed, that, number, ten, year, when, when, the, year, Years, have, after, some, re, announced, because, even, if, then, f, that, this, or, and, any, zero.

In the same manner, we removed the Arabic stop words by determining them in the following Arabic list to explain them.

فى ، في ، كل ، له ، من ، هو ، هي ، كما ، لها ، منذ ، وقد ، هناك ، وقال ، وكان ، نهاية ، وقالت ، وكانت ، فيه ، كلم ، وفي ، وقف ، يوم ، فيها ، منها ، يكون ، يمكن ، حيث ، امس ، التي ، التي ، ايضا ، الاخيرة ، الذي ، الذي ، الان ، امام ، ايام ، خلال ، حوالى ، الذين ، بين ، ذلك ، حول ، حين ، الى ، انه ، اول ، ضمن ، انها ، جميع ، الماضي ، الوقت ، المقبل ، اليوم ، قد ، مع ، مساء ، هذا ، اضاف ، و اضافت ، فان ، قبل ، قال ، كان ، لدى ، نحو ، هذه ، وان ، واكد ، كانت ، و اوضح ، عشر ، عدد ، عدة ، عشرة ، عام ، عاما ، عن ، عند ، عندما ، على ، عليه ، عليها ، زيارة ، سنة ، سنوات ، تم ، بعد ، بعض ، اعادة ، اعلنت ، بسبب ، حتى ، اذا ، احد ، اثر ، غدا ، شخصا ، صباح ، اطار ، اخرى ، بان ، اجل ، بشكل ، حاليا .

3.2.4 Stemmer

The main objective of this technique is to reduce the real words found in a text to their shorter possible shape without losing their meaning. Due to the complexity of the Arabic language, stemming represents a significant approach in information retrieval and text mining systems carried out to address the word to its root [23]. One root will represent several shapes of words written in several suffixes and prefixes.

Importance of stemming in SSA come from adding more effectiveness to the search on the root of the words in the lexicon due to it adds extend recognize of the origin of existing words in order to determine subjectivity and sentiment analysis. It also provides an ability to identify the polarity of words that may be difficult to identify with their shape of written in tweets. While if we do not use stemming, some of the opinion words in the tweets will not be able to recognize them because there are additions to the suffix or prefix of those words comparison with these found in lexicon.

We use the Arabic stemming technique, which processes the Arabic word to acquire the root of a word. However, occasionally the root word does not relate to the meaning of the original words (e.g., for the stemmer for the Arabic word 'جمال' pronounced 'jamal' meaning 'beauty', the stemming for it is 'جمل' given the root here appearing as 'camel'). Losing the meaning of this sometimes causes limitations to the result of this technique or it does not benefit the classifier.

After finishing the implementation of the preprocessing steps within the Python programs, we moved to applying the extraction of new features which depend on the

words resulting from the preprocessing of tweets and on two defined lexicons, as will be shown in the following paragraph.

3.3 Feature Extraction

Features are functions with a bounded real valued extracted from the original dataset (in our research tweets). They are added to the original training dataset to improve and increase annotation of the dataset. We annotate the corpus by capturing a variety of features for the text of the tweets to improve the significance of the data.

Our processing depends on the principle that feature extraction is a process used to reduce the complexity of the context by transforming the original content into other features which are more significant. Not every feature has the same importance to the classifier process, while feature selection is a process used to reduce dimensionality by selecting a subset of the original dataset [24].

Enhancing the analysis and improving the classifier process of Arabic require that features be extracted from a text. Following to the author [12], when processing English text, we deal with the morphological richness of Arabic by reducing words to their roots and then execute the more considerable feature. Semantic features mostly deal with the meanings of words or phrases, such as constructing sets of strong subjectivity, weak subjectivity by adding polarity or by affecting intensity linking scores to words and phrases in the lexicon.

We do our processing with a different set of extracted features for semantics and statistics, which include positive word counters, negative word counters, high counts, states, positive weight, negative weight, score of positive, score of negative, high score, negation counters and negation weight.

3.3.1 Counters for Opinion Words Features

This feature extract from tweets depends on searching for each single tokenization of a tweet through the positive and negative lexicon. We modify the Arabic lexicon freely available on <http://www.macs.hw.ac.uk/~eaaar1/Eshrag%20Refaee/myResearch1.html>

by checking the polarity of the words found in this lexicon; remove the repetition words, and adding some new opinion words found in our training dataset. To become positive lexicon content on 1083 positive Arabic opinion words, and negative lexicon content on 2057 negative Arabic opinion words. We wrote the program in the Python language to implement these tasks with a positive counter to calculate the positive opinion words and a negative counter to calculate the negative opinion words in each instance. The counter adds one for each word found from the lexicon. Table 2 shows examples of the calculations of positive and negative words and comparisons of which number is larger to represent the state of the polarity.

3.3.2 Weight of Opinion Words Features

After extracting the positive counter and negative counter, it appeared to us that we needed to calculate the weight of the sentence according to words number of the tweet by dividing the counters by the words number of the tweet so that both the positive and negative weights can be calculated.

Figure 7 shows the steps of implementing the extracted features depending on semantics from two-list lexicon, one for the positive opinion words and another for negative opinion words to calculate the polarity of tweet. The state of polarity will be positive if the positive weight is greater than the negative weight, and it will be negative if the positive weight is less than the negative weight. In cases of equal weight, the tweets will have equal opinion words and are considered to be in a neutral state.

Table 3 Classification Examples by weight for positive and negative words

Arabic tweets	English meaning	Pos. weight	Neg. weight	Polarity
نفرح عندما نرى ابتسامة الفقير	We <u>rejoice</u> when see the poor man's <u>smile</u>	$2/5=0.4$	$0/5= 0$	Positive
الجو الممطر ممل و مزعج للقيادة	The rainy weather is <u>boring</u> and <u>tired</u> drive	$0/6= 0$	$2/6=0.33$	negative

We explain in Table 3 an example to calculate the weight of an Arabic sentence. We assume, after analyzing a sentence in the Python program, that the result contains two positive words and no negative words. We then assume that the extracted number of split words in this sentence is 5, the number of opinion words divided by the number of words found in the sentence. We obtain the weight of the positive and negative opinions for this sentence; then we compare them to determine the state of the polarity. According to the high percentage of opinion in this example, the positive percentage is 0.4, which is a positive weight. In addition, the negative weight will be zero, so the state of polarity will be positive.

Through our experiments, it appears to us that these features have more effect than the counters of opinion words on the performance measure; therefore, we will be depending on this in our research.

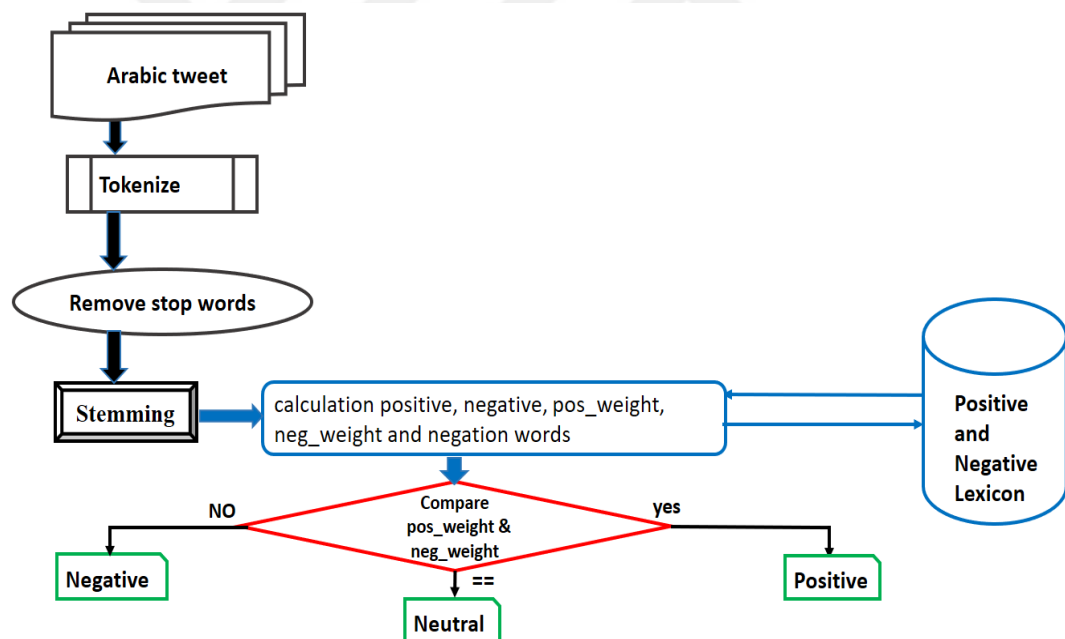


Figure 7 Workflow for extraction features with the first method

3.3.4 Score of Opinion Words Feature

Using this small Arabic lexicon available gratis online for research purposes from the authors [19] at <http://saifmohammad.com/WebPages/SCL.html#ATSL> provides an Arabic lexicon with 1366 positive and negative words, as more Arabic opinion words are being used on Twitter. The Score is a number which is a given value for the intensity score for each word found in this lexicon, which, according to the authors [19], is called the sentiment lexicon. The intensity score of the opinion words is the degree of positivity for the positive words in positive number and the intensity score for negativity with a negative number for negative words indicates the degree of each opinion word. For instance, the intensity score for “-1.000=ارهابي” (which means “-1.000 = Terrorist”) equals -1.000, which is the lowest value in the sentiment lexicon, where the intensity score for “0.963=مبروك” (which means “0.963 = Congratulations”) is the highest value in the sentiment lexicon.

Table 4 Classification examples of Arabic with scores of opinion words

Arabic tweets	English meaning	Pos. score	Neg. score	polarity
نفرح عندما نرى ابتسامة الفقير	We <u>rejoice</u> when we see the poor man's <u>smile</u>	0.887+	0	
		1.812	0	positive
الجو الممطر ملل و تعب للقيادة	The rainy weather is <u>boring</u> and <u>tired</u> drive	0	-0.65+	
		0	-1.363	negative

We explain in Table 4 an example to calculate the score of an Arabic sentence. We built a program in Python for this part of our research to apply the preprocessing steps and then look through the words comprising the tweets about the opinion words found in the sentiment lexicon. Next, will determine the intensity scores corresponding to those words and we then collect the positive scores together and the negative scores together for each tweet. The next step is to compare the positive score and the absolute

negative score. If the positive score is a higher result, the polarity state of the tweets is positive, and if the absolute negative score is higher, the result of the polarity state of the tweets is negative. If the words are not found in this lexicon, the intensity score will be zero and the polarity state is neutral. Figure (8) shows the steps our methodology to extract the intensity score features each tweet. We use the following equation in our tools [25].

$$\sum_{i=0}^m S_i \dots\dots\dots(3.1)$$

where m is total number of words found in each tweet and S_i is the intensity score for this word, wherein the result of the positive score compared with the absolute value of the negative score determines the polarity state of the instance.

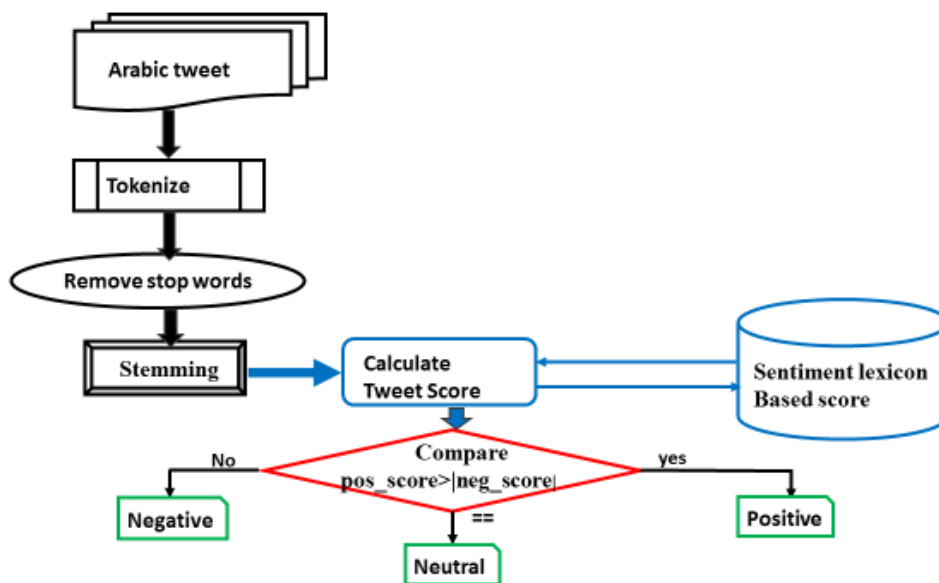


Figure 8 Workflow for extraction features with the second method

Through our experiments, it appears that the high score feature is more effective than the positive score and negative score together on the performance measure with our classifiers; therefore, we depend on it in our research.

3.3.5 Negation Counter and Negation Weight Features

The negation style is important to calculate, which causes a switch of the intended meaning of the opinion words due to the phrase being given an inverse meaning.

Negation in Arabic refers to using words that influence the meaning of a phrase by reversing the polarity state (sentiment) of the word coming after it, which is a phrase change sentiment from positivity into negativity, or if initially negative, from negativity into positivity. Moreover, although it is considered important, many Arabic studies and much research in this area avoid addressing them [17, 25].

Therefore, we employ our Python program to extract the number of Arabic negation words found in each tweet. We count them and calculate the negation weight of these words by dividing it with the number of words in the tweet, experiments proved influence it with two approach the weight feature and score feature through the treatment of the text. The authors [26] extracted features from the text by calculating the number of negation words and the negation weight.

Furthermore, negation of Arabic text is a property to be taken into consideration despite the difficulty of processing the text to achieve the correct extraction of the meaning to be delivered through this analysis. The user does not always using negation words to express an opinion as not every tweet contains negation.

In English the main negation words are “no, not, un” and in this research, we used the list below for the main words in Arabic negation.

ليس، ما، وما، لا، لم، لن، مش، ليست، لسناء، ليسا، ليستا، لسان لستم، لستن، ليسوا، لسن، دون، غير، مو،
عديم، بغير، عدم، بدون، بلا، ولا.

For example, tweet number 2703, which says

كلامك صحيح بس رأيي لا يقدم ولا يؤخر. منذ ان بدأ وانا أقول انه يفتقر لخطط ثلاث قبل وأثناء وبعد. لا "مجيب"

which means

"Your words are true, but my opinion does not offer and not delaying, I said since he started, he lacks three plans before, during and after. No answering. "

Which have three Arabic negation words, total number of words is 22, and negation weight is $3/22$ which equal 0.13636.

Through our experiments, the negation counter feature appeared to us to be more effective than the negation weight on the performance measure of our classifiers; therefore, we depended on it in our research.

3.4 Feature Selection

Generating and adding other features to the dataset makes it more significant. These are obtained by extracting them from the original tweets, after which a training dataset will be extended with a large amount of data. In this case, redundancy may occur. Additionally, classification may yield a complexity of a huge dimensionality of training dataset. Therefore, the purpose of feature selection is to find the best minimum subset of features to provide better accuracy and performance [24].

Therefore, there is the need to implement feature selection on the training dataset to transform it into reduce features. By ignoring the features that have no impact on the classification, the best performance is obtained by applying one of the feature selection methods to select a subset of relevant features for building our classification models.

Ranker methods rank attributes based on their individual evaluations. The behavior of a classifier will be impacted clearly by reducing overfitting, decreasing the time of training the dataset and increasing the classifier speed. In addition, by removing the low ranker features, the accuracy of the classifier will improve [21].

Moreover, feature selection techniques will provide a better understanding of the essential process that generates the data, by providing an individually a better definition of the features. Sometimes, these operations have options to select forwards and backwards for the trends of selection of features [27].

Feature selection methods, which select an optimal subset of features to reduce the data space dimensionality, help to reduce computational costs and improve classification performance. However, they did not garner sufficient attention to studies related to the Arabic language due to the principle of its work requiring many features to select between them.

In general, there are two categories of selection feature methods: filter and wrapper.

3.4.1 Filter with Information Gain Method

Generally, the filter method has several techniques to implement the task; one of these is Information Gain. This technique is implemented by calculations of information in partitions concerning the class when the presence of features is the only available information and corresponding to class distribution, such as instances with a class labelled as positive, which features corresponding information.

This technique is used to reduce the space of the data dimensionality and redundancy, which may be implemented during the preprocessing steps (Figure 9).

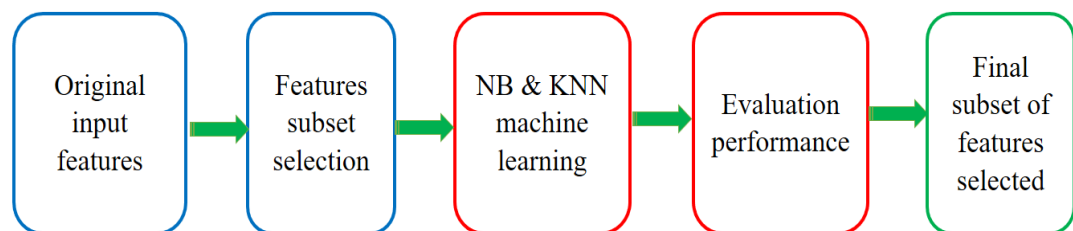


Figure 9 Workflow for feature selection with filter methods

In addition, the expected decrease in entropy (measure of the rate for transfer information) is measured in order to decide the importance of a given feature. An entropy function increases when the class distribution becomes sparser and it can be

recursively applied to find the subsets of entropy. The entropy function which satisfies these two requirements is given in the following equation:

$$H(D) = - \sum_{i=1}^c \frac{n_i}{n} \log(n_i/n) \dots \dots \dots (3.2)$$

where D is the dataset, n is a number of instances which are include in D , n_i is the members in class i and C equals the number of classes.

Furthermore, for entropy of subsets features is represented by the following equation:

$$H(D|x) = \sum_j \left(\frac{|x_j|}{n}\right) H(D|x = x_j) \dots \dots \dots (3.3)$$

Here, we have $H(D|x = x_j)$ which is the entropy correlated with the subset of instances which assumes a value of x_j for the feature x . For example, when feature x provides a good description of the class, the value which is associated with that feature assumes a low value of entropy in its class distribution.

Finally, the defined Information Gain as the reduction in entropy is as follows:

$$IG(x) = H(D) - H(D|X) \dots \dots \dots (3.4)$$

The result will be calculated according to the selection of a high value of IG for feature x . An experienced scientist will decide which limited low rank will be ignored and what will be kept and used in building the model [24].

3.4.2 Wrapper Methods

These methods depend on a machine learning algorithm to select the subsets of features depending on their predictive power as this method is considered to be a black box. Wrapper methods are remarkably simple and universal but occur at low speed compared to the filter methods, which are more expensive. In addition, they are unaffordable for the large number of features [27].

For the reasons above, we opted to use filter methods in our research. Figure 10 explains the general workflow of the wrapper method for the selection features approach.

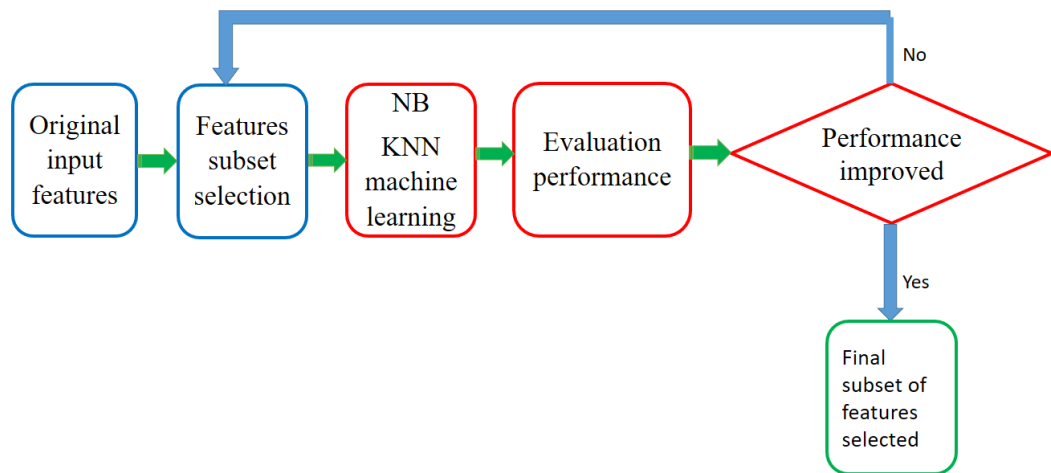


Figure 10 Workflow for feature selection with wrapper method

3.5 Dataset

Our dataset corpus was collected from Twitter using API Twitter (Application programming Interface), which is available at the special LREC repository <http://www.resourcebook.eu/shareyourlr/index.php>. The file format, ARFF (Attribute Relation File Format), is converted to CSV (Comma Separated Values) to simplify the addition and removal of features from the original dataset. The dataset contains 7390 Arabic tweets, which we subdivide into a training dataset comprising 6000 Arabic tweets to do our experiments in order to build a better model as well as a testing dataset comprising 250 unclassified Arabic tweets. Our better approach will execute it to the evaluation model. This dataset of the Arabic tweets corpus is annotated by 45 linguistically varied features relating to morphology, syntax, semantics and style, which is shared by the authors [15] for research purposes.

Our contribution will be the addition of more new features related to semantic linguistic analysis, followed by the evaluation all features to extract those with greater impact on classifier implementation in order to keep them. We ignore those that have little impact on classifier implementation.

3.6 Hardware Platform

Every process has been implemented on a Lenovo laptop operating Microsoft Windows 10 (64-bit) using a Core i5 processor and 4 GB RAM.

3.7 Natural Language Toolkit (NLTK)

The Natural Language Toolkit is a leading set of tools used for the analysis and application of the preprocessing steps in order to obtain the simple form of the meaning of the text via Python programs. Moreover, it is open source and freely available online. It can implement effectively for the preprocessing steps (tokenization, removing stop words and stemming) on the dataset³.

3.8 Python

We used Python version 3.6 (64 bit) which as downloaded from a freely available open source website⁴. Python also needs a special environment to process text. Therefore, we apply our Python programs with Anaconda manager environments, which comprise a variety of environments. After searching for a suitable environment, we settled on the Spyder environment through Anaconda to apply NLTK.

3.9 Anaconda Environments Manager

Anaconda Environments Manager is an open source platform for academic use which enables academic researchers to have access to the best Python tools available. This

³ <http://www.nltk.org/>

⁴ <https://www.python.org/downloads/release/python-360/>

tool enables users to pursue research by solving the complexities of mixing traditional infrastructure with modern approaches.⁵

In addition, it provides a simple and flexible free installer for all the basic packages needed to work with datasets of all shapes and sizes in the Python language. Through Anaconda, we found the appropriate environment to implement our Python programs; when using several other packages (without Anaconda), we experienced difficulty.

For example,

```
from nltk.stem.isri import ISRISemmer
```

The environments included in Anaconda include the:

1. Jupyter environment, which is used in interactive data science and scientific computing for datasets using the Python language.
2. Spyder environment, which is one of the simplest and most useful environments provided by Anaconda. This environment is a scientific development environment with the Python language, and it enables us to work with different packages according the specification of the operating system and to implement our programs efficiently.

3.10 Arabic Natural Language Processing

Recently, Arabic Natural Language Processing has gained increasing importance, and several modern systems have been developed for a wide range of applications. Some have been successful, and others are still in development and need to more work. These applications deal with a number of complex problems relating to the nature and structure of the Arabic language, including machine translation, information retrieval and extraction and many other related issues.

Arabic is written and read from right to left, but numbers are written and read from left to right. Additionally, Arabic text has neither capitalization nor strict punctuation rules; thus, the absence of these factors creates much difficulty in recognizing Arabic

⁵ <https://anaconda.org/>

sentence boundaries in the phase of preprocessing [28]. For example, the Arabic sentence “Algeria became independent in 1962 after 132 years of French occupation.”

استقلت الجزائر في سنة 1962 بعد 132 عاما من الاحتلال الفرنسي.



3.11 WEKA

It is one of the best platforms for data mining [15] and one of the leading tools, which has a significant practical Graphical User Interface (GUI), this package is open source for data mining solutions. WEKA allows access to a variety processing steps such as supervised and unsupervised filtering, feature selection methods and machine learning algorithms.⁶

Another one of the WEKA features is the ability to apply different operations on a dataset, such as adding or removing attributes and instances. By using these, we are able to conduct a number of experiments to select the appropriate approach according the outcomes. It is a Java-based data mining library and environment, and it processes with multi files format such as ARFF, CSV and other file types.

In addition, it provides tools to divide our dataset into a training dataset and a testing dataset with a variety of filters and techniques via Resample, Remove Percentage and Remove Range. Furthermore, it provides a number of training test options, such as cross validation and a percentage split of the training dataset with assistant parameters for a collection of machine learning algorithms for data mining tasks, such as NB, KNN and others [23]. All these options are ready to do our experiments.

⁶ <http://machinelearningmastery.com/what-is-the-weka-machine-learning-workbench/>

3.12 Supervised Learning Algorithms

It is a classifier algorithm training on classified instances previously labelled manually or automatically. The aim is to build a more accurate model according to our dataset. New unclassified instances do not need to search again through the lexicon to determine the polarity of the sentences. Several kinds of algorithms (but not all of them) are suitable for our tasks (Arabic language, large size of instances, large data dimensionality); therefore, we selected two algorithms which are quite different, but each has been shown to be effective in text classification in previous studies, namely the Naive Bayes classification and the K-Nearest Neighbors Classifier [11].

3.12.1 Naïve Bayes Classifier

The basic idea of the Naïve Bayes Classifier (NB) is to find the probabilities of classes according to the given instances using the joint probabilities of words and each class. It is based on the assumption of word independence. It starts calculations with conditional probabilities for given instances through the class. Furthermore, the probability of each of its attributes occur in a given class that is independent [29]. Figure 11 shows that this NB classifier depending on two probabilities kinds of words found in sentences and values found in features which effects on the class label.

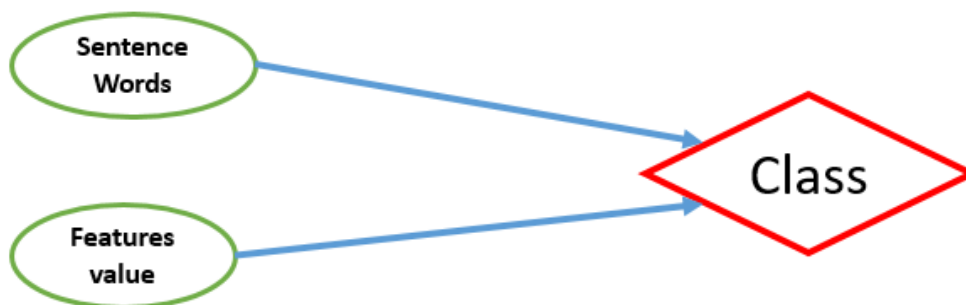


Figure 11 Probability of NB classifier effects on the class label

We confirm previous studies that widely use NB, which often provide good results, and benefit from the easy probabilistic interpretation of results.

For example, instance content 6 words “*This car is beautiful and powerful.*”

$w_i = \{w_1, w_2, \dots, w_6\}$ and C is the class (positive), calculating the probability for affecting every word in w_i on the class C leading classifier considering for the two words $w_4 = \text{“beautiful”}$ and $w_6 = \text{“powerful”}$.

In addition, calculating the probability for affecting every feature, weight feature will be equal to 0.33 leading classifier relate this value with class label.

3.12.2 K-Nearest Neighbors Classifier

K-Nearest Neighbors (KNN) is considered to be one of the simplest and most effective classification algorithms in use. This classifier is based on the class of new instances which are likely to be that for the majority being similar to their closest neighbors. K is the distance function between the attributes space, which is specified as a parameter in the algorithm on the dataset [29].

Generally, KNN is used in statistical estimations and pattern recognition during the training phase. It is known as a lazy learner algorithm that stores all available data points and classifies new instances based on a distance function where the algorithm simply stores those data points, including their class labels. All computation is deferred until the classification process.

For example, “*Ali likes iPhone with quiet sound*”, find the most similar sentences in training dataset (have the same words like *iPhone, quiet*), and classify it according majority class.

Figure 12 shows that this KNN has a boundary according to the k value with x_1 and x_2 as two features. Given the example for classification when $k = 3$, it searches for the nearest three classes and determines the classification of the green star to class B (positive), while selecting $k = 6$, the classification will be class A (negative). This classifier takes more time to classify new examples, but it is needed to calculate and

compare distances from new examples to all other examples. Finally, selecting k may be problematic and it needs a large number of samples for accuracy.

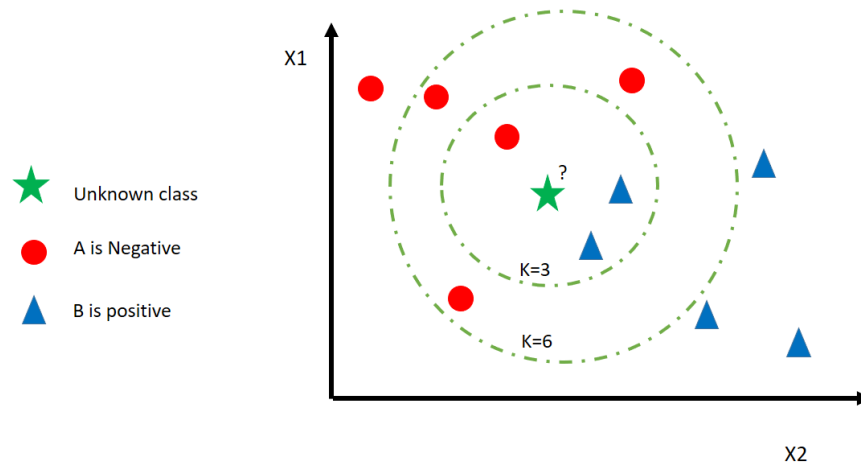


Figure 12 Boundaries of KNN classifier effects on the class label

3.13 Cross validation

Validation is an important step that allows us to test the accuracy of algorithms. The most common approaches to validation is the cross validation method. This method, by comparison, splits the data into testing and training; however, the data is scanned several times and each division, or part of the data, becomes accustomed in the training and testing phases.

To clarify, we use the ten-fold cross validation method wherein the data is divided into ten divisions or parts, one of which is used for testing and the other nine being used for training in the initial iteration. In the second run, a different part is used for testing and for training the other nine parts are used, including the one that was used for testing in the first run. The runs continue until each part or division is given the roles of training the dataset and the testing the dataset. The final accuracy is the average of the accuracies obtained in the ten runs [30].

In our research, we have used the ten-fold cross validation and carried out all our experiments through cross validation in order to give more power for building the model.

3.14 Performance measures

One of the most important issues is the correct evaluation of the learned model, helping us to decide which model is better than other models. The measures make the task easier, especially on the theoretical level. However, with the classification report, there are many measures appearing as the output, some of which without a clearly justified theoretical basis [31].

Table 5 Results report for score task with Naïve Bayes classifier

Stratified cross-validation							
Summary of the result							
	No. of instances			Percentage			
Correctly classified instances	5614			93.5667%			
Incorrectly classified instances	386			6.4333%			
Kappa statistic	0.8928						
Mean absolute error	0.0461						
Root mean squared error	0.1637						
Relative absolute error	12.0329%						
Root relative squared error	37.4123%						
Total Number of instances	6000						
Detailed accuracy by class							
	TP rate	FP rate	Precision	Recall	F-measure	Roc area	Class
	0.998	0	1	0.998	0.999	1	Neutral
	0.89	0.003	0.998	0.89	0.941	0.999	Positive
	0.992	0.074	0.698	0.992	0.82	0.999	Negative
Weighted average	0.936	0.012	0.954	0.936	0.939	0.999	

We explain through the list in the Table 5, example for variety numbers with one of results summary for one of our task (score) with NB classifier.

In this study, to evaluate and measure the effectiveness of classifiers based on our proposed features, we follow literatures [11, 14, 21, 25, and 29] by depending on Precision, Recall, F-measure and Accuracy, which are widely used as performance measures for classifiers. We documented the results and compare the approaches for building the better model, as shown in Figure 13, which explains the steps in the evaluation of our tasks.

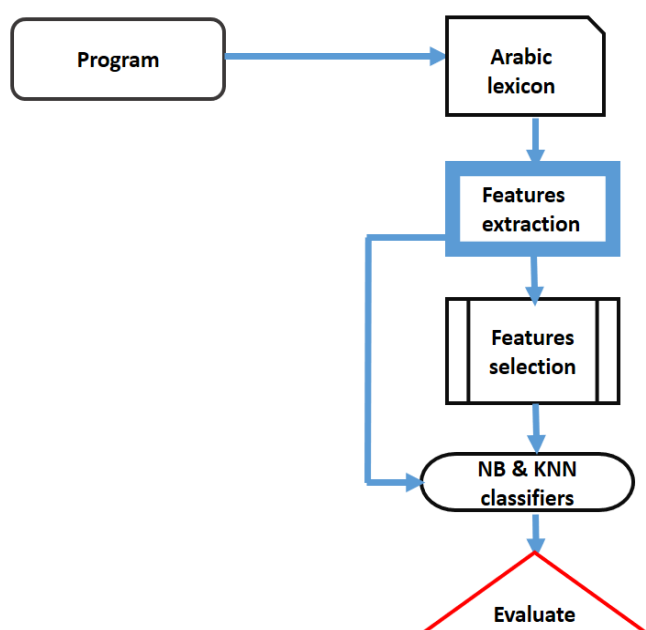


Figure 13 Methodology for evaluation in our tasks

3.14.1 Precision

Precision represents how many of the returned values are correct; for example, recall that class r is a ratio of the number of instances classified correctly as class r relative to the total number of instances, as shown in the following equation:

$$\text{Precision}(r) = \frac{TP}{TP+FP} \dots\dots\dots(3.5)$$

where TP is equal to the number of True Positives and FP is equal to the number of False Positives.

3.14.2 Recall

Recall is the number of positives that the model returns, or the number of truly relevant results; for example, the recall for class r is a ratio of the number of instances classified correctly as class r relative to the total number of correctly classified instances, as shown in the following equation:

$$\text{Recall}(r) = \frac{TP}{TP+FN} \dots\dots\dots(3.6)$$

where TP is True Positive and FN is False Negative.

3.14.3 F-measure

F-measure is a measure of the accuracy of a test by using the results from equations (3.5) and (3.6) which are defined as the weighted harmonic mean of the precision and recall of the test done on the dataset by the classifier, as shown in the following equation:

$$\text{F-measure} = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \dots\dots\dots(3.7)$$

Which represents much of study on the metrics of accuracy; in addition, the important in case of our study is the correctness of the classification of instances in tweets, which is a measure of the model's success, this implement by the next measure.

3.14.4 Accuracy

The accuracy is for the measure of the classifier performance as a ratio between the number of correctly classified instances to the total number of instances. We follow the studies which represent this scale as a base for comparison between the tasks because we are interested in correct classified instances. The results here are calculated according to the following equation:

$$\text{Accuracy} = \frac{\text{No. of correct classified instances}}{\text{Total No. of instances}} \dots\dots\dots(3.8)$$



CHAPTER 4

RESULTS AND DISCUSSION

4.1 Introduction

In our research, we aim to explain and clarify how to build a model to achieve the requirements of sentiment analysis relating to two issues: the large numbers of tweets and the Arabic language, depending on a lexicon-based approach in order to inform and enhance better decision-making. To implement this, we need to evaluate our processing in every phase and in every task. Therefore, we found the evaluation for the NB and KNN classifiers performance in two phases:

1. Features extracted
2. Features selected

Figure 13 shows the workflow for the evaluations.

In this chapter, we explain the outcomes for the sequences of our three phases being applied to determine the high accuracy for classifier performance, which represents creating a more suitable model.

4.2 Apply Preprocessing Steps

The purpose of the initialization preprocessing is to obtain a simple formula for important words. These words may be loaded with sentiments to take advantage of them before entering them in the search in the lexicon in addition to removing unbeneficial words and characters.

At the beginning of our research and work, we started by building a Python program to implement the preprocessing steps. We applied the initialization preprocess to

separate one sentence (tweet) from their words. After the Python program succeeded with its tasks, we modified it so it can read a file containing thousands of tweets.

In the following list, we explain example of the implementation of the program about splitting the tweets into all the words that comprise these tweets. This list is part of our dataset.

```
words of Arabic tweets are ['ناس', 'كثير', 'تعمرو', 'اجيانك', 'بس', 'مايلت', 'نظرك', 'غير', 'اشخاص', 'قليل', 'لكن', 'على', 'قلوبهم', 'الا', 'انهم', 'فادين', 'يحطوا', 'lat', '']
words of Arabic tweets are ['مات', 'لانها', 'lat', 'ا', 'كل', 'محبة', 'زالت', 'ال', 'انها', 'غير', 'الله', 'وكل', 'طاعة', 'لم', 'تقبل', 'لانها', 'غير', 'الله', 'وكل', 'lat', '']
words of Arabic tweets are ['هدية', 'بدون', 'مناسبة', 'كلمة', 'حلوة', 'تي', 'وقت', 'عشوائي', 'الانتسامة', 'وسط', 'طريق', 'مزحوم', 'دعوة', 'لك', 'تي', 'ظهور', 'lat', '']
words of Arabic tweets are ['الخاصين', 'الونز', 'ابن', 'التائبين', 'الونز', 'ضياء', 'تي', 'قلوب', 'الصادقين', 'اللهم', 'lat', 'الونز', 'انيس', 'المؤمنين', 'الونز', 'lat', '']
words of Arabic tweets are ['', 'lat', '']
words of Arabic tweets are ['', 'lat', '']
words of Arabic tweets are ['', 'lat', '']
words of Arabic tweets are ['', 'lat', '']
words of Arabic tweets are ['', 'lat', '']
words of Arabic tweets are ['', 'lat', '']
words of Arabic tweets are ['', 'lat', '']
words of Arabic tweets are ['', 'lat', '']
words of Arabic tweets are ['', 'lat', '']
words of Arabic tweets are ['', 'lat', '']
words of Arabic tweets are ['', 'lat', '']
words of Arabic tweets are ['', 'lat', '']
words of Arabic tweets are ['', 'lat', '']
words of Arabic tweets are ['', 'lat', '']
words of Arabic tweets are ['', 'lat', '']
words of Arabic tweets are ['', 'lat', '']
words of Arabic tweets are ['', 'lat', '']
words of Arabic tweets are ['', 'lat', '']
```

Then we applied the removal of the punctuation marks by using the following punctuation list.

```
' ! ( ) - [ ] { } ; : ‘ “ \ , < > . / ? @ # $ % ^ & * _ ~
```

In same way we remove the Arabic stop words, according to a list of Arabic stop words, which we explained previously in Chapter 3.

Finally, the ending of the preprocessing steps with a light stemmer for the Arabic language is provided with NLTK.

By implementing the preprocessing steps, we are now ready to seek the opinion words to extract the features that have the information about each word in each tweet relating to the Arabic lexicon and via semantic and statistical as morphology types.

We applied the first Python program in first method to store the outcomes to a csv file as matrices representing the words that the program looked at through the lexicons. Then it can be used by adding it to our original dataset to evaluate the classifier performance measures. In the following Table, we show the outcomes for this program that extracted the new features depending on the Arabic lexicon with two split lists.

Table 6 Outcomes of first method with lexicon has two split lists

1	positive_count	negative_count	High_count	state	pos_weight	neg_weight	negator_count	negation_weight	pos_words	neg_words	negator_words
2	0	2	2	negative	0	0.18181818	0	0	[]	[تكر، اخبت]	[]
3	1	1	1	negative	0.11111111	0.11111111	0	0	[جمل]	[طيبا]	[]
4	1	1	1	negative	0.07142857	0.07142857	1	0.07142857	[ال]	[امسل]	يا
5	2	1	2	positive	0.14285714	0.07142857	0	0	[احد، شكر]	[احظ]	[]
6	2	2	2	negative	0.15384615	0.15384615	0	0	[اود، يا]	[اوى، اندم]	[]
7	1	2	2	negative	0.07142857	0.14285714	1	0.07142857	[وما]	[انتك، نخل]	وما
8	1	1	1	negative	0.06666667	0.06666667	1	0.06666667	[طيب]	[ارعل]	ما
9	2	1	2	positive	0.15384615	0.07692308	0	0	[شكر، نكر]	[الحب]	[]
10	5	0	5	positive	0.27777778	0	0	0	[تهيد، عزز، طيب، يبي، صدق]	[]	[]
11	1	1	1	negative	0.05555556	0.05555556	0	0	[اني]	[صدم]	[]
12	2	1	2	positive	0.10526316	0.05263158	2	0.10526316	[اليم، لهم]	[ارو]	لا
13	3	1	3	positive	0.16666667	0.05555556	0	0	[ترض، اصلاح، زدي]	[جبر]	[]
14	1	2	2	negative	0.04545455	0.09090909	1	0.04545455	[تخص]	[الحا، ورج]	لم
15	1	2	2	negative	0.04166667	0.08333333	0	0	[زحد]	[كبه، نخل]	[]
16	2	1	2	positive	0.08	0.04	1	0.04	[السي، اخل]	[كبه]	ما
17	5	1	5	positive	0.23809524	0.04761905	0	0	[ياد، اخو، رجم، ال]	[ال]	[]
18	4	1	4	positive	0.21052632	0.05263158	1	0.05263158	[ارع، الم، تنيب، تنيب]	[احرب]	لم
19	3	1	3	positive	0.125	0.04166667	0	0	[تصو، بيك، احص]	[ازير]	[]
20	2	2	2	negative	0.11764706	0.11764706	0	0	[احضر، اظبا]	[ندخل، قف]	[]
21	5	2	5	positive	0.2631579	0.10526316	0	0	[صدق، اصلاح، برح، ياد، ال]	[صناع، اقم]	[]
22	4	0	4	positive	0.18181818	0	0	0	[رجم، ال، جمل، رجم]	[]	[]
23	7	2	7	positive	0.30434783	0.08695652	0	0	[احظ، اخل]	[نخل، لودي]	[]
24	4	4	4	negative	0.18181818	0.18181818	1	0.04545455	[اقرى، ستر، يرقا، ال]	[الن، كلم، الن، قنصر]	لا
25	3	1	3	positive	0.16666667	0.05555556	0	0	[ظني، ال، ملي]	[مطلق]	[]
26	5	2	5	positive	0.22727273	0.09090909	0	0	[احد، ال، قطع، نكر، حيب]	[السر، كبه]	[]
27	3	1	3	positive	0.13043478	0.04347826	1	0.04347826	[احص، ال، ال]	[كبه]	مو
28	2	1	2	positive	0.0952381	0.04761905	2	0.095238095	[جمل، ستر]	[قنصر]	لا
29	4	0	4	positive	0.2	0	0	0	[الوز، العوز، اعمل، اصلاح]	[]	[]

We applied the second Python program in second method to store the outcomes. In the following Table, we show the result for the second Python program to extract new features depending on the Arabic lexicon having intensity scores for each words found in it.

Table 7 Outcomes of second method with lexicon has intensity scores

1	positive_score	negative_score	negation_weight	negation_count	High_score	state
2	0	-0.83	0.181818182	0	0	-0.83 negative
3	0.01	0	0.111111111	0	0	0.01 positive
4	0	0	0.071428571	1	0	0 neutral
5	1.538	0	0.071428571	0	0	1.538 positive
6	0	-0.438	0.153846154	0	0	-0.438 negative
7	0	-0.588	0.142857143	1	1	-0.588 negative
8	1.112	-0.625	0.066666667	0	1	1.112 positive
9	1.188	0	0.076923077	0	0	1.188 positive
10	1.725	0	0	0	0	1.725 positive
11	0	0	0.055555556	0	0	0 neutral
12	0	0	0.052631579	2	0	0 neutral
13	0.01	0	0.055555556	0	0	0.01 positive
14	0.388	0	0.090909091	1	1	0.388 positive
15	1.412	0	0.083333333	0	0	1.412 positive
16	0	-0.35	0.04	1	1	-0.35 negative
17	0.8	0	0.047619048	0	0	0.8 positive
18	0.526	-1.4	0.052631579	1	1	-1.4 negative
19	0.725	0	0.041666667	0	0	0.725 positive
20	0.287	0	0.117647059	0	0	0.287 positive
21	0.8	-0.463	0.105263158	0	0	0.8 positive
22	0.7	0	0	0	0	0.7 positive
23	0.1	0	0.086956522	0	0	0.1 positive
24	0.449	0	0.181818182	1	1	0.449 positive
25	0.963	0	0.055555556	0	0	0.963 positive
26	0.6	0	0.090909091	0	0	0.6 positive
27	0.475	0	0.043478261	1	1	0.475 positive
28	0.275	-0.225	0.047619048	2	2	0.275 positive
29	0.676	0	0	0	0	0.676 positive

4.3 Display Feature Extraction Result

Extract 7 has new features relating to statistical and semantic orientation by implementing the Python program dealing with two split lexicon files, (See Table 1 in Chapter 1). Then, we compare the selection using the features which are more impactful on classifier performance in order to build our highly accurate model. We found that the NB classifier performed better than the KNN classifier according to the added new features of (positive weight and negative weight with the negation counter) with an accuracy of 88.21 in this phase, which is more suitable for our aim of building a better model.

While implementing the second Python program with the lexicon with intensity scores for the opinion words, we extracted five features, as shown in Table 1.

When doing our experiments to select the features which are more impactful on classifiers, we started with the NB classifier and the performance measure. The results are reported in Table 8, which shows the performance measures for training the dataset after adding the features of weight, scores and negation.

All the experiments and tasks were applied under condition cross validation 10.

Table 8 NB performance measures with feature extraction phase

Tasks execute cross valid. 10	Precision %	Recall %	F-measure %	Accuracy %
NB classifier with original dataset	64.4	44.4	48.6	44.4
NB classifier after adding Weight features	86	85.8	85.8	85.83
NB classifier after adding Weight & Negation features	88.3	88.2	88.2	88.21
NB classifier after adding Score feature	81.2	74.3	73.7	74.33
NB classifier after adding Score & Negation features	84.3	80.8	80.9	80.8

We then conducted our experiments with the same features through the other selected classifier (KNN), and similarly, we report the performance measures for this classifier, as shown in Table 9, which shows that there is an improvement in the performance of KNN, but fewer benefits than our other classifier.

Table 9 KNN performance measures with Feature extraction phase

Tasks execute cross valid. 10	Precision %	Recall %	F-measure %	Accuracy %
KNN classifier original dataset	54.1	54.9	54.4	54.89
KNN classifier after adding Weight features	56.1	59.1	56.8	59.13
KNN classifier after adding Weight & Negation features	67.2	67.6	66.6	67.6
KNN classifier after adding Score feature	56.1	57.6	55.3	57.6
KNN classifier after adding Score & Negation features	54.3	56.5	53.8	56.53

The following chart in Figure 14 shows the comparisons between our two classifiers when using the features extracted in our tasks. They show an improvement in the performance of the NB classifier when adding more features more than KNN, but there are fewer benefits for the KNN classifier.

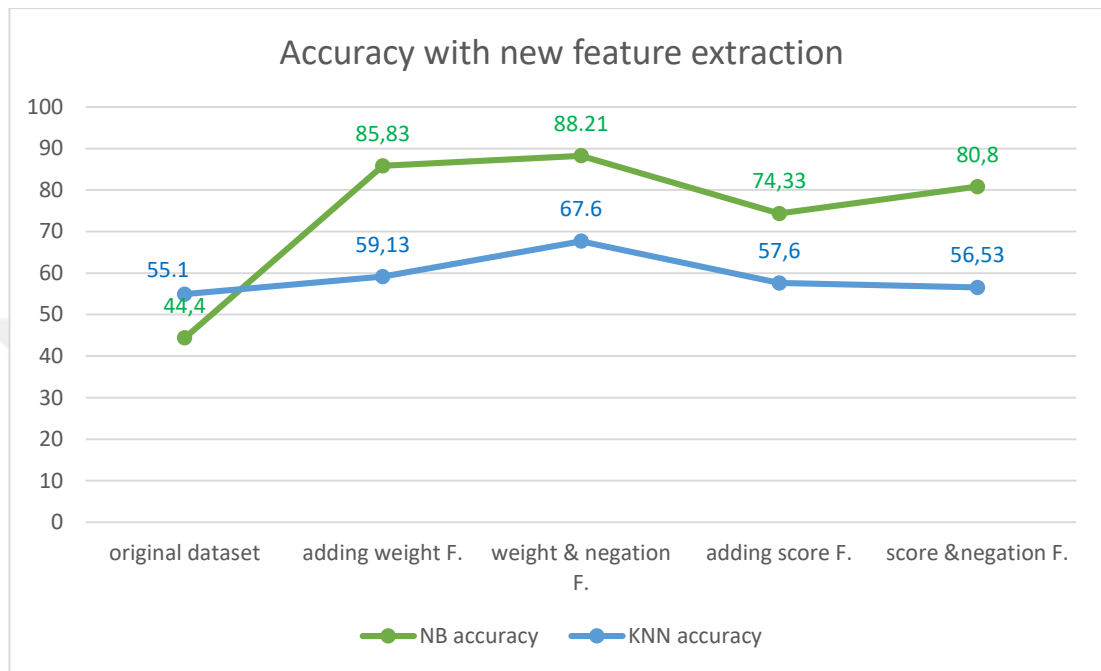


Figure 14 Comparison NB and KNN through tasks in feature extraction

Green line for NB. It is clear that the classifier took advantage of the weight features; the accuracy is improved to become 85.83, and then improved again when we added the negation feature to reach 88.21. While in the case of the score features, the benefit of the classifier reached an accuracy of 74.33 and when adding the negation feature, the accuracy improved to become 80.8.

Generally, the NB classifier benefitted from the weight and negation features while there was less benefit with score and negation features.

Blue line for KNN. Here, the classifier benefitted less from the weight features and the accuracy improved to become 59.13. It then improved when we added the negation feature to reach 67.6. In the case of score features, there was less benefit for the

classifier, with an accuracy of 57.6, and when adding the negation feature, it was 56.53. Generally, the KNN classifier benefitted from the weight features and negation, and it had less benefit with score and negation features.

In this phase according performance measures, NB with weight and negation features proved to be more accurate.

4.3 Display Feature Selection Result

After we extract new statistical and semantic orientation features from our dataset, we sometimes observe features information with two values – “False” or “True” – which does not help the classifier learning to predict the outcomes or the correlations with the label of class.

Therefore, we apply our experiments to know which features are useful and which are not useful or limited in terms of benefits for classification by using the feature selection methods. The philosophy is to remove them due to the fact that these features will not help us to build a better model for Arabic sentiment analysis.

Furthermore, we implement the feature selection method after the classifiers complete their tasks with feature extraction and we explain their results and their impact, and then by determining which features are more impactful on the performance of the classifiers through Information Gain attribute evaluation and Ranker search methods.

By selecting the features with high values through the ranking and ignoring those with low-value rankings, Table 10 shows the old and new numbers of features for our approaches before and after using the feature selection method. We then describe the effect the number of these features on classifier performance.

Table 10 Number of features through feature selection phase

Tasks	No. Features before F.S.	No. Features after F.S.	No. Features removed
Original dataset	45	24	21
Dataset with Weight features	47	16	31
Dataset with Weight & Negation features	48	17	31
Dataset with Score feature	46	15	31
Dataset with Score & Negation features	47	16	31

We removed the features that show low value through the ranking, particularly 21 features from the original dataset, and we removed 31 features from every approach, which were the same features from every dataset. Then, we conducted our experiment with the NB classifier, the results of which yielding high values of accuracy with the approach of the score features of 93.56, as shown in Table 11.

Table 11 NB performance measures with feature selection phase

Tasks execute cross valid. 10	Precision %	Recall %	F-measure %	Accuracy %
NB classifier task F.S. on original dataset	64.6	44.5	48.8	44.45
NB classifier task Weight feature after F.S.	89.9	89.9	89.9	89.85
NB classifier task Weight & Negation features after F.S.	89.9	89.9	89.9	89.85
NB classifier task Score feature after F.S.	95.4	93.6	93.9	93.56
NB classifier task Score & Negation features after F.S.	91.1	89.1	89.3	89.08

Then we apply our experiments with the same features through the other selected classifier, that is the KNN selector, and in the same manner, we report the performance measures for this classifier, as shown in Table 12.

Table 12 KNN performance measures with feature selection phase

Tasks execute cross valid. 10	Precision %	Recall %	F-measure %	Accuracy %
KNN classifier task F.S. on original dataset	57.5	61	58.1	61.03
KNN classifier task Weight features after F.S.	92.6	92.5	92.4	92.51
KNN classifier task Weight & Negation features after F.S.	92	91.9	91.8	91.93
KNN classifier task Score feature after F.S.	92.5	91.7	91.8	91.71
KNN classifier task Score & Negation features after F.S.	90.8	90	90.1	90

Generally, the KNN classifier gains much benefit from the F.S. which appears clearly in all results, and which is good accuracy in all tasks.

Figure 15 shows in this phase the performance measures for NB with task score features proving to have higher accuracy at 93.56.

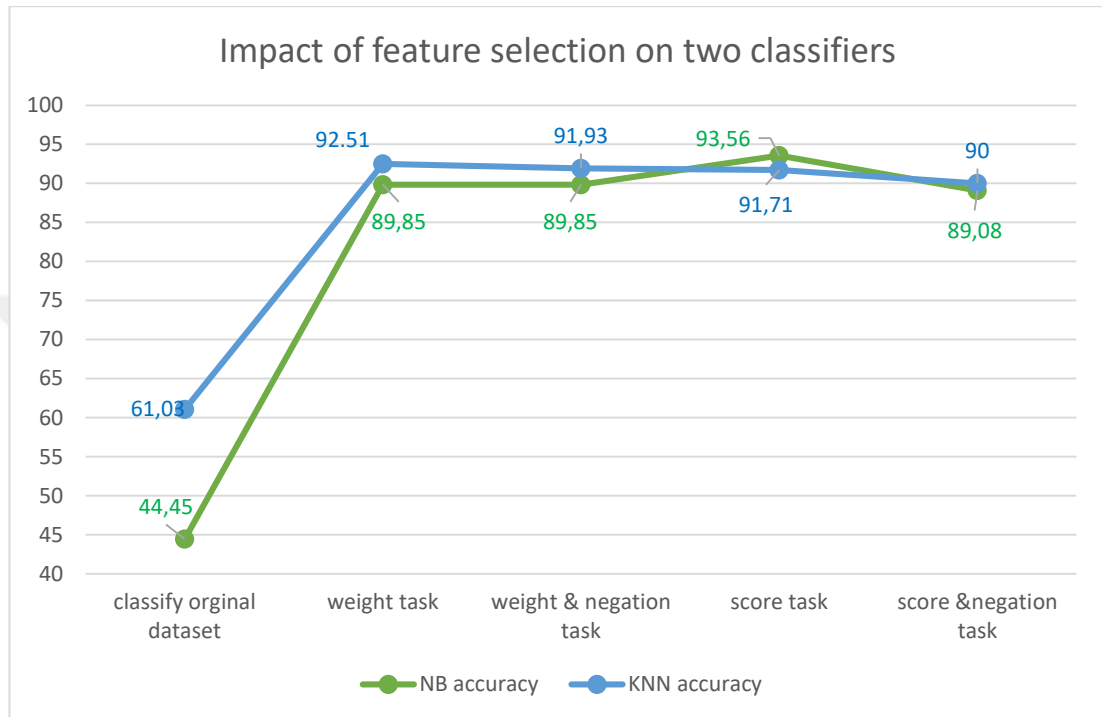


Figure 15 Comparison of NB and KNN through tasks in feature selection

Green line for NB, This classifier takes advantage of F.S. and the accuracy improves to become 89.85, but stays at the same percentage of improvement when we added the negation feature to reach 89.85. While in the case of the score features, it is clear that the classifier benefits from F.S., and the accuracy become 93.56, while adding negation features in this task through the F.S. phase does not have a positive impact on the accuracy measure to become 89.08.

Generally, the NB classifier benefits from F.S., especially with the score features task while there is less benefit with the weight and negation features.

Blue line for KNN, In this phase, the results explain clearly that the classifier benefits from the F.S. in every task. With weight features, the accuracy is high and improves to become 92.51; then the accuracy is less improved when we added the negation feature to reach 91.93, while in the case of the score features, the classifier with accuracy reached 91.71, but when adding the negation feature, it more or less became 90.00.

Our results explain the big role for using lexicon in classification for Arabic tweets, that we getting 88,21 depending on NB classifier, this result through the task of adding weight and negation features for training dataset. After implement F.S method we getting improve for two of our classifiers, generally KNN is more benefit from this phase and the performance measures becomes 92,51 with weight task, but still accuracy value with NB classifier getting high with 93,56.

CHAPTER 5

CONCLUSION

5.1 Limitations

For our research, we have found a number of limitations regarding Arabic natural language processing. For example, some words are combined between the remove stop words list and the list of negation words, thereby causing the program occasionally to be unable to recognize negation sentences, (e.g. ‘لا’ pronounced ‘la,’ meaning ‘not’).

Another one of the limitations of the Arabic language is that it does not have a lexicon for opinion words covering a large number of terms. The basic opinion words or opinion phrases in the lexicon are insufficient to determine the polarity to which group these tweets belong. Here, if the words in the sentence do not belong to the lexicon, then the tweet will be considered to be neutral. In this research, we deal with two lexicons, both of which need to be modified and which do not represent every Arabic opinion word.

One more limitation with regard to using the stemming process is the processing of the word to obtain the root of the word. However, sometimes the root word does not relate to the meaning of the original word (e.g. the stemmer for the Arabic word ‘جمال’ pronounced ‘*jamal*,’ meaning ‘*beauty*’, the stemming for which is ‘جمل’ given the root here appearing as ‘*camel*’).

5.2 Research Conclusions Illustrated

Achieving several aims from this research, one of the important aims is to explain to decision makers and researchers the approach of dealing with complex Arabic and extensive language by building a model that depends on three phases: preprocessing,

features extraction and features selection methods, and carrying this out through appropriate classifiers to achieve high performance measures.

We demonstrated in this research the important role for our preprocessing steps by analyzing and dividing Arabic tweets into useful words through a classification process. We applied our approach of preprocessing in several steps: tokenization, removing punctuation marks, removing stop words and stemming. We found that this approach of preprocessing would be of benefit for similar thousands of Arabic instances with a good impact on classifier performance, which is the answer to Research Question No. 1.

We answering Research Questions No. 2 and No. 5 and we proved that extracted features depended on two types of Arabic lexicon being clearly useful. Moreover, we found that using NB and KNN classifier algorithms is appropriate for a large Arabic corpus dataset, from which we obtained a 93.56 accuracy with the score task for the NB classifier.

We apply one of our aims about contributing to increasing the research area of limited resources for the Arabic corpus, which suffer from a paucity of research in comparison to the other main languages. A huge amount of Arabic information enters at all times, so different solutions are needed for each domain and for each genre (business, politics, sport, education health and so on).

We answered Research Question No. 3 explaining the optimization of the outcomes and enhancement of the performance measures by using the features extracted. These carry information for the words found in tweets and leading the classifiers will benefit from dealing with these features by linking them with labels of class, including positive weight, negative weight, negation counter, negative score, positive score, and high score. We found the important role for statistical and semantic features as a type and the information that is carried with them on the classifier performance measures.

In addition, we answer the question about quantity and type of features extraction. We found and demonstrated that not every feature extracted from texts are useful for classifier performance in equal measure. Adding more statistical and semantic features

with values (information) representing the words found in sentences are more useful for the classifier.

Finally, we answer Research Question No. 4 by showing the impact of the feature selection method on classifier performance measures by using the Information Gain attribute evaluation and Ranker search method as suitable methods to optimize the results and reduce data dimensionality. We determined the number of features that can be removed in order to improve the process of the classifier being variable according to the dataset. In the case of the original dataset, the total number of features that can be deleted and dispensed is 21, this did not improve the performance measures; it only reduced the data dimensionality. However, after adding our own statistical and semantic features, it has become possible to dispense with 31 features and resulting in the performance measures of the classifier improving.

REFERENCES

- 1 **Liu, B., 2010.** “Sentiment analysis and subjectivity”. In *Handbook of Natural Language Processing, Second Edition* (pp. 627-666). Chapman and Hall/CRC.
- 2 **Agarwal, A., Xie, B., Vovsha, I., Rambow, O. and Passonneau, R., 2011, June.** “Sentiment analysis of twitter data”. In *Proceedings of the workshop on languages in social media* (pp. 30-38). Association for Computational Linguistics.
- 3 **Abdul-Mageed, M., Kuebler, S. and Diab, M., 2012.** “SAMAR: A system for subjectivity and sentiment analysis of social media Arabic”. In *Proceedings of the 3rd Workshop on Computational Approaches to Subjectivity and Sentiment Analysis (WASSA), ICC Jeju, Republic of Korea.*
- 4 **Al-Ayyoub, M., Essa, S.B. and Alsmadi, I., 2015.** “Lexicon-based sentiment analysis of Arabic tweets”. *International Journal of Social Network Mining*, 2(2), pp.101-114.
- 5 **Korayem, M., Crandall, D. and Abdul-Mageed, M., 2012, December.** “Subjectivity and sentiment analysis of Arabic: A survey”. In *International Conference on Advanced Machine Learning Technologies and Applications* (pp. 128-139). Springer Berlin Heidelberg.
- 6 **Taboada, M., Brooke, J., Tofiloski, M., Voll, K. and Stede, M., 2011.** Lexicon-based methods for sentiment analysis”. *Computational linguistics*, 37(2), pp.267-307.
- 7 **Ding, X., Liu, B. and Yu, P.S., 2008, February.** “A holistic lexicon-based approach to opinion mining”. In *Proceedings of the 2008 international conference on web search and data mining* (pp. 231-240). ACM.
- 8 **Khan, A.Z., Atique, M. and Thakare, V.M., 2015.** “Combining lexicon-based and learning-based methods for Twitter sentiment analysis”. *International Journal of Electronics, Communication and Soft Computing Science & Engineering (IJECSCE)*, p.89.
- 9 **Duwairi, R.M., Ahmed, N.A. and Al-Rifai, S.Y., 2015.** “Detecting sentiment embedded in Arabic social media—a lexicon-based approach”. *Journal of Intelligent & Fuzzy Systems*, 29(1), pp.107-117.

- 10 **Alotaibi, S.S., 2015.** “*Sentiment Analysis in the Arabic Language Using Machine Learning*”, (Doctoral dissertation, Colorado State University. Libraries).
- 11 **Abdulla, N.A., Ahmed, N.A., Shehab, M.A., Al-Ayyoub, M., Al-Kabi, M.N. and Al-rifai, S., 2016.** “Towards improving the lexicon-based approach for Arabic sentiment analysis”. In *Big Data: Concepts, Methodologies, Tools, and Applications* (pp. 1970-1986). IGI Global.
- 12 **Hailong, Z., Wenyan, G. and Bo, J., 2014, September.** “Machine learning and lexicon based methods for sentiment classification: A survey”. In *Web Information System and Application Conference (WISA), 2014 11th* (pp. 262-265). IEEE.
- 13 **Muhammad, I. and Yan, Z., 2015.** “Supervised Machine Learning Approaches: A Survey”. *ICTACT Journal on Soft Computing*, 5(3).
- 14 **Shoukry, A. and Rafea, A., 2015, April.** “A Hybrid Approach for Sentiment Classification of Egyptian Dialect Tweets”. In *Arabic Computational Linguistics (ACLing), 2015 First International Conference on* (pp. 78-85). IEEE.
- 15 **Refaee, E. and Rieser, V., 2014, May.** “Subjectivity and sentiment analysis of Arabic twitter feeds with limited resources”. In *Workshop on Free/Open-Source Arabic Corpora and Corpora Processing Tools Workshop Programme* (p. 16).
- 16 **Wilson, T., Wiebe, J. and Hoffmann, P., 2005, October.** “Recognizing contextual polarity in phrase-level sentiment analysis”. In *Proceedings of the conference on human language technology and empirical methods in natural language processing* (pp. 347-354). Association for Computational Linguistics.
- 17 **Shoukry, A. and Rafea, A., 2012, May.** “Sentence-level Arabic sentiment analysis”. In *Collaboration Technologies and Systems (CTS), 2012 International Conference on* (pp. 546-550). IEEE.
- 18 **Duwairi, R.M., Marji, R., Sha'ban, N. and Rushaidat, S., 2014, April.** “Sentiment analysis in Arabic tweets”. In *Information and communication systems (icics), 2014 5th international conference on* (pp. 1-6). IEEE.
- 19 **Kiritchenko, S., Mohammad, S.M. and Salameh, M., 2016.** “SemEval-2016 Task 7: Determining sentiment intensity of English and Arabic phrases”. *Proceedings of SemEval*, pp.42-51.

- 20 **Abbasi, A., Chen, H. and Salem, A., 2008.** “Sentiment analysis in multiple languages: Feature selection for opinion classification in web forums”. *ACM Transactions on Information Systems (TOIS)*, 26(3), p.12.
- 21 **Alarifi, A., Alsaleh, M. and Al-Salman, A., 2016.** “Twitter turing test: Identifying social machines”. *Information Sciences*, 372, pp.332-346.
- 22 **Prusa, J.D., Khoshgoftaar, T.M. and Dittman, D.J., 2015, May.** “Impact of Feature Selection Techniques for Tweet Sentiment Classification”. In *FLAIRS Conference* (pp. 299-304).
- 23 **Shoukry, A. and Rafea, A., 2012, November.** “Preprocessing Egyptian dialect tweets for sentiment mining”. In *The Fourth Workshop on Computational Approaches to Arabic Script-based Languages* (p. 47).
- 24 **Cateni, S., Vannucci, M., Vannocci, M. and Colla, V., 2012.** “Variable selection and feature extraction through artificial intelligence techniques”. *Multivariate Analysis in Management, Engineering and the Science*, pp.103-118.
- 25 **Abdulla, N.A., Ahmed, N.A., Shehab, M.A. and Al-Ayyoub, M., 2013, December.** “Arabic sentiment analysis: Lexicon-based and corpus-based”. In *Applied Electrical Engineering and Computing Technologies (AEECT), 2013 IEEE Jordan Conference on* (pp. 1-6). IEEE.
- 26 **Hamouda, A.E.D.A. and El-taher, F.E.Z., 2013.** “Sentiment analyzer for Arabic comments system”. *Int. J. Adv. Comput. Sci. Appl*, 4(3).
- 27 **Guyon, I. and Elisseeff, A., 2003.** “An introduction to variable and feature selection”. *Journal of machine learning research*, 3(March), pp.1157-1182.
- 28 **Farghaly, A. and Shaalan, K., 2009.** “Arabic natural language processing: Challenges and solutions”. *ACM Transactions on Asian Language Information Processing (TALIP)*, 8(4), p.14.
- 29 **Kalaivani, P. and Shunmuganathan, K.L., 2013.** “Sentiment classification of movie reviews by supervised machine learning approaches”. *Indian Journal of Computer Science and Engineering*, 4(4), pp.285-292.

- 30 **Duwairi, R.M. and Qarqaz, I., 2016.** “A framework for Arabic sentiment analysis using supervised classification”. *International Journal of Data Mining, Modelling and Management*, 8(4), pp.369-381.
- 31 **Ferri, C., Hernández-Orallo, J. and Modroi, R., 2009.** “An experimental comparison of performance measures for classification”. *Pattern Recognition Letters*, 30(1), pp.27-38.



APPENDIX A

CURRICULUM VITAE



PERSONAL INFORMATION

Name, Surname: Naseer Albuhruzi

Nationality: Iraqi

Date and Place of Birth: 01 Oct.1971, Diyala, Iraq

Marital status: Married

Phone: 009647700446188

E-mail: naseeriborn@gmail.com

EDUCATION

Degree	Institution	Year of Graduation
M.Sc.	Çankaya University Mathematics and Computer Science	2017
B.Sc.	University of Technology Baghdad	1994

WORK EXPERIENCE

Year	Place	Enrollment
1998-2000	Diyala Engineering University	Ass. Programmer
2001-2004	Sanaa, Yemen	Programmer
2005-2009	Ministry of Trade, Diyala branch, Iraq	Programmer
2009- Present	Ministry of Trade, Diyala branch, Iraq	Administrator computer department

FOREIGN LANGUAGES

English